

pyShore: A deep learning toolkit for shoreline structure mapping with high-resolution orthographic imagery and convolutional neural networks

Zhonghui Lv^{a,b}, Karinna Nunez^a, Ethan Brewer^{b,c} and Dan Runfola^b

^aCenter for Coastal Resource Management, Virginia Institute of Marine Science, William and Mary

^bDepartment of Applied Science, William and Mary

^cDepartment of Computer Science and Engineering, New York University

ARTICLE INFO

Keywords:
deep learning
coastal management
remote sensing
semantic segmentation

ABSTRACT

The process of mapping shoreline structures (i.e., riprap, groins, breakwaters or bulkheads) is heavily reliant on in-situ field surveys and manual delineation using orthoimagery or aerial imagery. These processes are time and resource intensive, resulting in update times of longer than a decade for larger waterbodies. In this study, we explore the effectiveness of a deep learning approach to map shoreline armoring structures from remotely sensed high-resolution imagery. We focus on computationally efficient techniques which can be deployed in desktop environments similar to those used by human coders today, with the goal of providing a semi-automated technique which reduces the total amount of time required to delineate shoreline structures. We test a range of architectures using a dataset of over 10,000 observations of four classes of shoreline structure, finding that a ResNet18 based Pyramid Attention Network (PAN) architecture achieves 72% overall accuracy (60 cm resolution), with 80% and 94% prediction accuracy in breakwater and groins, respectively. This relatively lightweight implementation enabled a 1.5 kilometers of shoreline to be processed in 1.4 seconds (GPU) to 2.16 seconds (CPU) in simulated user environments. Finally, we present pyShore, an implementation of this deep learning algorithm made available for human coders to apply as a part of a semi-automated workflow.

CRedit authorship contribution statement

Zhonghui Lv: conceptualized the project, selected and refined relevant algorithms, algorithmic development, data preparation and testing, drafted the paper. **Karinna Nunez:** conceptualized the project, and paper editing. **Ethan Brewer:** data and paper editing. **Dan Runfola:** conceptualized the project, selected and refined relevant algorithms, and paper editing.

1. Introduction & Related Work

Maintaining a current inventory of shoreline structures is essential to assessing shoreline stabilization and the health of nearshore habitats (Berman and Hershner, 1999; Nunez et al., 2020). Shoreline hardening structures that protect upland property from erosion act not only as barriers which inhibit marsh migration (Titus et al., 2009; Gittman et al., 2015; Hill, 2015; Enwright et al., 2016), but also represent barriers for sediment exchange between the marsh habitat and the upland (Hill, 2015). Data on the location and nature of shoreline structures is thus an important element in coastline management strategies (Nunez et al., 2022; Enwright et al., 2016; Hill, 2015; Gittman et al., 2015; Titus et al., 2009).

Today, this data is primarily provided by shoreline inventory reports generated by government bodies (for example,

ORCID(s): 0000-0001-5186-3699 (Z. Lv)

in the United States these are produced by the state of Virginia (CCRM, 2019), Delaware (Berman et al., 2013), Washington (State of Washington, 2009) and others (Fontenault et al., 2013; MGS, 2000)). These reports describe a range of shoreline structures, including erosion control structures such as riprap, bulkheads, and breakwaters; structures such as groins, designed to accumulate sand in transport; and recreational structures, built to enhance public or private use of the water (CCRM, 2019; Fontenault et al., 2013).

The traditional approach to collecting and mapping shoreline structures consists of a GPS field survey to collect coordinates and attribute information of coastal structures, and then manually delineating the structures and extracting basic feature information (e.g., type, material, and length) from remotely sensed data or other available digital images (CCRM, 2019; Fontenault et al., 2013). These processes require time-intensive in-situ surveys and well-trained technicians to carry them out. For example, the latest shoreline inventory for Virginia was a 12-year process (from 2007 to 2019), including in-situ surveys by trained fieldworkers as well as image digitization by experts. This process resulted in updated historic shoreline reports for Virginia and new inventories in locations with no historic information (CCRM, 2019). As another example of the cost of such efforts, the first Massachusetts Inventory of Publically-owned Coastal Infrastructure was collected over a eight-year period beginning in 2002 using a similar approach including GPS field surveys and desktop interpretation of remotely sensed imagery. In this case, the in-situ survey work took 5 years, while the digitization and GIS data production stage took 3 years of effort (Fontenault et al., 2013). While this approach to assessing the coastal shoreline inventory can generate spatially explicit and highly-resolved outputs, the decade-long process of generating such information can also inhibit contemporary accuracy as shoreline structures are removed or added due to shoreline erosion, sea-level rise, or construction (Gittman et al., 2015; Enwright et al., 2016). This can produce a large gap between the generated data product and the real-world shoreline structure topography (Niculescu et al., 2021).

Over the past three decades, a variety of different techniques have been developed for the extraction of shoreline from remotely sensed imagery. Imagery sources used in the literature range from coarse resolution (Bagli and Soille, 2004; Ghorai and Mahapatra, 2020; Kuleli et al., 2011; Spinosa et al., 2021) to high-resolution (Sekovski et al., 2014; Tarmizi et al., 2014), and include LiDAR-derived data (Palaseanu-Lovejoy et al., 2016; Eboigbe et al., 2020; Liu et al., 2011), Synthetic Aperture Radar (SAR) imagery (Mason and Davenport, 1996; Spinosa et al., 2021), and Multispectral Satellite Imagery (MSI) (Sekovski et al., 2014). A common methodology includes classification of pixels in satellite imagery to segment the image into different features (Bagli and Soille, 2004; Di et al., 2003; Kuleli et al., 2011), then using machine learning (Kumar et al., 2020; McAllister et al., 2022a) or threshold analysis (Kuleli et al., 2011; Ghorai and Mahapatra, 2020; Mason and Davenport, 1996) to classify the features into different categories of land or water body, extracting the shorelines from the boundaries between the two. During analysis of the data, techniques such as image enhancement (adjustment of brightness and contrast of individual features (Loos and Niemann, 2002)),

edge detection (Mason and Davenport, 1996; Loos and Niemann, 2002; Kumar et al., 2020; Spinosa et al., 2021), morphological operations (Spinosa et al., 2021; Ghorai and Mahapatra, 2020), water indexing (Ghorai and Mahapatra, 2020; McAllister et al., 2022b), and GIS processing (Ghorai and Mahapatra, 2020; Zarillo et al., 2008) may be applied to improve the primary technique and achieve higher accuracy.

Some studies use object-based image analysis (OBIA) which first groups individual pixels into groups of pixels likely reflective of the same objects using a range of algorithms (McAllister et al., 2022a; Gong et al., 2013), treating the grouped pixels as the unit of analysis. The spectral information of the grouped pixels, shapes, texture, and other topological features are then used to classify the segments into different categories (McAllister et al., 2022b; Niculescu et al., 2021). Application-specific tools and metrics are also commonly developed such as beachtool (Zarillo et al., 2008) to extract shoreline locations, and cliffmetric to extract cliffs in coastal areas (Liu et al., 2011; Payo et al., 2018). However, to date the existing literature has focused on the extraction of shorelines from satellite images — to our knowledge, no studies have focused on automatically extracting different types of shoreline armoring structures, the focus of this study.

In this context, the automatic delineation of shoreline armoring structure based on orthographic imagery can help to “fill in the gaps” between in-situ survey efforts, as well as provide more temporally-explicit information on the creation and removal of coastal features. Such automated delineation would benefit many different research fields, including hazard management and mitigation, coastal vulnerability and erosion assessment, environmental risk analysis, and coastal spatial planning and management.

Advances in computer vision for remote sensing applications suggest promise for shoreline structure detection algorithms. In recent years, object detection via remotely sensed images has received considerable attention due to the availability of Very High-Resolution (VHR; i.e., sub-meter) and multispectral sensors (NOAA, 2018; DigitalGlobe, 2018). Since 2012, orthographic imagery analysis using deep learning methods, specifically convolutional neural networks (CNNs) has grown in popularity (Krizhevsky et al., 2012). Recent examples of the use of CNNs with orthoimagery include the detection of buildings (Yang et al., 2017), roads (Brewer et al., 2021; Narayan et al., 2017), marine debris (Kikaki et al., 2022), coastal vegetation mapping (Li et al., 2021; Guirado et al., 2017), landuse mapping (Bhosle and Musande, 2019), and other types of analysis and applications (Runfola et al., 2022, 2021; Goodman et al., 2021; Brewer et al., 2022). The wide range of applications of deep learning techniques more broadly have driven a number of research groups to explore ways to tailor approaches to orthoimagery, with a wide range of modeling approaches being promoted in recent years (Runfola, 2022; Wang et al., 2022; An et al., 2021; Roberts et al., 2022; Tran et al., 2022; Aravena Pelizari et al., 2021).

In this study, we compare multiple state-of-the-art deep learning techniques, specifically an adapted U-Net architecture (Ronneberger et al., 2015), DeeplabV3 (Chen et al., 2016), and Pyramid Attention Network (PAN) (Li et al.,

2018) to estimate the spatial location of shoreline structures using high resolution orthographic imagery. The proposed methodology will advance shoreline inventory protocols by identifying and mapping shoreline structures at a regional scale, but with a high spatially explicit resolution within an efficient time frame. We specifically seek to introduce a technique that can be used as a part of a semi-automated pipeline for shoreline inventory collection, aiding individuals digitizing orthoimagery in delineating line features more efficiently than would be feasible without automated shoreline feature detection.

Our paper is organized as follows: in section 1.1 (“Related Work”), we provide a brief overview of relevant literature on mapping and detection tasks, including structure detection, using orthoimagery and deep neural networks (DNNs). In section 2 (“Material & Methods”), we discuss the data and technical approach we use in the experiment. We introduce our results and provide a discussion in sections 3 and 4. In the discussion, we further present a semi-automated toolkit, *pyShore*, for shoreline armoring structure classification in the ArcGIS Pro software with a user-friendly interface to facilitate the data collection of future work (section 4.5). Finally, we provide a brief conclusion summarizing our findings in section 5.

1.1. Related Work

A significant body of literature exists on the use of segmentation and object detection algorithms with remotely sensed imagery (for a review, see Cheng and Han (2016)). Object detection algorithms have been applied for detection and extraction of features of interest including roads (Wan et al., 2021), buildings (Etten et al., 2021), and vehicles (Chen et al., 2014). Several issues - such as varying size, background, and orientation of target objects - make the automatic detection of features in remotely sensed imagery a challenging problem (Brewer et al., 2022; Bhil et al., 2022). Over the past few years, U-Nets (Ronneberger et al., 2015), and variants of region-based convolutional neural networks (R-CNNs) (He et al., 2017), have become popular deep neural network (DNN) based approaches for object detection and segmentation in the greater computer vision community, including within the computer vision remote sensing (CVRS) domain. Some examples of the use of deep learning architectures in CVRS include building detection from high-resolution multispectral imagery (Prathap and Afanasyev, 2018), and building footprint and road detection within OpenStreetMap (OSM) fused with Sentinel-1 and 2 imagery (Ayala et al., 2021).

A small number of studies have been conducted extending specifically to marine science. In 2021, Li et al. employed a U-Net and developed an adaptive deep learning approach to map salt marshes in estuarine emergent wetlands in South Carolina, USA from 20 Sentinel-2 images with an accuracy of 90% (Li et al., 2021). In 2022, marine debris was identified from sentinel-2 imagery using a random forest model (Breiman, 2001) and a deep learning architecture (Kikaki et al., 2022). Other work has explored the effectiveness of deep learning to identify marine objects such as ships (Stofa et al., 2020) and whales (81% detection accuracy) (Guirado et al., 2019). No work to-date has explored

138 2. Material & Methods

139 2.1. Study area

140 In this study, we focus on 17,239 kilometers of shoreline located in the state of Virginia in the United States (see
141 Figure 1). The majority of this shoreline surrounds the Chesapeake Bay, a protected estuary responsible for over
142 a hundred billion US dollars of economic output every year, predominantly related to commercial fishing, tourism,
143 recreation, and timber (Phillips and McGee, 2014). The Chesapeake Bay is a focus of estuary protection and renewal
144 activities from the State of Virginia, and also many US federal agencies (Phillips and McGee, 2014). It is further a
145 region with multiple climate mitigation efforts to protect both natural and human built structures (Najjar et al., 2010;
146 Toft et al., 2017; Du et al., 2016).

