PAPER • OPEN ACCESS

Shoreline Recognition Using Machine Learning Techniques

To cite this article: Yunshun Zhong and Tamer El-Diraby 2022 *IOP Conf. Ser.: Earth Environ. Sci.* **1101** 022025

Shoreline Recognition Using Machine Learning Techniques

Yunshun Zhong, Tamer El-Diraby

27 King's College Circle, Toronto, Ontario, M5S 1A1, Canada

E-mail: yunshun.zhong@mail.utoronto.ca, tamer@ecf.utoronto.ca

Abstract. Coastal areas have emerged to be the most significant and dynamic regions worldwide. Therefore, automating shoreline recognition will aid non-profit conservation authorities to reduce public budget expenditures, relieve erosion damage, and increase the climate resilience of the natural environment. In this paper, advanced ML boosting algorithms including XGBoost, and LGBM are firstly applied into shoreline recognition with aerial images (of Lake Ontario in this study). This paper first discussed the significance and a literature review of recent progress in shoreline detection. Then, this paper adopted semantic segmentation instead of detecting shoreline directly, which enables the (Machine Learning) ML model to achieve relatively high accuracy with a small amount of data. 5 high-resolution images are used for training the model in which shorelines are detected. The work was carried out in four steps: 1) labeling the contents of shoreline images as areas of water and banks; 2) training ML algorithms; 3) using the trained algorithms to classify the image content as either water or land objects; 4) post-processing by de-noising image pixels (applying a Fourier transform algorithm) to obtain a defined shoreline. The averaged training time per image for Random Forest, XGBoost, and LGBM algorithms are 195.2 sec, 71.0 sec, and 8.6 sec, respectively. The averaged accuracy is 95.6%, 96.0%, and 94.8%, respectively; the XGBoost algorithm has slightly higher accuracy, while LGBM has a significantly shorter runtime. Cross-validation of the LGBM algorithm reduced the training time by around 23% (7.0 sec) and increased the accuracy by only 1.1% (to 95.9%).

1. Background

Coastal areas have emerged to be the most important and dynamic regions, worldwide. They are unique areas on earth because they are the connection between water body and land. They encompass a rich ecosystem with a high diversity of species and are among the most vulnerable to climate change, natural hazards and human intervention[1]. Shoreline is a line that forms the boundary between the land and the water body. It is estimated that there are about 504,000 km of shoreline worldwide, and more than 50% of the world's population lives within 100 km of the sea[2]. Shoreline erosion is a natural process whereby soil is dislodged and worn away due to an erosive agent—typically storm water, waves, and wind. Land use policy and climate change contribute significantly to accelerating natural erosion. Flooding, hurricane, sea-level rise, warmer water temperature, changes in the intensity of waves, as well as coastal construction may cause erosion of shoreline. One of the most obvious negative impact of erosion in this context is the damage to adjacent properties. For example, it has been estimated that the erosion of shorelines in California could impact the value for properties worth up to $100 Billion[3]. Shoreline erosion can also have significant impacts on key enviro-economic factors—for example, shellfish and other marine habitats, water quality transportation channels, and recreation facilities.

Detecting and monitoring shorelines are consequently of significant economic and social importance in managing shorelines assets due to risks of shoreline erosion, especially if we know that climate change has devastating effects on coastal areas. Therefore, shoreline recognition plays an essential role for conservation authorities to manage environmental and natural assets, and structures near shorelines. Automating this procedure will help conservation authorities such as Toronto and Region Conservation Authority (TRCA) to reduce public budget and erosion damage, improve environmental sustainability, and increase climate resilience of natural environment and ecological services. TRCA is one of 36 non-profit conservation authorities in Ontario, Canada, created to safeguard and enhance the health and well-being of watershed communities through the protection and restoration of the natural environment and the ecological services the environment provides. The objective of this research was to apply advanced image recognition systems to help TRCA track shorelines and the erosion of shorelines automatically. This research used existing aerial images to develop machine learning and deep learning algorithms to automatically detect shorelines of Lake Ontario.