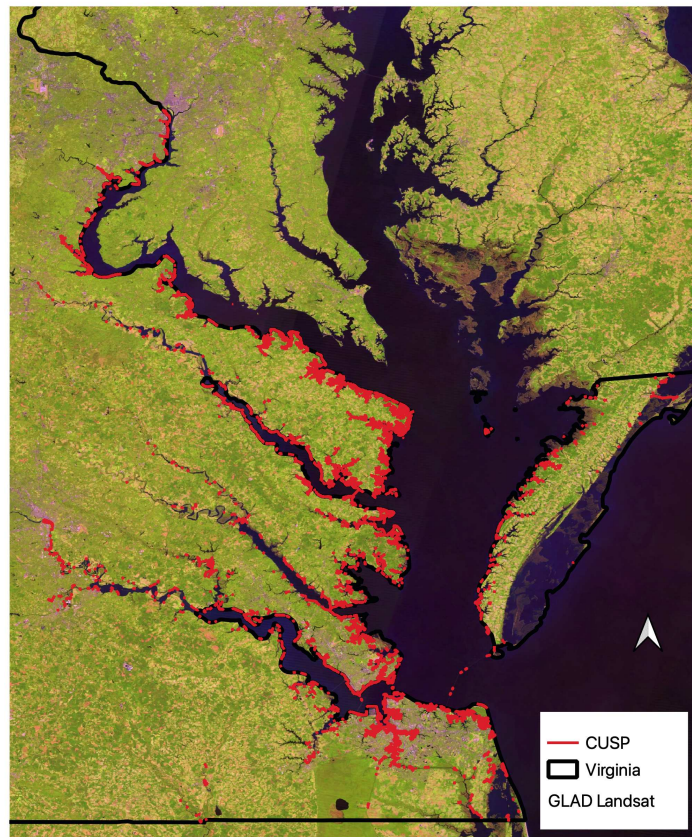


Figure 1: Shoreline structures extracted from CUSP (NOAA, 2021a) along Virginia's coastal regions (Runfola et al., 2020). Basemap is Landsat imagery downloaded from the Global Land Analysis and Discovery (GLAD, 2018).

2.2. Data - Labels

Two different datasets are leveraged to identify the geographic location of shoreline features (i.e., to label areas as belonging to a class of shoreline structure). This information is used to train the proposed deep learning approach regarding what geometric features and colors are associated with each type of shoreline structure. The two datasets we use for labeling are the NOAA Continually Updated Shoreline Product (CUSP), and the Massachusetts Private Shoreline Stabilization Structures dataset.

2.2.1. NOAA's Continually Updated Shoreline Product

The Continually Updated Shoreline Product (CUSP) published by the National Oceanic and Atmospheric Administration (NOAA) National Geodetic Survey (NGS) provides a complete shoreline representation along the U.S. coastline and its territories (NOAA, 2021a,b). It was created to deliver accurate, contemporary shoreline information across the continental U.S to support various applications including coastal planning, hazard management and mitigation, storm management, environmental studies, and informing policy and decision makers in coastal management (NOAA, 2021b). The geographic location of shoreline along the coast of Virginia were extracted from CUSP to establish the geographic location of eligible pixels to consider for classification. It has an average line resolution of 103 meters, and covers a total of 17,239 kilometers of coastline in Virginia.

From the full coastline, we subset to 945 kilometers of shoreline at which one of four hardened shoreline structure types was identified - the focus structures of this study. These four shoreline structure types - breakwater, rip rap, groin, and bulkhead - make up the majority of shoreline structures that are built in Virginia to defend against sea level rise, flooding, and coastal erosion (CCRM, 2019). Breakwater (Fig. 3d) is an always-dry (not covered at high water under average meteorological conditions) structure protecting a shore area, harbor, anchorage, or basin from waves. Riprap, as seen in Fig. 3c, is a shoreline structure built by layers of broken rock, cobbles, boulders, or fragments of sufficient size to resist the erosive forces of flowing water and wave action. Groin (Fig. 3b) is an always-dry, shallow, artificial, wall-like structure of durable material extending from the land to seaward for a particular purpose, such as to prevent coast erosion. Bulkheads (or sea walls; Fig. 3a) are embankments or walls for protection against waves or tidal action along a shore or waterfront. The shoreline structures metadata within CUSP was used to label pixels to its corresponding shoreline structure category. Fig. 2 summarizes the number of features (i.e., discrete lines representing shore features) used to label each pixel.

2.2.2. Massachusetts Private Shoreline Stabilization Structures

As figure 2 illustrates, the data labels from CUSP in Virginia are highly imbalanced, largely due to a limited number of labeled features for groins and breakwater. In order to increase the number of training dataset examples and create a more balanced dataset for training, we extract additional groins features from a secondary dataset published by the

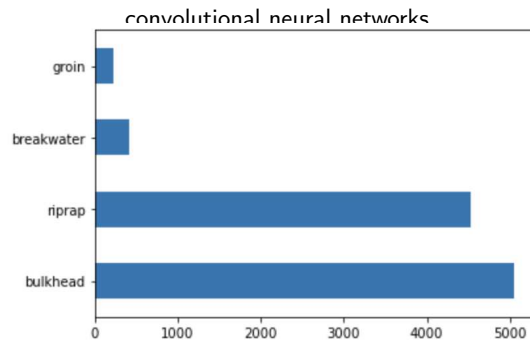


Figure 2: CUSP data distribution in Virginia.

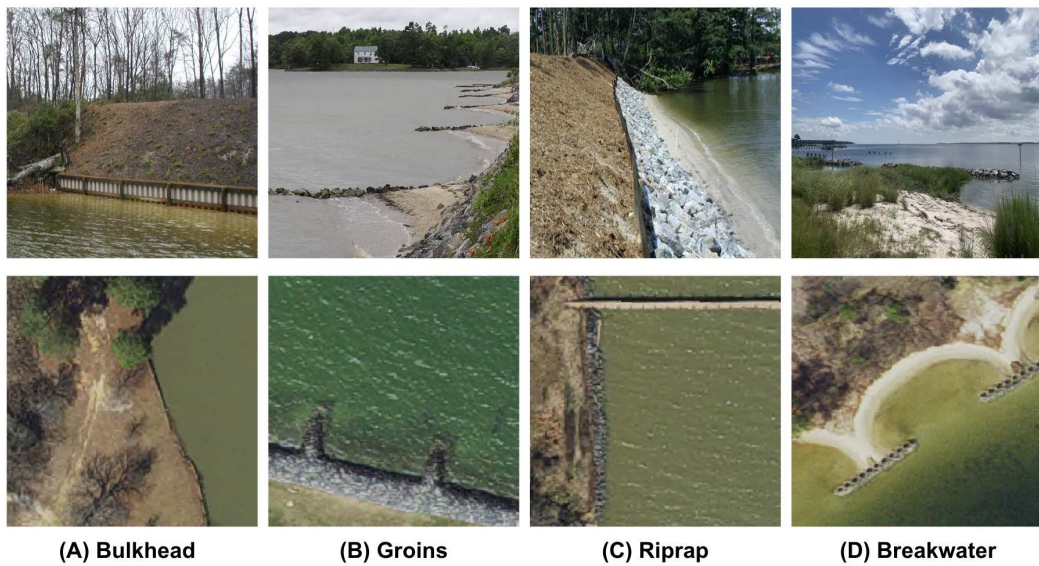


Figure 3: A visualization of the four shoreline armoring techniques we focus on in this study. The upper four images are taken on-site, and the lower four images are from VA orthoimagery

Massachusetts Coastal Infrastructure Inventory (Fontenault et al., 2013). This dataset was developed for Massachusetts Office of Coastal Zone Management to monitor coastal structures that were built to protect public and private development in dynamic coastal areas (Fontenault et al., 2013). There are 1,957 geocoded groins available in this dataset, collected in 2013 and digitized using 30 cm resolution 2008/2009 USGS color orthographic imagery. The digitized groins are combined with CUSP features to construct a broader dataset for model training and validation.

2.3. Data - Imagery

2.3.1. Virginia Base Mapping Program (VBMP)

The Virginia Base Mapping Program (VBMP) was initiated in 2001 to develop orthoimagery for the Commonwealth of Virginia. The purpose of this program is to create a consistent, accurate base map that all state, local, and federal government agencies, as well as private sectors and academic intuitions can use for spatial data applications

(VGIN, 2018). The VBMP imagery leveraged was captured at a 12-inch (30.48cm) ground sample distance and acquired during the leaf-off period in 2021 with 3-band spectral resolution (red, green, blue) across the Commonwealth of Virginia.

We specifically leverage a total of 1,103 image tiles (the size of each image tiles varied, and the average image size is 5000×5000 pixels) that intersect with the CUSP coastal line data. Files were retrieved from VGIN in MrSID format, converted to geotiffs, and then labeled using the georeferenced CUSP data for further analysis as described above.

2.3.2. Massachusetts High-Resolution Orthoimagery

Thirty-centimeter resolution 2008/2009 USGS Color Ortho Imagery was retrieved for all areas covered by the Massachusetts Coastal Infrastructure Inventory Groins, representative of the same imagery used for digitization (Fontenault et al., 2013). These images are 3-band (Red, Green, Blue) natural color. A map of the bounding boxes of these images can be seen in figure 4.

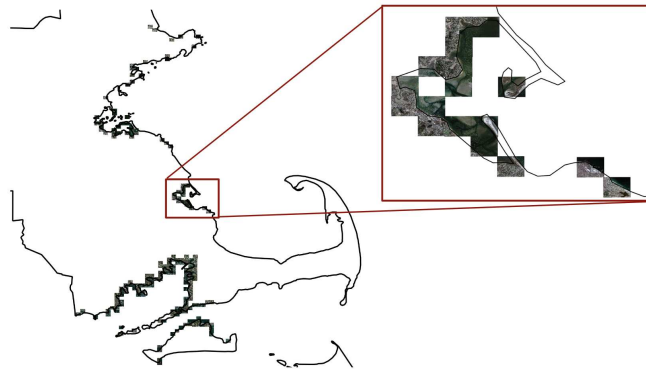


Figure 4: A map of the bounding boxes of imagery in Massachusetts with a cluster of groins near Plymouth, MA expanded. Basemap is from geoboundaries.org (Runfola et al., 2020) and imagery is from Mass.gov (USGS, 2012)

2.3.3. National Agricultural Imagery Program

Aerial images collected by the United States Department of Agriculture's (USDA) National Agricultural Imagery Program (NAIP) (OCM-Partners, 2022) were also acquired for this study in order to analyze the relationship between imagery resolution and model accuracy, as well as to explore the potential benefits of datasets incorporating a near-infrared band for shoreline structure identification.

The NAIP dataset is a time series aerial imagery database that has records extending to 2003, sponsored through a collaboration between the United States Geological Survey (USGS) and U.S. state governments (OCM-Partners, 2022). The initial NAIP acquisition cycle was five years, before changing to three years in 2009. NAIP imagery is generally acquired at a 1-meter spatial resolution with 4-band spectral resolution (red, green, blue and near-infrared) across the continental United States during the agricultural growing season. In recent years, some states began publishing the

dataset with higher spatial resolution (including Virginia). Imagery is provided orthorectified; for our study, we use the NAIP imagery acquired 2018 in Virginia with 0.6 m spatial resolution. A total of 316 image tiles (standardized to a size of 5000×5000 pixels) provided complete coverage of all coastal features in the CUSP database.

2.4. Data processing - annotation process & protocol

A python processing script was developed to automate data pre-processing, labelling, and generation of training and validation data. Of the 10,194 filtered shoreline structures identified in CUSP coastal lines (1,055 km), 5,040 (529 km) were labeled bulkhead, 4,519 (489 km) riprap, 221 (6 km) groin, and 414 (31 km) breakwater. A 3-meter buffer was constructed along the filtered shoreline structures in order to provide context for image classification as well as mitigate minor errors or differences in georeferencing across imagery and shoreline label sources.

Here we present a brief example of our processing procedures, using NAIP imagery as an illustrative case. First, each 5000×5000 pixel NAIP image tile is cropped into $400 \times 256 \times 256$ image patches. Only image patches that intersect with CUSP features are kept for further data processing and analysis, resulting in a dataset of 4,532 256×256 tiles covering coastal Virginia. The pixels in these image patches are labeled using the CUSP dataset, by first masking all pixels that are not shoreline as background, and then labeling the remaining pixels according to the corresponding class values in CUSP (bulkhead, riprap, groin, and breakwater). This procedure results in a series of 256×256 masks with each shoreline pixel labelled. Fig. 5 is an example of an original NAIP tile, cropped to 256×256 pixels, and the corresponding label mask on the right. This data is used to construct training, validation, and testing datasets for the model development, calibration, and validation.

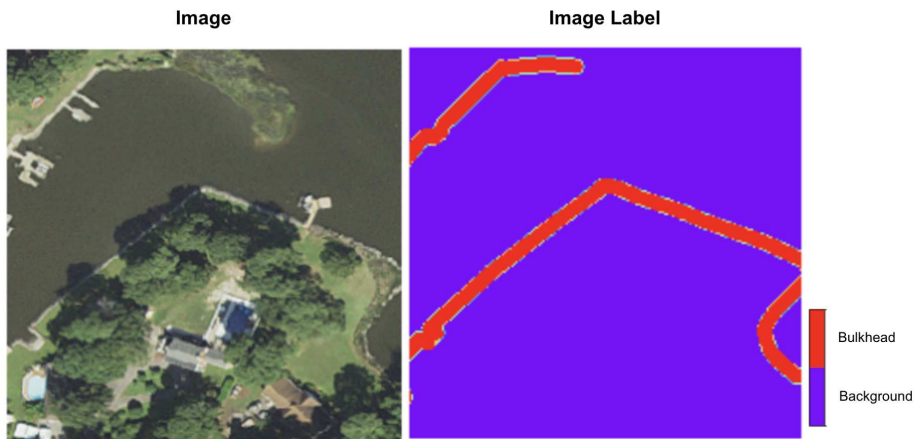


Figure 5: An example case of cropped image patch and its associated mask.

In addition, for each 256×256 tile, a secondary dataset is generated in which the entire image is labeled as a single class based on the most common shoreline structure. This secondary dataset is used to establish baseline accuracy of

Table 1

Summary of the number of pixels (and percentage) in each dataset used in model training and testing. The green column represents number of pixels labeled as bulkhead, the red column represents number of pixels labeled as riprap, the yellow column indicates the number of pixels labeled as groin, and the blue column shows the number of pixels labeled as breakwater.

Dataset Name	Training Data				Testing Data			
	(pixel count and percentage per class)				(pixel count and percentage per class)			
VA CUSP/VBMP	1806539	1704174	91117	214871	391008	330605	23377	50356
	47%	45%	2%	6%	49%	42%	3%	6%
VA CUSP/VBMP & MA	1806539	1704174	374952	214871	406597	427725	270849	81867
	44%	42%	9%	5%	29%	24%	43%	4%
VA CUSP/NAIP	2727071	2660187	60866	242668	599562	536705	14662	54891
	48%	47%	1%	4%	50%	46%	1%	6%

methodologies that do not involve localization (i.e., to contrast pixel vs. scene resolution classification accuracy).

This process is repeated for three datasets - VBAP images, MA high-resolution orthoimagery, and NAIP. We leveraged these datasets to conduct a wide range of tests regarding the performance of our approach under a range of different circumstances.

Of the collected shoreline structure images and labels in the three datasets, the number of pixels per each class in the training and testing subsets are summarized in Table 1. Observed data imbalance is taken into account in the training process with a weighting scheme on the loss function; we present details of our approach in section 2.5.

2.5. Methodology

2.5.1. Architectures for image segmentation

Three predefined state-of-the-art semantic segmentation architectures are tested and compared considering their generalization capacities, U-Net (Ronneberger et al., 2015), DeeplabV3 (Chen et al., 2016), as well as Pyramid Attention Network (PAN) (Li et al., 2018).

U-Net: The U-Net is a well-established, relatively light weight (under 10 million parameters and forward-pass timings of less than a second for 256×256 images) deep learning algorithm for semantic segmentation based on convolutional network architectures (Ronneberger et al., 2015). The algorithm's architecture includes two parts: down-sampling and up-sampling, also called the encoder and decoder. The encoder extracts varying resolution feature maps through a series of convolutional, rectified linear units (ReLU), and max-pooling layers (Li et al., 2021). The decoder stage contains and combines (a) each feature map from the down-sampling process, and (b) spatial information through an up-sampling and concatenation process (Fig. 6). This data flow of down-sampling and up-sampling constructs a U-shape of architecture, thus the output layer maintains the same resolution as the input layers. One key benefit of using this learning architecture is that it has been shown to be effective in cases with few training images, while still

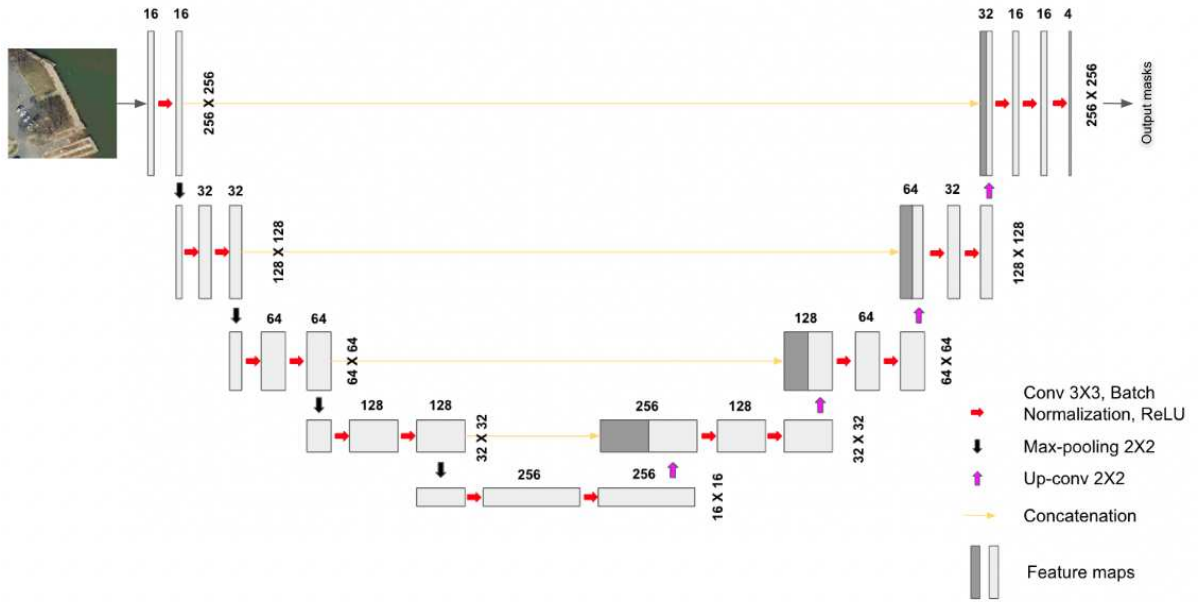


Figure 6: The U-Net architecture (Example of a 3-band input image with 256 × 256 pixel-size). The boxes indicate the feature maps at each layer, and the number on the top of each feature map shows the depth of feature map (channel). Numbers on the right side of each feature map are image/feature maps dimension.

251 In this piece, we modify the U-Net architecture in order to identify a deep learning model that can both accurately
 252 delineate shoreline structures, and do so in a way that is computationally suitable for deployment on contemporary
 253 desktops and laptops. We modify the U-Net in two key ways: (a) we provision alternative classifiers to handle en-
 254 coding and decoding tasks, and (b) we adapt the algorithm to dynamically accept either 4-band (RGB,NIR) or 3-band
 255 (RGB) high-resolution imagery (noting our labeled data is sourced from both three and four band sensors). The spe-
 256 cific alternative classifiers that are contrasted in this piece to assess their capabilities in shoreline structure detection
 257 include modifications to the U-Net which leverage a small set of convolutional approaches chosen for their relative
 258 computational efficiency - Resnet-18, Xception, and the original U-Net implementation (Ronneberger et al., 2015) -
 259 as well as two deeper (i.e., more computationally intensive) networks to provide contrasting cases - Resnet-101 and
 260 InceptionResNet-v2.

261 *DeeplabV3:* The DeeplabV3 algorithm is designed for image segmentation. It introduces atrous convolution (Chen
 262 et al., 2016) in the down-sampling process to efficiently produce high level feature maps while keeping accurate lo-
 263 cation information, which can effectively enlarge the field of view of filters without increasing the number of training
 264 parameters. The deeplabV3 also uses atrous spatial pyramid pooling (ASPP) to capture objects and useful image con-
 265 text at multiple scales (Chen et al., 2016). This method has recently been applied in many semantic segmentation tasks

with remote sensing data and proved to be an effective technique in many fields (Wang et al., 2022; Liu et al., 2021).

Pyramid Attention Network: Different from U-Net and DeeplabV3, PAN is built on an attention mechanism, which aims to imitate human visual systems to effectively find salient regions in a complex scene. The attention mechanism is combined with spatial pyramid to extract precise dense features for pixel labeling (Li et al., 2018), instead of using atrous convolution like DeeplabV3 and artificially designed decoder networks like U-Net.

Within each of these architectures, we further test a range of up- and down-sampling convolutional strategies, including Resnet-18, Resnet-101, xception, and InceptionResnet.

2.5.2. Loss functions

The loss function is the proximity between the ground truth label data and the predicted class. In this study, we implement a multi-class cross-entropy loss:

$$loss = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (1)$$

where M is the total number of shoreline structure classes (M = 4 in this study, background is excluded from the loss) and y is the binary indicator (0 or 1) if class label c is the correct classification for observation o, p is the predicted probability observation o is of class c. Due to the imbalance of our pixel-level data distribution, a weighting scheme is used in the training process, in which classes are weighed according to their representation in the labeled data (Kikaki et al., 2022; Paszke et al., 2016):

$$W_{class} = \frac{1}{\ln(c + p_{class})} \quad (2)$$

where W_{class} is a multiplicative weight applied to the loss function for observations of a given class, $p_{class} = N_{pixels of class} / N_{total pixels}$, where $N_{total pixels}$ excludes the background pixels in the training set, and c is an additional hyper-parameter set to 1.03 (following past literature; see Kikaki et al. (2022)). During the training process, Adam optimization is used to minimize the cross-entropy loss with an initial learning rate of 1×10^{-3} .

2.5.3. Model validation and application

We implement a wide range of validation metrics to assess pixel-level semantic segmentation performance. First and foremost, we present overall accuracy (equation 3) to give guidance to the overall performance a user might expect. In addition, we evaluate the intersection-over-union (IoU) (equation 4), the average F1 score for each class (Macro-F1)

(equation 5), and the precision (equation 6) and recall (equation 7) at the pixel level for each class.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{IoU}(U, V) = \frac{|U \cap V|}{|U \cup V|} \quad (4)$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

In equations 3, 5, 6, and 7, TP (true positive) represents the number of pixels in which the model correctly predicts ground truth, TN (true negative) represents outcomes in which the model correctly predicts cases that are different than ground truth, FP (false positive) is predictions to a class that do actually not belong to that class, and FN (false negative) are the number of predictions belonging to a class but were predicted to be in a different class. The F1 score is a harmonic mean between precision and recall.

The IoU (equation 4), defined as the size of the intersection divided by the size of the union of two label sets, is used to compare the set of predictions for a sample to the corresponding set of ground truth labels. IoU provides a conservative estimate of model performance, penalizing for both FP and FN at the pixel level (Rezatofighi et al., 2019). Lastly, a cross-validation matrix for each model is additionally generated to compare the prediction accuracy for each type of shoreline structure.

3. Results & Analysis

The results of each test, including permutations of the datasets described in section 2.3 and the architectures described in section 2.5.1, are presented in Table 2; the results for tests using NAIP imagery are summarized in Table 3.

The first set of results (the first three rows) in Table 2 present a U-Net implementation in which the encoder and decoder is the same as in the original definition of the architecture (Ronneberger et al., 2015) (i.e., a “Backbone” of “Original”). In Table 2, in order to evaluate model performance in the Virginia region specifically, we delineate between models that are tested solely on coastal features in Virginia, and those that are tested on a dataset which includes both Virginia and Massachusetts features. For example, the results indicate that when the “Original” U-Net implementation is tested (with the default encoder and decoder):

1. When trained and tested on the integrated Virginia and Massachusetts (VA and MA) datasets, the overall accuracy (OA) is 75%; the best performing class is Groins ($F1 = 0.96$); worst is Breakwater ($F1 = 0.4$).
2. When trained on VA and MA, but then tested only on VA, the overall accuracy is 59%.
3. When trained on VA data alone, and then tested on VA alone, the overall accuracy is 61%.

Thus, the first set of our results provides evidence that suggests that - if a default U-Net implementation is leveraged - the best overall model performance across all of our data is found if all data from VA and MA is pooled. However, for the model we are most interested in - a model that succeeds in classifying the Chesapeake Bay and related shorelines in Virginia - we find that limiting training data to Virginia can result in a slightly better performing model based on overall accuracy.

We expand this study to a range of alternative classifiers (“backbones”), in which we exchange the default encoder and decoder of the U-Net, DeeplabV3, PAN architecture with a range of popular network architectures (Resnet-18 and Resnet-101 for all three architectures, xception and inceptionResnetV2 for U-Net, with transfer learning weights based on ImageNet). Across all of these permutations, we identify PAN-resnet101 as the best global classifiers, achieving 83% overall accuracy when trained and tested on the integrated Virginia and Massachusetts dataset. The performance of PAN-resnet18 is particularly notable in this scenario, as it achieved 81% overall accuracy, but with far less complexity (11M parameters) than the best classifier (43M). This indicates that a lighter-weight architecture may be appropriate for shoreline structure delineation, which can enable deployment on - for example - lower memory environments.

When focusing only on validation with Virginia basemap (3-band) imagery data, we find the highest overall validation accuracy is 72% (using PAN-resnet101 implementations). Using the lighter weight PAN-resnet18, the overall validation accuracy achieved was 69%. Similar to the joint Massachusetts and Virginia case, this highlights the value

Table 2

Results of Architecture Tests. Results are presented broken out by model implementation, test and training datasets, and each of the four tested classes. Each tested class is presented with four statistics: F1 score (F1), Intersection over Union (IoU), Precision (Prec.), and Recall (Rec.). Global statistics are presented in the final four columns, including mean Intersection over Union (mIoU), Overall Accuracy (OA), and Macro-Averaged F1 score (F1-macro).