2. Literature Review

There are three main ways for shoreline recognition: Firstly, traditional ground-based techniques such as topographic surveying and GPS measurements which are surveyed repeatedly[4]: this approach is accurate, but labor-intensive and time-consuming. Secondly, remote sensing-based approaches to mapping shoreline change using satellite images[5]: the spatial coverage is usually large but the spatial resolution of satellite images are often low. For shorelines that have structures nearby such as shorelines in Lake Ontario, the spatial resolution is usually not enough to determine whether the structures will be influenced by the erosion of shorelines or not. Thirdly, aerial photographs-based approaches[6]: the advent of relatively inexpensive and easily used balloon or Unmanned aerial vehicles (UAV) photography systems facilitates the acquisition of high-frequency and high-resolution aerial imagery[7]. In this study, high-resolution aerial images obtained from TRCA will be used for shoreline detection.

There have been many attempts to effectively detect the shoreline from aerial and satellite images. Ryan et al. (1991) proposed an image segmentation approach that was tested on scanned U.S. geological survey (USGS) aerial photographs[8]. Li et al. (2003) used satellite imagery for mapping shorelines by using the photogrammetry techniques[9]. Lee et al. (2010) proposed a method for shoreline extraction from integrated LiDAR point cloud data and aerial orthophotos using mean shift segmentation[10]. The traditional approaches mentioned above used well-established computer vision (CV) techniques and feature extraction is an essential step for this task. Before the emergence of Machine Learning (ML), pre-processing, shoreline detection, and post-processing are all designed in a hand-crafted way. Usually, each data set of images or even each image in the data set (if images in same data set have large variance) need different procedures. It is up to the CV engineer's judgment and a long trial and error process to decide what kind of procedure can lead to the best results. Moreover, designing each procedure requires dealing with a plethora of parameters, all of which must be fine-tuned by the CV engineer[11]. ML-based approaches have the ability to automate this procedure by optimizing parameters with gradient descent or closed form solutions and optimizing hyper-parameters, parameters whose values are used to control the learning process, with cross-validation. There are two main different categories of machine learning methods: supervised learning and unsupervised learning. Supervised learning is defined by its use of labeled datasets, whereas unsupervised learning analyze and cluster unlabeled data sets. Lee et al. (2012) utilized high-resolution satellite imagery for mapping shorelines by using the unsupervised segmentation method[12]. Hannv et al (2013) utilized Landsat imagery for mapping the shoreline by using the support vector machines (SVM) (a supervised classification method)[13]. Compared to detect shoreline directly as above mentioned approaches, performing semantic segmentation to classify each pixel of the image into water and land then applying edge detection on the boundary of water and land to extract shoreline have the following advantages: 1) Each image usually has large amount of pixels. Hence there will be significant amount of training data to train the model, which means this approach is able to perform well on very small data set. 2) The loss function for detecting shoreline directly can be tricky since there are many definitions of "distance" between the actual and predicted shorelines and some definitions are vague. However, Cross Entropy Loss can be used for semantic segmentation as it's a classic classification task. 3) Semantic segmentation is less computationally expensive than detecting shoreline directly. Therefore, semantic segmentation is adopted in this study. Masria et al. (2015) utilized Landsat Satellite images to extract shoreline positions and estimate shoreline change rates of the Nile delta coast by using SVM with semantic segmentation[14]. Bengoufa et al. (2021) utilized remotely sensed images to extract Mostaganem coastline (Algeria) by using SVM and random forest (RF) with semantic segmentation[15]. However, SVM is a linear classifier which may prone to constantly misclassify certain data. Even using SVM with kernel methods, SVM is still linear in high dimensional space and designing a kernel (if necessary) require engineer's judgment and a trial and error process. Non-linear ML algorithms using boosting technique such as XGBoost[16] and LGBM[17], a highly efficient gradient goosting decision tree, usually outperform random forest because it reduces bias linearly whereas variance increases in log scale. RF can reduce variance but bias may increase at the same magnitude, which limits it's accuracy. XGBoost and LGBM has not been used on shoreline recognition from aerial images. In this study, XGBoost and LGBM will be implemented to detect shoreline automatically. RF will also be used as a comparison.

The difficulty with ML approaches is that it is necessary to choose which features are important in each given image. As the number of images to classify increases, feature extraction becomes more and more cumbersome. Deep Learning (DL) introduced the concept of end-to-end learning where the machine is just given a dataset of images which have been annotated with what classes of object are present in each image[18]. Thereby a DL model is 'trained' on the given data, where neural networks (NN) discover the underlying patterns in classes of images and automatically works out the most descriptive and salient features to classify pixels as water or land. However, training a NN usually require large datasets which is not the case in this study. In addition, training a NN can take a very long time[11]. Training may take hours or days depending on the hardware availability. Moreover, DL models have numerous hyper-parameters in which tuning them needs trial and error for many iterations. Therefore, DL models are not used in this study.