Model Details		Dataset Name		Bulkhead				Riprap				Groins				Breakwater				mIoU	OA	F1-macro
Architecture	Decoder	Trained	Tested	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.			
UNet	Original param: 0.8M	VA&MA	VA&MA	0.66	0.5	0.68	0.65	0.54	0.37	0.55	0.53	0.96	0.93	0.99	0.94	0.4	0.25	0.29	0.6	0.51	0.75	0.64
		VA	VA	0.67	0.5	0.68	0.65	0.55	0.38	0.57	0.53	0.27	0.16	0.42	0.2	0.41	0.26	0.31	0.6	0.32	0.59	0.47
		VA	VA	0.67	0.51	0.71	0.64	0.59	0.42	0.57	0.6	0.28	0.16	0.4	0.21	0.42	0.26	0.34	0.54	0.34	0.61	0.49
UNet	Resnet-18 14.3M	VA&MA	VA&MA	0.68	0.52	0.71	0.65	0.61	0.44	0.59	0.63	0.96	0.93	0.96	0.97	0.55	0.38	0.52	0.58	0.57	0.78	0.7
		VA&MA	VA	0.68	0.52	0.71	0.65	0.61	0.44	0.6	0.63	0.35	0.21	0.3	0.41	0.56	0.39	0.53	0.58	0.39	0.63	0.55
		VA	VA	0.73	0.57	0.73	0.72	0.64	0.47	0.63	0.65	0.41	0.25	0.73	0.28	0.52	0.35	0.47	0.59	0.41	0.67	0.57
UNet	Resnet-101 51.5M	VA&MA	VA&MA	0.72	0.56	0.74	0.69	0.66	0.49	0.62	0.69	0.97	0.94	0.97	0.96	0.56	0.39	0.59	0.54	0.59	0.8	0.73
		VA&MA	VA	0.72	0.56	0.75	0.69	0.66	0.49	0.63	0.69	0.41	0.26	0.4	0.42	0.57	0.4	0.61	0.54	0.43	0.68	0.59
		VA	VA	0.75	0.6	0.75	0.75	0.68	0.51	0.67	0.68	0.53	0.36	0.68	0.44	0.6	0.43	0.57	0.64	0.48	0.71	0.64
UNet	xception 29M	VA&MA	VA&MA	0.73	0.58	0.78	0.69	0.69	0.53	0.63	0.77	0.98	0.96	0.99	0.97	0.58	0.41	0.69	0.5	0.62	0.82	0.75
		VA&MA	VA	0.73	0.58	0.79	0.69	0.69	0.53	0.63	0.77	0.55	0.38	0.64	0.49	0.59	0.42	0.71	0.5	0.48	0.7	0.64
		VA	VA	0.77	0.62	0.76	0.77	0.7	0.54	0.69	0.7	0.47	0.31	0.43	0.52	0.6	0.43	0.7	0.53	0.47	0.72	0.63
UNet	inceptionresnetv2 62M	VA&MA	VA&MA	0.74	0.58	0.76	0.71	0.68	0.52	0.64	0.73	0.98	0.96	0.99	0.97	0.59	0.42	0.67	0.52	0.62	0.82	0.75
		VA&MA	VA	0.74	0.58	0.77	0.71	0.69	0.52	0.65	0.73	0.58	0.41	0.63	0.53	0.59	0.42	0.69	0.52	0.48	0.7	0.65
		VA	VA	0.75	0.6	0.76	0.73	0.69	0.52	0.66	0.72	0.53	0.36	0.74	0.42	0.51	0.34	0.52	0.5	0.46	0.7	0.62
DeeplabV3	Resnet-18 16M	VA&MA	VA&MA	0.66	0.49	0.7	0.62	0.58	0.41	0.55	0.62	0.95	0.9	0.95	0.94	0.48	0.31	0.44	0.53	0.53	0.76	0.67
		VA&MA	VA	0.66	0.49	0.7	0.62	0.59	0.42	0.56	0.62	0.41	0.26	0.32	0.56	0.54	0.37	0.56	0.53	0.38	0.61	0.55
		VA	VA	0.68	0.52	0.7	0.67	0.61	0.44	0.59	0.63	0.36	0.22	0.42	0.31	0.56	0.39	0.59	0.53	0.39	0.64	0.55
DeeplabV3	Resnet-101 59M	VA&MA	VA&MA	0.7	0.54	0.72	0.68	0.61	0.44	0.59	0.63	0.95	0.91	0.96	0.94	0.46	0.3	0.41	0.53	0.55	0.78	0.68
		VA&MA	VA	0.7	0.54	0.73	0.68	0.62	0.45	0.61	0.63	0.36	0.22	0.32	0.41	0.51	0.34	0.49	0.53	0.39	0.64	0.55
		VA	VA	0.74	0.58	0.76	0.72	0.67	0.51	0.64	0.7	0.49	0.33	0.53	0.46	0.52	0.35	0.55	0.5	0.44	0.69	0.61
PAN	Resnet-18 11M	VA&MA	VA&MA	0.74	0.58	0.72	0.75	0.65	0.48	0.65	0.65	0.97	0.94	0.97	0.96	0.6	0.43	0.66	0.55	0.61	0.81	0.74
		VA&MA	VA	0.74	0.58	0.73	0.75	0.65	0.48	0.65	0.65	0.47	0.31	0.45	0.5	0.64	0.48	0.78	0.55	0.46	0.69	0.63
		VA	VA	0.73	0.57	0.72	0.74	0.66	0.49	0.65	0.67	0.52	0.35	0.83	0.38	0.7	0.54	0.83	0.6	0.49	0.69	0.65
PAN	Resnet-101 43M	VA&MA	VA&MA	0.76	0.61	0.76	0.75	0.69	0.53	0.69	0.7	0.98	0.95	0.99	0.97	0.63	0.46	0.57	0.7	0.64	0.83	0.76
		VA&MA	VA	0.76	0.61	0.76	0.75	0.69	0.53	0.69	0.7	0.51	0.34	0.56	0.46	0.65	0.48	0.61	0.7	0.49	0.72	0.65
		VA	VA	0.75	0.6	0.75	0.76	0.68	0.51	0.66	0.7	0.51	0.34	0.79	0.37	0.56	0.39	0.66	0.49	0.46	0.7	0.63

of the lighter weight PAN-resnet18 implementation; further, we note that UNet-xception came in a close second with 70% accuracy (29M parameters).

In the results presented in Table 2, the models trained on the Virginia dataset have no particular class correlated closely with overall performance. The overall accuracy in Virginia shoreline features ranges from 61% to 72%, with each model performing relatively poorly with groins (noting model performance varies most in this class). With 391,008 pixels labeled as bulkhead in the Virginia test set, each model performed relatively well at prediction of the class, with relatively little variance (0.05%) across models.

Introducing additional examples of groin features into the training dataset proved to be relatively inefficient in improving overall accuracy for the Virginia data, suggesting that groin features in Massachusetts and those in Virginia may be distinct enough to require specialized approaches to cross-domain learning. However, for model development that may be focusing on global results (i.e., not domain specific to Virginia), the additional samples had a range of advantages. Adding the geocoded groin features from MA into training, the precision for predicting groins improved dramatically - for UNet-xception tested on VA to the same model tested on the merged MA and VA dataset - from 43% to 99%. For groin features specifically, the best model tested on the merged dataset achieves 99% accuracy, 97% recall, and 98% F1 score. Figure 7 presents the visualization of predictions from models trained on Virginia VBMP dataset.

Table 3 highlights results when models are tested using only Virginia imagery (testing and training) with the NAIP dataset. We find broadly comparable results to the cases presented in Table 2, with a slight performance gain in the

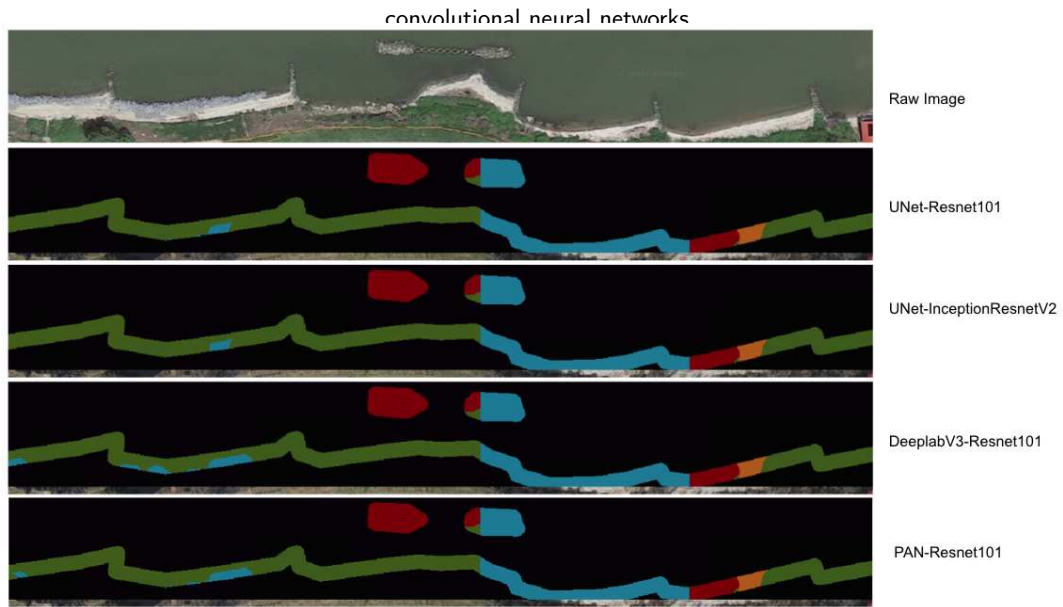


Figure 7: Result visualization predicted by different models trained on VBMP 3-band imagery.

Table 3

Architecture results with NAIP imagery. Each tested class is presented with four statistics: F1 score (F1), Intersection over Union (IoU), Precision (Prec.), and Recall (Rec.). Global statistics are presented in the final four columns, including mean Intersection over Union (mIoU), Overall Accuracy (OA), and Macro-Averaged F1 score (F1-macro).

Model Details		Dataset Name		Bulkhead				Riprap				Groins				Breakwater				mIOU	OA	F1-macro
Architecture	Decoder	Trained	Tested	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.			
UNet	Original params: 0.8M	VA	VA	0.7	0.54	0.73	0.68	0.65	0.5	0.67	0.67	0.4	0.29	0.38	0.55	0.65	0.48	0.53	0.84	0.45	0.68	0.62
UNet	Resnet-18 14.3M	VA	VA	0.73	0.57	0.72	0.74	0.67	0.51	0.67	0.67	0.54	0.37	0.76	0.42	0.7	0.54	0.79	0.63	0.5	0.7	0.66
UNet	Resnet-101 51.5M	VA	VA	0.73	0.58	0.73	0.74	0.68	0.51	0.68	0.68	0.58	0.41	0.67	0.51	0.71	0.56	0.75	0.68	0.51	0.71	0.68
UNet	Xception 29M	VA	VA	0.74	0.58	0.72	0.76	0.68	0.51	0.69	0.67	0.6	0.43	0.76	0.5	0.74	0.58	0.85	0.65	0.53	0.71	0.69
UNet	InceptionResnetV2 62M	VA	VA	0.76	0.62	0.74	0.79	0.71	0.55	0.73	0.69	0.68	0.52	0.77	0.61	0.75	0.6	0.85	0.68	0.57	0.72	0.73
DeeplabV3	Resnet-18 16M	VA	VA	0.72	0.56	0.7	0.74	0.66	0.49	0.67	0.64	0.57	0.4	0.67	0.5	0.68	0.51	0.72	0.64	0.49	0.69	0.66
DeeplabV3	Resnet-101 59M	VA	VA	0.73	0.58	0.72	0.74	0.68	0.52	0.69	0.67	0.49	0.32	0.48	0.5	0.66	0.49	0.67	0.65	0.48	0.7	0.64
PAN	Resnet-18 11M	VA	VA	0.75	0.6	0.71	0.8	0.68	0.51	0.72	0.64	0.55	0.38	0.94	0.39	0.76	0.61	0.8	0.72	0.53	0.72	0.69
PAN	Resnet-101 43M	VA	VA	0.75	0.6	0.72	0.78	0.68	0.5	0.7	0.67	0.54	0.37	0.86	0.4	0.67	0.5	0.79	0.57	0.5	0.71	0.66

best overall accuracy at 74% (UNet-inceptionResnetV2, 62 million parameters). Each model performs relatively well at predicting bulkhead and riprap features, with significantly less variance between the models than observed in the models trained and validated based on the Virginia basemap and Massachusetts 3-band orthoimagery. Of note is that the PAN-Resnet18 model achieves 94% and 80% precision in groins and breakwater, despite the relatively small number of labeled cases available (see Fig.2). Fig. 8 summarizes the precision improvement for each class when contrasting models fit in Virginia using the VBMP orthoimagery and NAIP imagery.