3. Research Methodology

In summary, the objective of this study was to accurately and automatically detect shoreline from aerial images by using ML and DL algorithms. Of special interest is to examine the use of image processing to detect patterns in shoreline erosion, and, possibly in the future, predict scenarios for erosion. Firstly, this study used semantic segmentation to classify the image into water and land then performed edge detection to generate shoreline instead of detecting shoreline directly. Due to the large amount of pixels in each image, the semantic segmentation approach in this research is able to obtain relative high accuracy with only a tiny amount of training images (5 high-resolution images in this study). Finally, XGBoost and LGBM, and RF are firstly used in classification of shoreline using aerial images. This study used aerial images covering portions of Lake Ontario shoreline in PNG and JPEG format. Of course, the algorithm proposed by this study can be applied to other types of images such as orthoimages.

The paper is structured as follows: Background discusses the significance of automatic shoreline detection in helping conservation authorities to reduce public budget and erosion damage, improve environmental sustainability, and increase climate resilience of natural environment and ecological services. Literature review provides a review of recent progress in shoreline detection using image analysis, including advances in the use of ML. Research methodology lists the steps of the analysis. A section on the results of the analysis is then presented. Finally, a discussion section is presented along with recommendations for future work. For this project, the general workflow can be divided into three parts: labeling data, training machine learning (ML) model, post-processing and applying Fourier transform edge detection.

3.1. Label Images

Apeer Annotate, a web-based annotation tool developed for ML tasks was used in labeling images. Each circle (label) on the left graph of Figure 1 contains the same number of pixels. The number of labels for water and bank are almost equal and uniformly distributed.

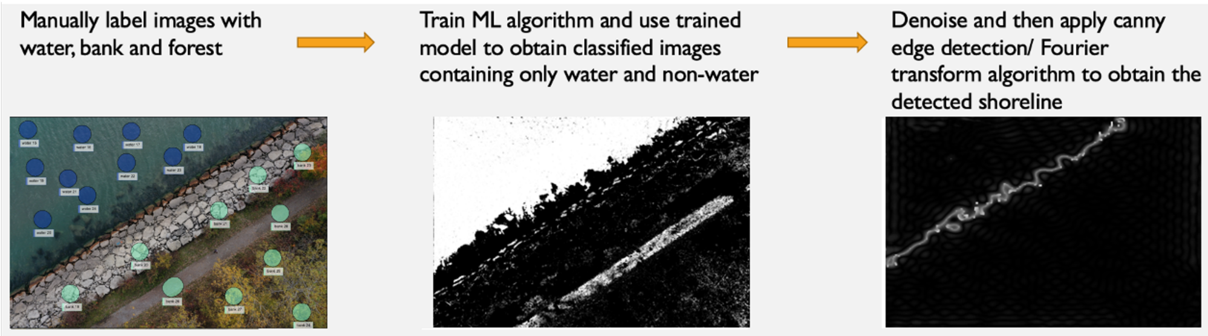

Figure 1. General Workflow

3.2. Machine Learning Analysis

For training the ML algorithm, the following steps were implemented as shown in Figure 2 below:

(i) image graying;
(ii) feature selection with filters such as Canny edge, Roberts edge, and Sobel;
(iii) adding labels as a new layer into training images;
(iv) applying machine learning algorithms (we considered Random Forest, XGBoost, and LGBM).

3.2.1. Pre-process Data and Feature Selection for ML Algorithms

1. Image greying:

Processing color images usually requires more time (especially for high-resolution images). In addition, many edge detection and filtering algorithms can only be applied to single-channel images, whereas color images have three channels (red, green, and blue). Therefore, color images were converted into grayscale images. The formula for image graying is shown in Equation 1 [19]:

$$f = 0.299R + 0.587G + 0.114B \tag{1}$$

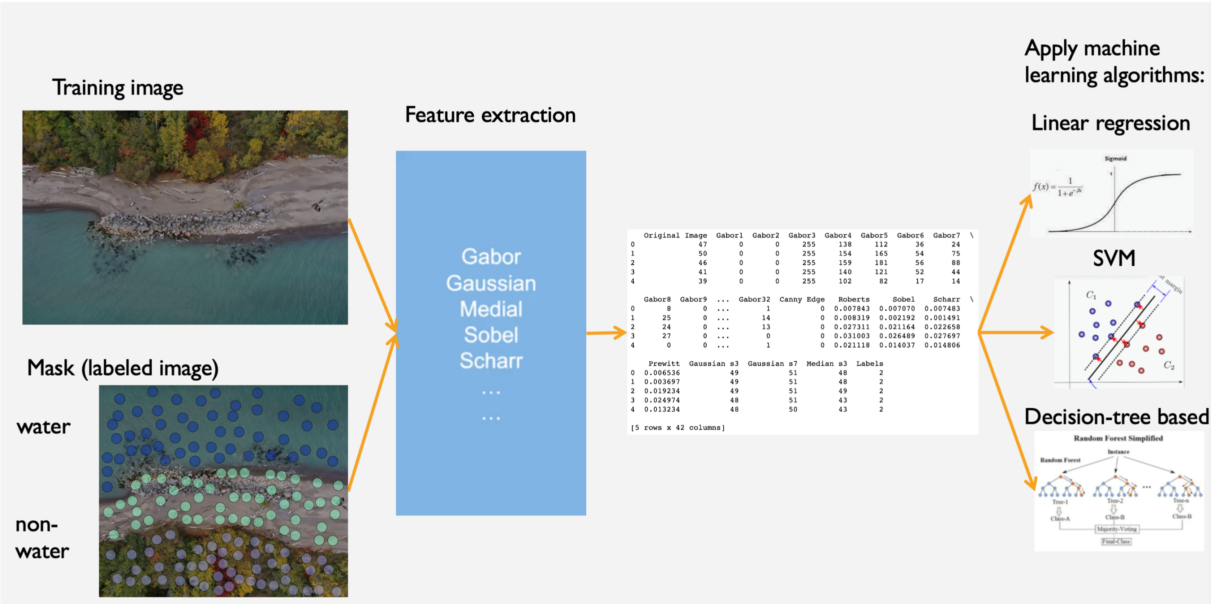

Figure 2. Workflow for Machine Learning Algorithm

Where R, G, and B are the values of the red, green, and blue color channels of the pixel, respectively.

2. Feature selection:

First, the pixel value (0/1) was read as a feature. Then, using Gabor filters, a group of Gabor features with parameters $\theta \in \left\{0, \frac{\pi}{4}\right\}, \sigma \in \{1, 3\}, \lambda \in \left\{0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}\right\}, \gamma \in \{0.05, 0.5\}$ were generated as features in this study. Gabor filter is a linear filter used for texture analysis. It examines whether there is any specific frequency content in the image in specific directions within a localized region around the point or region of analysis. The equation for the Gabor filter is shown in Equation 2 below:

$$g(x, y; \theta, \lambda, \psi, \sigma, \gamma) = \exp\left(-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x\prime}{\lambda} + \psi\right) \tag{2}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$.

In this equation, λ represents the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma/standard deviation of the Gaussian envelope and *gamma* is the spatial aspect ratio (it specifies the ellipticity of the support of the Gabor function). After applying Canny edge, Roberts edge, Sobel, Scharr, and Prewitt filters to image separately, a group of other features were also generated.

3. Read labeled images (masks) and add them as a new layer into training images

Labeled images (masks) were read into the commonly used pandas dataframe, a two-dimensional, size-mutable, tabular data which is widely used in data analysis, in the same way as training images. Then dataframe for masks and training images were concatenated together into one dataframe. The unlabeled data in the dataframe were removed to reduce computation time.

4. Prepare ready-to-use training data

The values of bank labels were set to 1 and the values of water labels pixels were set to 0. Then, the two labels were designated as the target (Y) vector, while all the features selected

IOP Conf. Series: Earth and Environmental Science 1101 (2022) 022025 doi:10.1088/1755-1315/1101/2/022025

above were designated as variable matrix X. This allows the algorithms to classify pixels as either water or non-water. Y was encoded with Label Encoder from Python library scikit-learn, which can facilitate using other tools such as ROC plots. Data were split into training and test data with 80% and 20% of the whole labeled dataset, respectively.

5. Define the classifier and model training

Machine learning classifiers such as RF, XGBoost, and LGBM were examined. Python library scikit-learn was used to build up random forest, XGBoost, and LGBM models. In all cases, a tree is composed of a set of levels, at each, there are a set of options (leaves) for the algorithm to choose from.