To better understand the contribution of the NIR band in NAIP imagery to model performance compared to using only RGB orthoimagery, we further trained each model with 3-band (RGB) NAIP imagery - excluding the fourth

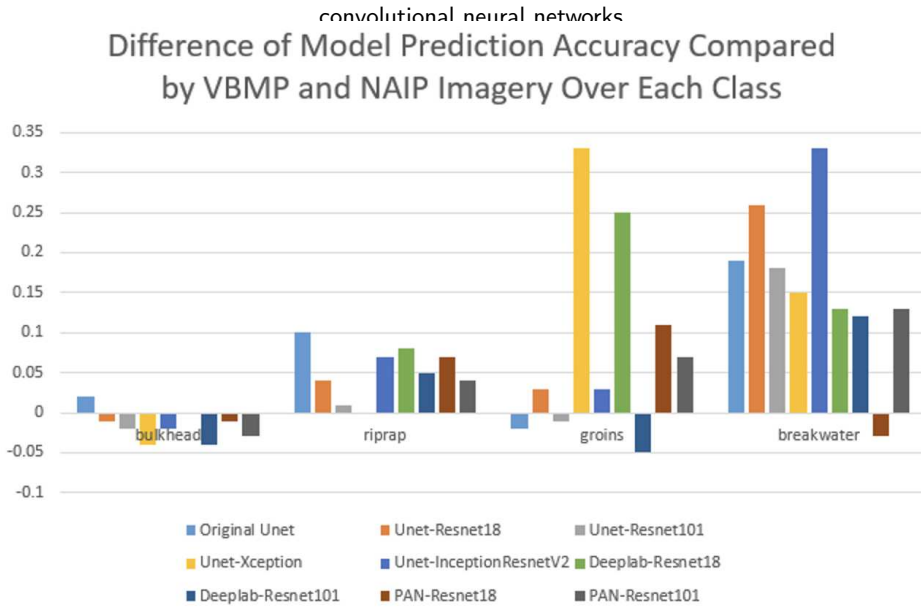


Figure 8: Performance comparison between models trained on VBMP and NAIP

The values in this chart are calculated by subtracting the model prediction accuracy of each class trained on VBMP imagery from the prediction accuracy trained on NAIP imagery. Positive values correspond to improvement after using NAIP imagery.

Table 4

Model performance using 3-band NAIP imagery (excluding near infrared).

Model Details		Dataset Name		Bulkhead				Riprap				Groins				Breakwater				mIOU	OA	Fa-macro
Architecture	Decoder	Trained	Tested	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.			
UNet	Resnet-18	RGB	RGB	0.7	0.54	0.71	0.7	0.66	0.49	0.64	0.67	0.59	0.42	0.83	0.46	0.69	0.53	0.79	0.61	0.5	0.68	0.66
UNet	Resnet-101	RGB	RGB	0.72	0.56	0.7	0.73	0.66	0.5	0.67	0.66	0.48	0.31	0.63	0.38	0.65	0.48	0.75	0.58	0.46	0.69	0.63
UNet	Xception	RGB	RGB	0.74	0.59	0.72	0.76	0.69	0.52	0.7	0.68	0.62	0.45	0.86	0.48	0.75	0.6	0.86	0.67	0.54	0.72	0.7
UNet	InceptionResnetV2	RGB	RGB	0.76	0.61	0.74	0.77	0.71	0.55	0.71	0.71	0.7	0.53	0.87	0.58	0.73	0.57	0.85	0.64	0.57	0.72	0.72
DeeplabV3	Resnet-18	RGB	RGB	0.71	0.55	0.69	0.73	0.64	0.47	0.65	0.63	0.53	0.36	0.65	0.45	0.68	0.52	0.75	0.63	0.47	0.67	0.64
DeeplabV3	Resnet-101	RGB	RGB	0.74	0.59	0.73	0.76	0.69	0.52	0.7	0.68	0.54	0.37	0.63	0.48	0.68	0.51	0.74	0.62	0.5	0.71	0.66
PAN	Resnet-18	RGB	RGB	0.74	0.59	0.71	0.78	0.67	0.51	0.71	0.64	0.6	0.43	0.89	0.45	0.74	0.59	0.81	0.69	0.53	0.71	0.69
PAN	Resnet-101	RGB	RGB	0.73	0.58	0.73	0.73	0.68	0.52	0.68	0.68	0.54	0.37	0.69	0.45	0.67	0.5	0.69	0.65	0.49	0.71	0.66

infrared band (see table 4). Compared to table 3, overall accuracy varied by 1% to 2%, depending on the model. Models with complex structures such as UNet-xception and DeeplabV3-Resnet101 tended to have a decrease in accuracy. The change in model performance for each architecture varies, with a small decrease in precision on groins in the more complex model architecture. However, when using PAN-resnet18, the prediction accuracy on groins increased to 94%.

4. Discussion

The overall results of this study illustrate that, leveraging NAIP 4-band NIR imagery, classification accuracy of bulkhead, riprap, groins and breakwater shoreline structures can reach approximately 74% in the Virginia study area; accuracy of up to 83% was observed in the models pooling Virginia and Massachussets shoreline structure data. Notably, this accuracy is a pixel-level metric - i.e., the number of approximately 60 centimeter pixels that are classified correctly. While higher levels of accuracy may be required for a fully automated approach, by integrating this into

pyShore: A deep learning toolkit for shoreline structure mapping with high-resolution orthographic imagery and

convolutional neural networks

a semi-automated pipeline in which first the presented algorithm is applied, and second a human coder corrects the output, significant time savings can be gained as contrasted to the multiple-year process currently being used by state and federal agencies. In this section, we explore model performance, highlight a number of directions for future research, and introduce a new ArcGIS python based toolkit that enables practitioners to leverage these algorithms in a semi-automated fashion.

4.1. Feature Detection

One frequent criticism of deep learning models is their black-box approach to image recognition, in which it can be difficult to ascertain why the model is performing (or failing) at a given task. To explore the mechanisms driving the presented models, we apply a SHAP visualization (SHapley Additive exPlanations visualization technique (Lundberg and Lee, 2017)) to understand what factors contribute to the model's capability to distinguish different shoreline features. Figure 9 shows an example of what factors played into the correct classification of two images in the ResNet-18 based network. In Figure 9A, an example of riprap is shown in the first row, and to the right are the features that were important for a given classification - i.e., the correct classification (riprap) was due to features shown in red on the image map. As this image shows, the algorithm leverages feature information spatially proximate to the class of interest. Conversely, the final column of Figure 9A also shows that indicators of riprap contribute negatively to classifications of groins - i.e., the algorithm can clearly distinguish between riprap and groin features. A more complex example is shown in Figure 9B, in which the true class (breakwater) is identified, but the information used for distinguishing between breakwater and groin is more diffuse - i.e., contextual information is being leveraged, rather than only pixels explicitly containing the breakwater feature.

4.2. Computational Efficiency as a product of Imagery Source & Bands

Our findings suggest that relatively less complex networks are some of the strongest performers for the task of shoreline feature detection. This is shown by the strong performance of PAN-Resnet18 in the NAIP tests, providing one of the highest levels of accuracy 72% in the Virginia case despite having one of the lowest number (11 million) of parameters to fit. While the other network architectures were able to achieve similar levels of accuracy in the context of a fourth band (UNet-InceptionResnetV2 achieved 72% accuracy), it does so with more model overhead (62 million parameters). This suggests that PAN-Resnet18 based implementations provide a strong balance between accuracy and computational costs for end-user deployment.

However, if the user only has 3-band information available to them, UNet-xception may provide the best combination of accuracy and complexity - providing similar levels of accuracy (82% in the MA/VA joint case) at a much lower parameter count than the similarly performing UNet-inceptionresnetV2 (29M vs. 62M in UNet-InceptionresnetV2). In terms of practical implementation, we find that our fully fit PAN-resnet18 model can be deployed with a memory

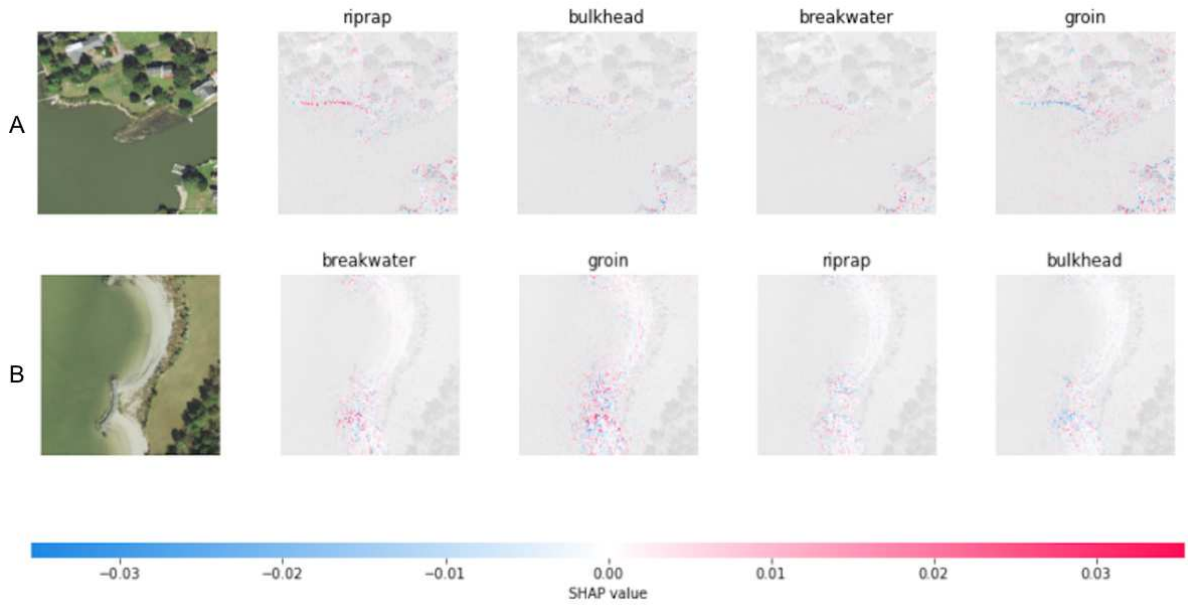


Figure 9: Virginia shoreline structure prediction analysis.

The above shows two random correctly classified test images in the first column. The second, third, fourth, and fifth columns show the pixels/features that contributed against and in favor of prediction for each of the four classes. For example, the upper-left image is a riprap structure that was predicted by the model most likely to be a riprap, then bulkhead, breakwater, and groin. Blue pixels represent areas that work against classification in a given class and the red pixels represent areas that work for classification. In this case, the Resnet-18 based network for image classification was investigated.

footprint of only 43.4 megabytes, and an estimate for 1.5 kilometers of shoreline can be generated in between 1.4 seconds (GPU) and 2.16 seconds (CPU), indicating it is suitable for general use in desktop environments. For users that do not have a near-infrared band available, the UNet-xception implementation requires 110 megabytes of memory to load, and can generate estimates for 1.5 kilometers of shoreline in between 2 seconds (GPU) and 9.52 seconds (CPU), see Table 5 for computational time tested using 3-band VBMP imagery. This highlights the importance of identifying the fourth band of information to enhance efficiency, but we also note that both models are reasonable to consider for deployment relative to the decades-long time periods fully manual workflows require.

4.3. Difference in Shoreline Structure Data Sources

When comparing groin structures from the VA CUSP and MA data, we find that the features in MA are consistently labeled as perpendicular to the shoreline; this contrasts to the CUSP data, in which many groin features are coded along the shoreline and difficult to distinguish from other structures by eye (see Figure 10). Additionally, we found that digitization in the VA dataset is frequently not fully aligned with the structures visible in imagery (see Figure 11). These differences result in a number of limitations - and future directions - for this study. Because the Virginia data

Table 5

A summary of computational time when making the 5 km prediction with 3-band VBMP imagery

Model Name	Computational Time (seconds)
UNet-Resnet18	5.27
UNet-Resnet101	13.07
UNet-xception	9.52
UNet-InceptionResnetV2	15.12
DeeplabV3-Resnet18	11.59
DeeplabV3-Resnet101	36.89
PAN-Resnet18	3.66
PAN-Resnet101	11.27

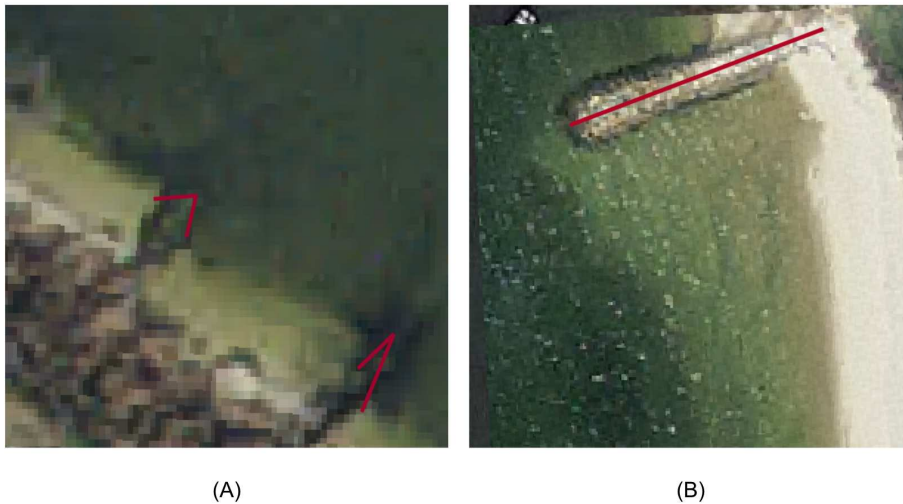


Figure 10: A comparison of geocoded groins from VA CUSP and MA data. (A) shows the groins (red lines) from VA CUSP overlaid with VBMP imagery, and (B) shows the groins from MA data overlaid with MA orthoimagery.

is not directly aligned in all cases, the three-meter buffer applied may have been insufficient to capture the contextual pixels required to accurately classify all cases; this is in contrast to Massachusetts (Figure 10B) where such errors did not exist across most observations. Because the overall model accuracy was significantly higher (up to 83% overall accuracy) when Massachusetts data was included, it is feasible that our validation is understating the overall accuracy of the model in Virginia simply because our validation and calibration data itself has apparent errors. Future testing across separately digitized datasets could help to test this hypothesis; alternatively, work to correct the errors across the VA CUSP dataset could bear useful fruit for this and other similar initiatives.

4.4. Scale of Classification

The primary unit of classification in this work was the pixel - i.e., we sought to train an algorithm to take in an image tile, and output which pixels belonged to each of four classes. In practice, this level of spatial precision is higher than what may be required for some tasks: i.e., if one has a shoreline defined by a line feature, one could break that shoreline up with an object-based strategy, and then classify each resultant subset of the shoreline as belonging to



Figure 11: Example of features from VA CUSP not fully aligned with the imagery. On the left is a riprap, and on the right is two breakwater structures from VA CUSP.

a given class (rather than classify each pixel). More broadly, many small errors - especially errors on the landward side of a given coastal feature - may not negatively influence decision making. Future work exploring the semantic definition of shoreline, and how that may interrelate with the appropriate scale and unit of observation for automated approaches to classification, could provide a fruitful pathway forward.

4.5. pyShore: A semi-automated toolkit for shoreline structure classification in ArcGIS

A core motivation for this work is to mitigate the total amount of time required to digitize vast amounts of shoreline; as noted in our introduction, today shoreline mapping processes take around a decade (CCRM, 2019; Fontenault et al., 2013). While future iterations of the work presented here may enable a fully automated pipeline for shoreline feature detection, the accuracy of the presented approach renders it best suited to a semi-automated approach in which human experts correct the output of the automated procedures. To enable this, here we present *pyShore*, a semi-automated toolkit for shoreline structure classification in the ArcGIS Pro (ESRI, 2022) software package.

This tool is designed to take in two files - imagery and a definition of shoreline - and output for each pixel in the shoreline a best estimate of the type of shoreline feature present. Using PAN-Resnet18 as an example, these files are passed into the pre-trained PAN-Resnet18 model presented in this work, specifically the best performing NAIP 4-band model with 72% overall accuracy.¹ The user must set a single parameter - buffer distance.

The full workflow of the tool follows a multiple step procedure in which (see Fig. 12):

1. The user defines a working folder (F) in which to save all processing and result data.
2. The user defines a folder with all source imagery in a georeferenced *geoTiff* format (F_{img}).
3. The user defines a single geometry file (i.e., shapefile) that defines the geographic locations of shoreline within

¹Of note, depending on the spatial resolution of input imagery, for some users it may make sense to implement one of the VA or MA orthoimagery pre-trained models. We provide an optional download of the best performing weights for these data to enable such an analysis.

the provided imagery (S).

4. The user provides a buffer distance (B), or the distance from the shoreline that should be extracted from the imagery to serve as the target for the classification.
5. All images in folder *Fimg* are cut into non-overlapping 256×256 patches (defined as P) suitable for input into the trained PAN model architecture.
6. Taking in all patches P , line S , and buffer B , we identify the image patches in P that have overlap with buffered line S ; image patches with no overlap and dimension smaller than 256×256 are discarded.
7. Each patch is subjected to a forward pass through the Resnet-18-based PAN trained in this paper, creating an output (O) of the same dimensions (256×256) in which each pixel is classified as one of four classes (four shoreline structures).
8. Pixels intersecting with the shoreline feature (S) with buffer distance, B , are extracted from output, O , and then vectorized to provide a final shoreline output.

Figure 12 summarizes the workflow of image processing, prediction, and post-processing. In addition to outputting the shoreline, a metric of confidence is generated to help guide user efforts, i.e., identifying the areas where the algorithm was least confident in its prediction so a human coder may select those areas for editing during a post-processing stage. This confidence level for each line segment (represented as a 3-meter buffered polygon) is generated by:

$$\text{Confidence} = \frac{1}{n} \sum_{i=1}^n \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (8)$$

where X_i is the absolute score the PAN-Resnet18 estimates for the class a given pixel (i) was predicted as. $X_{\min}$ and $X_{\max}$ are the smallest and largest such estimate across all pixels in a target region. To calculate the confidence value for a given 3-meter buffered polygon, we define n as the set of pixels a buffered polygon overlaps. The confidence value for any target polygon is defined as the average confidence value of pixels in the layer that overlap with the polygon.

4.6. Contributions to the Literature

The work presented in this paper provides four core contributions. First, we introduce the first application of shoreline structure classification and mapping using deep learning techniques with pixel-scale localization, illustrating that accuracy of up to 83% is possible with contemporary deep learning techniques. Second, we provide some of the first evidence of the value of near-infrared spectral data in the context of deep learning for shoreline feature identification, specifically finding that NIR information can help reduce the computational complexity of algorithms required for generating sufficiently accurate estimates for use in semi-automated workflows. Third, we introduce a toolkit to employ a deep learning framework within the ArcGIS Pro environment to facilitate rapid shoreline feature identification. Fourth,

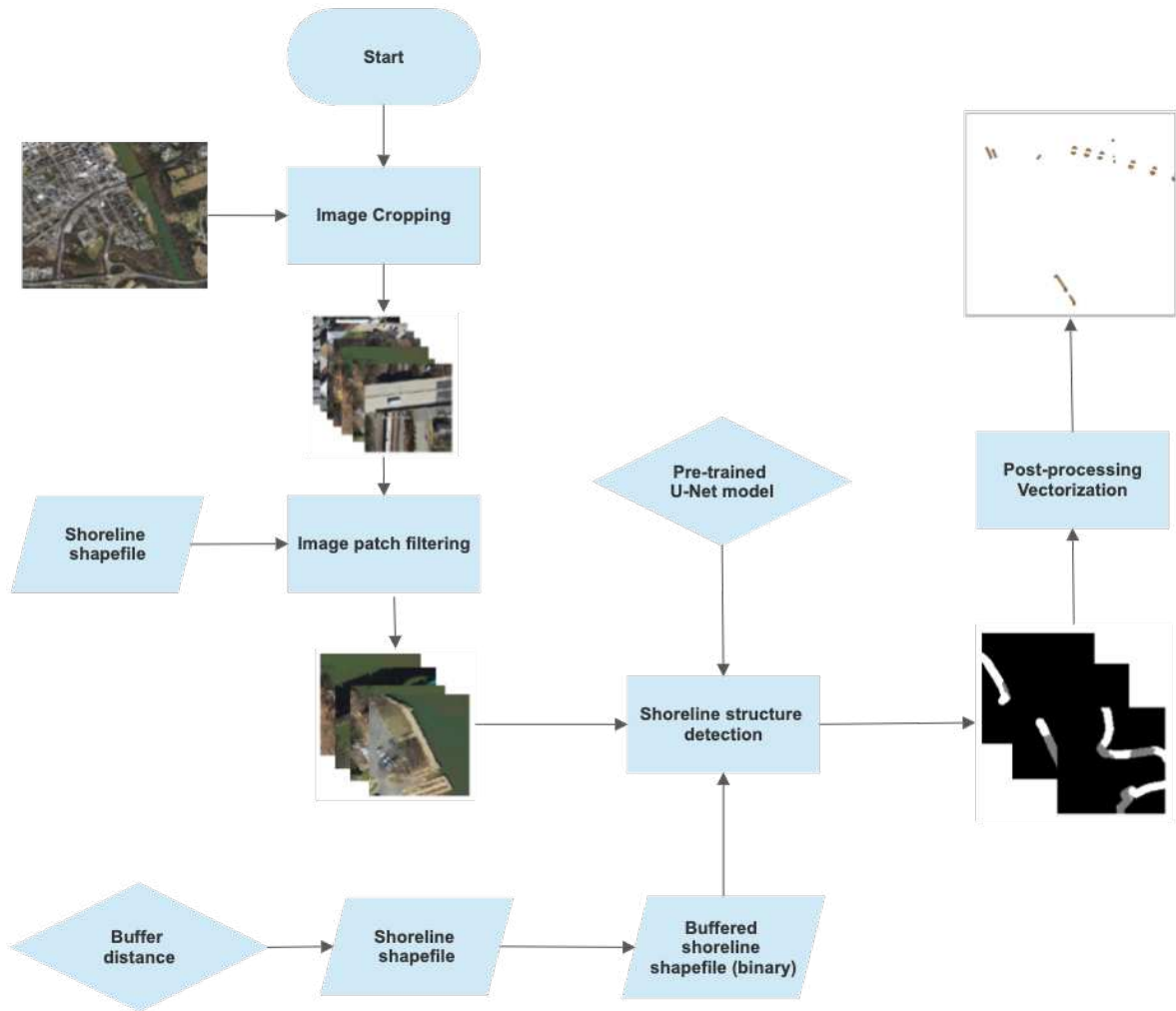


Figure 12: Workflow of shoreline structure detection using the pyShore ArcGIS Pro toolkit.

from a shoreline management and sedimentation modeling perspective, the generated shoreline structure locations and types would be used as input for a shoreline management model (Nunez et al., 2022) which provides recommendations for the best management practices for defended and undefended shorelines. These contributions will enable more rapid digitization of shoreline features than was previously possible, aiding in increasing the rate at which shoreline maps can be generated and updated.

5. Conclusion

Accurate and precise low-cost shoreline structure mapping has the potential to improve the baseline information supporting shoreline management, thus improving the decision-making capacity of local, state, and national governments. Today, shoreline structure mapping is a challenging task due to requirements for in-situ data collection, manual

pyShore: A deep learning toolkit for shoreline structure mapping with high-resolution orthographic imagery and

convolutional neural networks

image digitization, and the inevitability of concurrent shoreline change during years-long production processes.

In this paper, we sought to explore the capability of convolutional neural network architectures to identify shoreline structures from remotely sensed imagery. This study provides (1) an initial benchmark accuracy (72%) for deep learning-based shoreline structure localization in Virginia (82% across a combination of Virginia and Massachusetts datasets), and (2) a computationally efficient toolkit that can be deployed in desktop environments which harnesses the proposed method for applied use. These findings, and the related toolkit provide a new method for local and state governments in the United States to generate shoreline inventories, and improve the management of coastal resources and infrastructure.

6. Acknowledgments

The authors acknowledge William & Mary Research Computing for providing computational resources and technical support that have contributed to the results reported within this paper. URL: <https://www.wm.edu/it/rc>.

pyShore: A deep learning toolkit for shoreline structure mapping with high-resolution orthographic imagery and
convolutional neural networks

Code availability section

Name of the code/library

Contact: zlv@wm.edu

Hardware requirements: Python 3.7.10

Program language: Python

Software required: ArcGIS Pro 2.3.0

Github size: 107MB

The source codes are available for downloading at the link: <https://github.com/MirandaLv/pyShore>

References

- An, Y., Guo, J., Ye, Q., Childs, C., Walsh, J., Dong, R., 2021. Deep convolutional neural network for automatic fault recognition from 3d seismic datasets. *Computers & Geosciences* 153, 104776. URL: <https://www.sciencedirect.com/science/article/pii/S0098300421000807>, doi:<https://doi.org/10.1016/j.cageo.2021.104776>.
- Aravena Pelizari, P., Geiß, C., Aguirre, P., Santa María, H., Merino Peña, Y., Taubenböck, H., 2021. Automated building characterization for seismic risk assessment using street-level imagery and deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing* 180, 370–386. URL: <https://www.sciencedirect.com/science/article/pii/S0924271621001817>, doi:<https://doi.org/10.1016/j.isprsjprs.2021.07.004>.
- Ayala, C., Sesma, R., Aranda, C., Galar, M., 2021. A deep learning approach to an enhanced building footprint and road detection in high-resolution satellite imagery. *Remote Sensing* 13. URL: <https://www.mdpi.com/2072-4292/13/16/3135>, doi:10.3390/rs13163135.
- Bagli, S., Soille, P., 2004. Automatic delineation of shoreline and lake boundaries from landsat satellite images. *Proceedings of initial ECO-IMAGINE GI and GIS for Integrated Coastal Management*.
- Berman, M., Hershner, C., 1999. Development of guidelines for generating shoreline situation reports establishing protocols for data collection and dissemination. Available online: http://ccrm.vims.edu/gis_data_maps/shoreline_inventories/shoreline_situation.pdf.
- Berman, M., Hershner, C., Angst, K., Killeen, S., Nunez, K., Rudnick, T., Schatt, D., Stanhope, D., Weiss, D., 2013. Delaware shoreline inventory: Rehoboth bay, comprehensive coastal inventory program. Center for Coastal Resources Management Virginia Institute of Marine Science, College of William and Mary, Gloucester Point, Virginia, 23062. <https://www.vims.edu/ccrm/research/inventory/delaware/index.php>.
- Bhil, K., Shindihatti, R., Mirza, S., Latkar, S., Ingle, Y.S., Shaikh, N.F., Prabu, I., Pardeshi, S.N., 2022. Recent progress in object detection in satellite imagery: A review, in: Aurelia, S., Hiremath, S.S., Subramanian, K., Biswas, S.K. (Eds.), *Sustainable Advanced Computing*, Springer Singapore, Singapore. pp. 209–218.
- Bhosle, K., Musande, V., 2019. Evaluation of deep learning cnn model for land use land cover classification and crop identification using hyperspectral remote sensing images. *Journal of the Indian Society of Remote Sensing* 47, 1949–1958. URL: <https://doi.org/10.1007/s12524-019-01041-2>, doi:10.1007/s12524-019-01041-2.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32. doi:10.1023/A:1010950718922.
- Brewer, E., Lin, J., Kemper, P., Hennin, J., Runfola, D., 2021. Predicting road quality using high resolution satellite imagery: A transfer learning approach. *PLOS ONE* 16, 1–18. URL: <https://doi.org/10.1371/journal.pone.0253370>, doi:10.1371/journal.pone.0253370.