4. Findings and Discussion

4.1. Initial Results

Random forest, XGBoost and LGBM has similar accuracy, which are summarized in Table 1 below:

	Training time (s)	Accuracy
Random Forest	195.2	95.6%
XGBoost	71.0	96.0%
LGBM	8.6	94.8%

Table 1. Training Time and Accuracy for Three Algorithms

The training time is the averaged training time for 5 images on a 2.6 GHz 6-Core Intel Core i7 processor. Accuracies are averaged over 5 images. XGBoost has the highest accuracy. However, the accuracies for all three algorithms are very close. LGBM has significantly less runtime compared to the other two algorithms. Since training time is not a major concern, XGBoost is selected as the final approach. The receiver operating characteristic curve (ROC) curve for XGBoost model is shown below in Figure 3. Since ROC curves are similar for all images, only the ROC curve for the first image is shown.

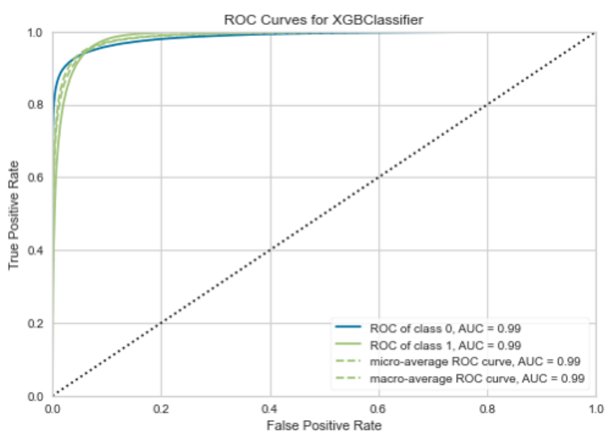

Figure 3. ROC Curve for XGBoost

The final model is applied to the shoreline images in Figure 4. The segmentation results from the XGBoost model is shown in Figure 5 below. The detected shorelines by Fourier transform edge detection are shown in Figure 6. This result will be discussed in detail in Section 4.3.

Figure 4. Original Shoreline Images

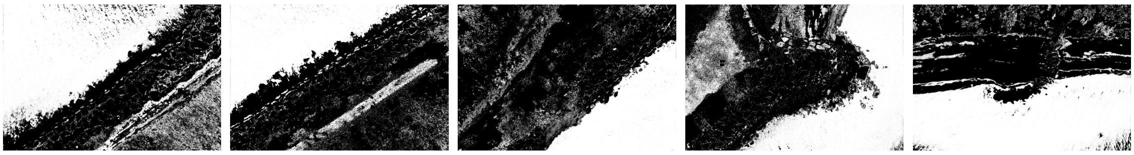

Figure 5. Shoreline Images After Segmentation

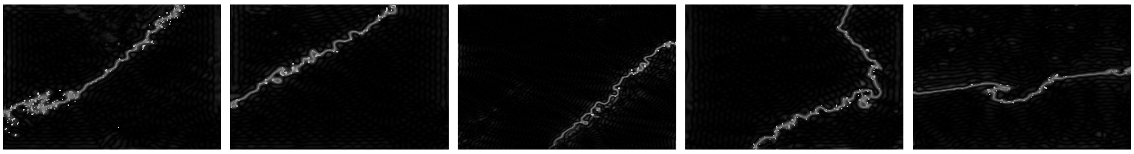

Figure 6. Shoreline Images After Fourier Transform Edge Detection

4.2. Cross Validation for Hyperparameters Selection

Much like all machine learning algorithms, the accuracy of decision tree based algorithms are sensitive to the parameters selected in building them at the start (seeding). An iterative approach called cross-validation was used to examine different sets of parameters. LGBM algorithm is used in this section as an example. Hyperparameters before and after cross-validation are summarized in Table 2. Training time and accuracy before and after cross-validation are summarized in Table 3. Cross validation reduces the training time by around 23% (7.0 seconds) and increases accuracy by 1.1% (to 95.9%). It's not surprising that cross-validation have only minor improvement on the accuracy of the model. The used parameters were empirically selected at the start (they were not default or random analysis parameters).

	learning rate	boosting type	objective	Num$_{l}eaves$	Max$_{d}epth$	Num$_{c}lass$
Before cross validation	0.05	dart	multiclass	100	10	2
After cross validation	0.1	dart	multiclass	150	15	2

Table 2. Hyperparameters in LGBM algorithm before and after cross-validation

	Training time (s)	Accuracy
Before cross validation	8.6	94.8
After cross validation	7.0	95.9

Table 3. Training time and accuracy before and after cross-validation for LGBM

4.3. Edge detection after machine learning segmentation

After the machine learning segmentation, edge detection algorithms should be used to classify images and extract the shoreline. However, pre-processing is needed to reduce the noise before

Fourier edge detection algorithm is applied.