- 523 Brewer, E., Lin, J., Runfola, D., 2022. Susceptibility & defense of satellite image-trained convolutional networks to backdoor attacks. *Information*
- 524 *Sciences* 603, 244–261. doi:10.1016/J.INS.2022.05.004.
- 525 CCRM, 2019. Virginia shoreline inventory database. Center for Coastal Resources Management, Virginia Institute of Marine Science, College of
- 526 William and Mary, Gloucester Point. <http://www.vims.edu/ccrm/research/inventory/index.php>.
- 527 Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2016. Deeplab: Semantic image segmentation with deep convolutional nets,
- 528 atrous convolution, and fully connected crfs. URL: <https://arxiv.org/abs/1606.00915>, doi:10.48550/ARXIV.1606.00915.
- 529 Chen, X., Xiang, S., Liu, C.L., Pan, C.H., 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience*
- 530 *and Remote Sensing Letters* 11, 1797–1801. doi:10.1109/LGRS.2014.2309695.
- 531 Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing* 117,
- 532 11–28. URL: <https://www.sciencedirect.com/science/article/pii/S0924271616300144>, doi:[https://doi.org/10.1016/j.](https://doi.org/10.1016/j.isprsjprs.2016.03.014)
- 533 [isprsjprs.2016.03.014](https://doi.org/10.1016/j.isprsjprs.2016.03.014).
- 534 Di, K., Wang, J., Ma, R., Li, R., 2003. Automatic shoreline extraction from high resolution ikonos satellite imagery. *ASPRS Annual Conference*
- 535 *Proceedings*.
- 536 DigitalGlobe, 2018. Digitalglobe. Retrieved from: <https://www.digitalglobe.com/>.
- 537 Du, J., Shen, J., Bilkovic, D.M., Hershner, C.H., Sisson, M., 2016. A numerical modeling approach to predict the effect of a storm surge barrier
- 538 on hydrodynamics and long-term transport processes in a partially mixed estuary. *Estuaries and Coasts* 2016 40:2 40, 387–403. URL: [https://](https://link.springer.com/article/10.1007/s12237-016-0175-0)
- 539 link.springer.com/article/10.1007/s12237-016-0175-0, doi:10.1007/S12237-016-0175-0.
- 540 Eboigbe, M.A., Ugwuoti, A.I., Sam, A.A., 2020. The effectiveness of the semi-automatic technique in extracting shoreline from digital elevation
- 541 models for sustainable development in africa. *European Journal of Environment and Earth Sciences* 1(4). URL: [https://www.ej-geo.org/](https://www.ej-geo.org/index.php/ejgeo/article/view/48)
- 542 [index.php/ejgeo/article/view/48](https://www.ej-geo.org/index.php/ejgeo/article/view/48), doi:10.24018/ejgeo.2020.1.4.48.
- 543 Enwright, N.M., Griffith, K.T., Osland, M.J., 2016. Barriers to and opportunities for landward migration of coastal wetlands with sea-level rise.
- 544 *Frontiers in Ecology and the Environment* 14, 307–316. doi:10.1002/FEE.1282/FULLPDF.
- 545 ESRI, 2022. ArcGIS Pro 2.3.0. Redlands, CA: Environmental Systems Research Institute.
- 546 Etten, A.V., Hogan, D., Martinez-Manso, J., Shermeyer, J., Weir, N., Lewis, R., 2021. The multi-temporal urban development spacenet dataset.
- 547 *CoRR* abs/2102.04420. URL: <https://arxiv.org/abs/2102.04420>, arXiv:2102.04420.
- 548 Fontenault, J., Vinhateiro, N., Knee, K., 2013. Mapping and analysis of privately-owned coastal structures along the massachusetts shoreline.
- 549 [https://www.mass.gov/doc/mapping-and-analysis-of-privately-owned-coastal-structures-along-the-ma-shoreline/](https://www.mass.gov/doc/mapping-and-analysis-of-privately-owned-coastal-structures-along-the-ma-shoreline/download)
- 550 [download](https://www.mass.gov/doc/mapping-and-analysis-of-privately-owned-coastal-structures-along-the-ma-shoreline/download).
- 551 Ghorai, D., Mahapatra, M., 2020. Extracting shoreline from satellite imagery for gis analysis. *Remote Sens Earth Syst Sci* 3, 13–22. URL:
- 552 <https://doi.org/10.1007/s41976-019-00030-w>.
- 553 Gittman, R.K., Fodrie, F.J., Popowich, A.M., Keller, D.A., Bruno, J.F., Currin, C.A., Peterson, C.H., Piehler, M.F., 2015. Engineer-
- 554 ing away our natural defenses: an analysis of shoreline hardening in the us. *Frontiers in Ecology and the Environment* 13, 301–307.
- 555 URL: <https://onlinelibrary.wiley.com/doi/full/10.1890/150065>[https://onlinelibrary.wiley.com/doi/abs/10.1890/](https://onlinelibrary.wiley.com/doi/abs/10.1890/150065)
- 556 [150065](https://onlinelibrary.wiley.com/doi/abs/10.1890/150065)<https://esajournals.onlinelibrary.wiley.com/doi/10.1890/150065>, doi:10.1890/150065.
- 557 GLAD, 2018. Landsat analysis ready data. Accessed on July 2022: <https://glad.umd.edu/>.
- 558 Gong, P., Wang, J., Yu, L., Zhao, Y., Zhao, Y., Liang, L., Niu, Z., Huang, X., Fu, H., Liu, S., Li, C., Li, X., Fu, W., Liu, C., Xu, Y., Wang, X., Cheng,
- 559 Q., Hu, L., Yao, W., Zhang, H., Zhu, P., Zhao, Z., Zhang, H., Zheng, Y., Ji, L., Zhang, Y., Chen, H., Yan, A., Guo, J., Yu, L., Wang, L., Liu, X.,
- 560 Shi, T., Zhu, M., Chen, Y., Yang, G., Tang, P., Xu, B., Giri, C., Clinton, N., Zhu, Z., Chen, J., Chen, J., 2013. Finer resolution observation and

- 561 monitoring of global land cover: first mapping results with landsat tm and etm+ data. *International Journal of Remote Sensing* 34, 2607–2654.
562 URL: <https://www.tandfonline.com/doi/full/10.1080/01431161.2012.748992>, doi:10.1080/01431161.2012.748992.
- 563 Goodman, S., BenYishay, A., Runfola, D., 2021. A convolutional neural network approach to predict non-permissive environments from moderate-
564 resolution imagery. *Transactions in GIS* 25, 674–691. URL: <https://onlinelibrary.wiley.com/doi/10.1111/tgis.12661>, doi:10.
565 1111/tgis.12661.
- 566 Guirado, E., Tabik, S., Alcaraz-Segura, D., Cabello, J., Herrera, F., 2017. Deep-learning versus obia for scattered shrub detection with google
567 earth imagery: Ziziphus lotus as case study. *Remote Sensing* 9, 1220. URL: <http://www.mdpi.com/2072-4292/9/12/1220>, doi:10.3390/
568 rs9121220.
- 569 Guirado, E., Tabik, S., L. Rivas, M., Alcaraz-Segura, D., Herrera, F., 2019. Whale counting in satellite and aerial images with deep learning.
570 *Scientific Reports* 9. doi:10.1038/s41598-019-50795-9.
- 571 He, K., Gkioxari, G., Dollár, P., Girshick, R.B., 2017. Mask R-CNN. *CoRR* abs/1703.06870. URL: <http://arxiv.org/abs/1703.06870>,
572 [arXiv:1703.06870](https://arxiv.org/abs/1703.06870).
- 573 Hill, K., 2015. Coastal infrastructure: a typology for the next century of adaptation to sea-level rise. *Frontiers in Ecology and the Environment* 13,
574 468–476. URL: <https://onlinelibrary.wiley.com/doi/full/10.1890/150088>[https://onlinelibrary.wiley.com/doi/abs/](https://onlinelibrary.wiley.com/doi/abs/10.1890/150088)
575 [10.1890/150088](https://onlinelibrary.wiley.com/doi/abs/10.1890/150088)<https://esajournals.onlinelibrary.wiley.com/doi/10.1890/150088>, doi:10.1890/150088.
- 576 Kikaki, K., Kakogeorgiou, I., Mikeli, P., Raitzos, D.E., Karantzalos, K., 2022. Marida: A benchmark for marine debris detection from sentinel-2
577 remote sensing data. *PLOS ONE* 17, e0262247. URL: [https://journals.plos.org/plosone/article?id=10.1371/journal.pone.](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0262247)
578 [0262247](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0262247), doi:10.1371/JOURNAL.PONE.0262247.
- 579 Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. *Neural Information Processing*
580 *Systems* 25. doi:10.1145/3065386.
- 581 Kuleli, T., Guneroglu, A., Karsli, F., Dihkan, M., 2011. Automatic detection of shoreline change on coastal ramsar wetlands of turkey. *Ocean En-*
582 *gineering* 38, 1141–1149. URL: <https://www.sciencedirect.com/science/article/pii/S0029801811000874>, doi:[https://doi.](https://doi.org/10.1016/j.oceaneng.2011.05.006)
583 [org/10.1016/j.oceaneng.2011.05.006](https://doi.org/10.1016/j.oceaneng.2011.05.006).
- 584 Kumar, L., Afzal, M.S., Afzal, M.M., 2020. Mapping shoreline change using machine learning: a case study from the eastern indian coast. *Acta*
585 *Geophysica* , 1–17.
- 586 Li, H., Wang, C., Cui, Y., Hodgson, M., 2021. Mapping salt marsh along coastal south carolina using u-net. *ISPRS Journal of Photogrammetry and*
587 *Remote Sensing* 179, 121–132. doi:10.1016/J.ISPRSJPRS.2021.07.011.
- 588 Li, H., Xiong, P., An, J., Wang, L., 2018. Pyramid attention network for semantic segmentation. URL: <https://arxiv.org/abs/1805.10180>,
589 doi:10.48550/ARXIV.1805.10180.
- 590 Liu, H., Wang, L., Sherman, D.J., Wu, Q., Su, H., 2011. Algorithmic foundation and software tools for extracting shoreline features from remote
591 sensing imagery and lidar data. *Journal of Geographic Information System*. Vol.3, No.2.
- 592 Liu, M., Fu, B., Xie, S., He, H., Lan, F., Li, Y., Lou, P., Fan, D., 2021. Comparison of multi-source satellite images for classifying marsh vegetation
593 using deeplabv3 plus deep learning algorithm. *Ecological Indicators* 125, 107562. URL: [https://www.sciencedirect.com/science/](https://www.sciencedirect.com/science/article/pii/S1470160X21002272)
594 [article/pii/S1470160X21002272](https://www.sciencedirect.com/science/article/pii/S1470160X21002272), doi:<https://doi.org/10.1016/j.ecolind.2021.107562>.
- 595 Loos, E., Niemann, K., 2002. Shoreline feature extraction from remotely-sensed imagery, in: *IEEE International Geoscience and Remote Sensing*
596 *Symposium*, pp. 3417–3419 vol.6. doi:10.1109/IGARSS.2002.1027201.
- 597 Lundberg, S., Lee, S.I., 2017. A unified approach to interpreting model predictions. URL: <https://arxiv.org/abs/1705.07874>, doi:10.
598 [48550/ARXIV.1705.07874](https://arxiv.org/abs/1705.07874).

- 599 Mason, D., Davenport, I., 1996. Accurate and efficient determination of the shoreline in ers-1 sar images. *IEEE Transactions on Geoscience and*
- 600 *Remote Sensing* 34, 1243–1253. doi:10.1109/36.536540.
- 601 McAllister, E., Payo, A., Novellino, A., Dolphin, T., Medina-Lopez, E., 2022a. Multispectral satellite imagery and machine learning for the
- 602 extraction of shoreline indicators. *Coastal Engineering* 174, 104102. URL: [https://www.sciencedirect.com/science/article/pii/](https://www.sciencedirect.com/science/article/pii/S0378383922000229)
- 603 [S0378383922000229](https://www.sciencedirect.com/science/article/pii/S0378383922000229), doi:<https://doi.org/10.1016/j.coastaleng.2022.104102>.
- 604 McAllister, E., Payo, A., Novellino, A., Dolphin, T., Medina-Lopez, E., 2022b. Multispectral satellite imagery and machine learning for the
- 605 extraction of shoreline indicators. *Coastal Engineering* 174, 104–102. URL: [https://www.sciencedirect.com/science/article/pii/](https://www.sciencedirect.com/science/article/pii/S0378383922000229)
- 606 [S0378383922000229](https://www.sciencedirect.com/science/article/pii/S0378383922000229), doi:<https://doi.org/10.1016/j.coastaleng.2022.104102>.
- 607 MGS, 2000. High resolution shoreline map data for tidewater maryland. Maryland Geological Survey [http://www.mgs.md.gov/coastal_](http://www.mgs.md.gov/coastal_geology)
- 608 [geology](http://www.mgs.md.gov/coastal_geology).
- 609 Najjar, R.G., Pyke, C.R., Adams, M.B., Breitburg, D., Hershner, C., Kemp, M., Howarth, R., Mulholland, M.R., Paolisso, M., Secor, D., Sell-
- 610 ner, K., Wardrop, D., Wood, R., 2010. Potential climate-change impacts on the chesapeake bay. *Estuarine, Coastal and Shelf Science* 86,
- 611 1–20. URL: <https://www.sciencedirect.com/science/article/pii/S0272771409004582>, doi:[https://doi.org/10.1016/j.](https://doi.org/10.1016/j.ecss.2009.09.026)
- 612 [ecss.2009.09.026](https://doi.org/10.1016/j.ecss.2009.09.026).
- 613 Narayan, A., Tuci, E., Labrosse, F., Alkilabi, M.H.M., 2017. Road detection using convolutional neural networks. *Proceedings of the 14th Euro-*
- 614 *pean Conference on Artificial Life ECAL 2017* , 314–321URL: https://www.mitpressjournals.org/doi/abs/10.1162/isal_a_053,
- 615 doi:10.7551/ecal_a_053.
- 616 Niculescu, S., Bengoufa, S., Mihoubi, M., Belkessa, R., Abbad, K., 2021. ROCKY SHORELINE EXTRACTION USING A DEEP
- 617 LEARNING MODEL AND OBJECT-BASED IMAGE ANALYSIS. *International Archives of the Photogrammetry, Remote Sensing*
- 618 *and Spatial Information Sciences XLIII-B3-2021*, 23–29. URL: <https://hal.archives-ouvertes.fr/hal-03274256>, doi:10.5194/
- 619 [isprs-archives-XLIII-B3-2021-23-2021](https://hal.archives-ouvertes.fr/hal-03274256).
- 620 NOAA, 2018. Sentinel. Retrieved from: <https://sentinel.esa.int/web/sentinel/home>.
- 621 NOAA, 2021a. Noaa continually updated shoreline product (cusp). Accessed on March 2022: [https://shoreline.noaa.gov/data/](https://shoreline.noaa.gov/data/datasheets/cusp.html)
- 622 [datasheets/cusp.html](https://shoreline.noaa.gov/data/datasheets/cusp.html).
- 623 NOAA, 2021b. Noaa’s continually updated shoreline product (cusp). Accessed on March 2022: [https://www.ngs.noaa.gov/INFO/](https://www.ngs.noaa.gov/INFO/OnePagers/CUSP_One-Pager.pdf)
- 624 [OnePagers/CUSP_One-Pager.pdf](https://www.ngs.noaa.gov/INFO/OnePagers/CUSP_One-Pager.pdf).
- 625 Nunez, K., Rudnick, T., Mason, P., Tombleson, C., Berman, M., 2022. A geospatial modeling approach to assess site suitability of living shorelines
- 626 and emphasize best shoreline management practices. *Ecological Engineering* 179, 106617. doi:10.1016/J.ECOLENG.2022.106617.
- 627 Nunez, K., Zhang, Y.J., Herman, J., Reay, W., Hershner, C., 2020. A multi-scale approach for simulating tidal marsh evolution. *Ocean*
- 628 *Dynamics* 2020 70:9 70, 1187–1209. URL: <https://link.springer.com/article/10.1007/s10236-020-01380-6>, doi:10.1007/
- 629 [S10236-020-01380-6](https://link.springer.com/article/10.1007/s10236-020-01380-6).
- 630 OCM-Partners, 2022. 2018 virginia naip digital ortho photo imagery. NOAA National Centers for Environmental Information. [https://www.](https://www.fisheries.noaa.gov/inport/item/58386)
- 631 [fisheries.noaa.gov/inport/item/58386](https://www.fisheries.noaa.gov/inport/item/58386).
- 632 Palaseanu-Lovejoy, M., Danielson, J.J., Thatcher, C.A., Foxgrover, A.C., Barnard, P.L., Brock, J., Young, A., 2016. Automatic delineation of
- 633 seaciff limits using lidar-derived high-resolution dems in southern california. *Journal of Coastal Research* , 162–173.doi:10.2112/SI76-014.
- 634 Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., 2016. Enet: A deep neural network architecture for real-time semantic segmentation. *CoRR*
- 635 [abs/1606.02147](https://arxiv.org/abs/1606.02147). URL: <http://arxiv.org/abs/1606.02147>, arXiv:1606.02147.
- 636 Payo, A., Antelo, B.J., Hurst, M., Palaseanu-Lovejoy, M., Williams, C., Jenkins, G., Lee, K., Favis-Mortlock, D., Barkwith, A., Ellis, M.A., 2018.

- Development of an automatic delineation of cliff top and toe on very irregular planform coastlines (cliffmetrics v1.0). *Geosci. Model Dev.*, 11, 4317–4337. doi:<https://doi.org/10.5194/gmd-11-4317-2018>.
- Phillips, S., McGee, B., 2014. The economic benefits of cleaning up the Chesapeake. Technical Report. Chesapeake Bay Program. URL: <https://www.cbf.org/news-media/features-publications/reports/economic-benefits-of-cleaning-up-the-chesapeake-bay/>.
- Prathap, G., Afanasyev, I., 2018. Deep learning approach for building detection in satellite multispectral imagery, in: 2018 International Conference on Intelligent Systems (IS), pp. 461–465. doi:10.1109/IS.2018.8710471.
- Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. URL: <https://arxiv.org/abs/1902.09630>, doi:10.48550/ARXIV.1902.09630.
- Roberts, J., Mwangi, R., Mukabi, F., Njui, J., Nzioka, K., Ndambiri, J., Bispo, P., Espirito-Santo, F., Gou, Y., Johnson, S., Louis, V., Pacheco-Pascagaza, A., Rodriguez-Veiga, P., Tansey, K., Upton, C., Robb, C., Balzter, H., 2022. Pyeo: A python package for near-real-time forest cover change detection from earth observation using machine learning. *Computers & Geosciences* 167, 105192. URL: <https://www.sciencedirect.com/science/article/pii/S0098300422001455>, doi:<https://doi.org/10.1016/j.jageo.2022.105192>.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351, 234–241. URL: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28, doi:10.1007/978-3-319-24574-4_28.
- Runfola, D., 2022. Computational geography. *The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2022 Edition)*, John P. Wilson (Ed.). URL: <https://gistbok.ucgis.org/bok-topics/computational-geography>, doi:10.22224/gistbok/2022.1.7.
- Runfola, D., Anderson, A., Baier, H., Crittenden, M., Dowker, E., Fuhrig, S., Goodman, S., Grimsley, G., Layko, R., Melville, G., Mulder, M., Oberman, R., Panganiban, J., Peck, A., Seitz, L., Shea, S., Slevin, H., Youngerman, R., Hobbs, L., 2020. geoboundaries: A global database of political administrative boundaries. *PLOS ONE* 15, e0231866. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0231866>, doi:10.1371/JOURNAL.PONE.0231866.
- Runfola, D., Baier, H., Mills, L., Naughton-Rockwell, M., Stefanidis, A., 2022. Deep learning fusion of satellite and social information to estimate human migratory flows. *Transactions in GIS* URL: <https://onlinelibrary.wiley.com/doi/full/10.1111/tgis.12953><https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12953><https://onlinelibrary.wiley.com/doi/10.1111/tgis.12953>, doi:10.1111/TGIS.12953.
- Runfola, D., Stefanidis, A., Baier, H., 2021. Using satellite data and deep learning to estimate educational outcomes in data-sparse environments. <https://doi.org/10.1080/2150704X.2021.1987575> 13, 87–97. URL: <https://www.tandfonline.com/doi/abs/10.1080/2150704X.2021.1987575>, doi:10.1080/2150704X.2021.1987575.
- Sekovski, I., Stecchi, F., Mancini, F., Rio, L.D., 2014. Image classification methods applied to shoreline extraction on very high-resolution multispectral imagery. *International Journal of Remote Sensing* 35, 3556–3578. URL: <https://doi.org/10.1080/01431161.2014.907939>, doi:10.1080/01431161.2014.907939, arXiv:<https://doi.org/10.1080/01431161.2014.907939>.
- Spinosa, A., Ziemba, A., Saponieri, A., Damiani, L., El Serafy, G., 2021. Remote sensing-based automatic detection of shoreline position: A case study in apulia region. *Journal of Marine Science and Engineering*. 9(6). URL: <https://www.mdpi.com/2077-1312/9/6/575>, doi:10.3390/jmse9060575.
- State of Washington, 2009. Shoreline master programs handbook. URL: <https://apps.ecology.wa.gov/publications/SummaryPages/1106010.html>.
- Stofa, M.M., Zulkifley, M.A., Zaki, S.Z.M., 2020. A deep learning approach to ship detection using satellite imagery. *IOP Conference Series: Earth*

- and Environmental Science 540, 012049. URL: <https://doi.org/10.1088/1755-1315/540/1/012049>, doi:10.1088/1755-1315/540/1/012049.
- Tarmizi, N.M., Samad, A.M., Yusop, M.S.M., 2014. Shoreline data extraction from quickbird satellite image using semi-automatic technique, in: 2014 IEEE 10th International Colloquium on Signal Processing and its Applications, pp. 157–162. doi:10.1109/CSPA.2014.6805739.
- Titus, J., Anderson, K., Cahoon, D., Gesch, D., Gill, S., Gutierrez, B., Thieler, E., William, S., 2009. Coastal sensitivity to sea-level rise: A focus on the mid-atlantic region. Accessed on August 2022: <https://digital.library.unt.edu/ark:/67531/metadc12028/>.
- Toft, J.D., Bilkovic, D.M., Mitchell, M.M., Peyre, M.K.L., 2017. A synthesis of living shoreline perspectives, in: Living Shorelines: The Science and Management of Nature-Based Coastal Protection. CRC Press. Taylor & Francis Group, pp. 483–488.
- Tran, S.T., Nguyen, M.H., Dang, H.P., Nguyen, T.T., 2022. Automatic polyp segmentation using modified recurrent residual unet network. IEEE Access 10, 65951–65961. doi:10.1109/ACCESS.2022.3184773.
- USGS, 2012. Massgis data - color ortho imagery (2008/2009). Retrieved on May 2022: <https://www.mass.gov/info-details/massgis-data-20082009-aerial-imagery>.
- VGIN, 2018. Virginia basemap program orthoimagery: Virginia geographic information network. Accessed on April 2022: <https://gismaps.vdem.virginia.gov/download/mrsid/>.
- Wan, J., Xie, Z., Xu, Y., Chen, S., Qiu, Q., 2021. Da-roadnet: A dual-attention network for road extraction from high resolution satellite imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14, 6302–6315. doi:10.1109/JSTARS.2021.3083055.
- Wang, Z., Wang, J., Yang, K., Wang, L., Su, F., Chen, X., 2022. Semantic segmentation of high-resolution remote sensing images based on a class feature attention mechanism fused with deeplabv3+. Computers & Geosciences 158, 104969. URL: <https://www.sciencedirect.com/science/article/pii/S0098300421002545>, doi:<https://doi.org/10.1016/j.cageo.2021.104969>.
- Yang, H.L., Lunga, D., Yuan, J., 2017. Toward country scale building detection with convolutional neural network using aerial images. International Geoscience and Remote Sensing Symposium (IGARSS) 2017-July, 870–873. doi:10.1109/IGARSS.2017.8127091.
- Zarillo, G.A., Kelley, J., Larson, V., 2008. A gis based tool for extracting shoreline positions from aerial imagery (beachtools) revised. Coastal and Hydraulics Laboratory Engineering Technical Note ERDC/CHL CHETN-IV-73. Vicksburg, MS: U.S. Army Engineer Research and Development Center. URL: <http://chl.erdrc.usace.army.mil/chetn>.