4.3.1. Pre-processing of segmented images The image pre-processing includes two steps:

1. Denoising by using non-local means (NLM) filter Non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. Suppose ω is the area of an image, and p and q are two points within the image. Then, the algorithm is [20]:

$$u(p) = \frac{1}{C(p)} \int_{\Omega} v(q) f(p,q) dq \tag{3}$$

where $u(p)$ is the filtered value of the image at point p, $v(q)$ is the unfiltered value of the image at point q, $f(p,q)$ is the weighting function, and the integral is evaluated at $\forall q \in \Omega$. $C(p)$ is a normalizing factor, given by:

$$C(p) = \int_{\Omega} f(p,q) dq \tag{4}$$

2. Dilation, erosion to further reduce noise Dilation means the expansion of the highlighted part of the image. It combines all background points in contact with the foreground area into the object and can be used to fill the holes in the foreground area. Corrosion means the corrosion of the highlighted part of the image. It is a process of shrinking the boundary inward and can be used to eliminate small and meaningless foreground areas [19].

4.3.2. Apply Fourier Transform Edge Detection Algorithm Fourier transform edge detection converts the image into frequency space. Edges are usually in the corner of the frequency space because they have high-frequency components. By masking out all contributions from the center, edges can be detected. Figure 6 shows the result after Fourier transform edge detection. Compared with Canny edge detection, Fourier transform edge detection has a much better ability to reduce noise and improve the detection result. Therefore, Fourier transform edge detection is selected as the edge detection method in this study. After edge detection, closure operation was performed to eliminate dis-connected noises from shoreline.

5. Conclusions and Further Research

The shoreline detection approach proposed in this report can largely identify the outline of the shoreline with some offset but cannot eliminate all the noises. This approach includes Image graying, feature selection with Gabor, Canny edge, Roberts edge, Sobel, Scharr, Prewitt filters, XGBoost algorithm, NLM filter, dilation, erosion, and Fourier transform edge detection. Compared to traditional shoreline detection methods such as using Fourier transform edge detection directly on original images, the proposed approach can detect shoreline from new images automatically without manually tuning parameters for each new image. The averaged training time per image for Random Forest, XGBoost, and LGBM algorithms are 195.2 sec, 71.0 sec, and 8.6 sec, respectively. The averaged accuracy is 95.6%, 96.0%, and 94.8%, respectively. XGBoost has the highest accuracy but accuracies for all three algorithms are similar. LGBM has significantly less runtime compared to the other two algorithms. Accuracy can potentially be further improved if the algorithm is trained on a larger dataset.

Comparing the accuracy from this study to similar studies may not be very meaningful. One reason is that a comprehensive and public-accessible shoreline dataset to assess ML model performance does not present. Even for the same model, it can perform well on satellite images while getting low accuracy on aerial images due to difference of two datasets. Even for two aerial image datasets, the model performance is still going to have large variation. Another reason is that the matrix for measuring the performance of models has large variation. Accuracy, confusion matrix, and buffer method are all being used in shoreline recognition. Therefore, this study did not compare results to other studies.

It is important to note that traditional machine learning methods such as XGBoost have an inherent disadvantage compared to deep learning methods: they lack the ability to change the weights and parameters of each filter during training. The "backpropagation" step in the artificial neural network does exactly that. Building a convolutional artificial neural network from scratch usually requires a large training dataset. A neural network model may be built in the future when we have more aerial image data.

6. Reference

[1] Vasilis Paravolidakis, Lemonia Ragia, Konstantia Moirogiorgou, and Michalis E Zervakis. Automatic coastline extraction using edge detection and optimization procedures. *Geosciences*, 8(11):407, 2018.
[2] Seynabou Toure, Oumar Diop, Kidiyo Kpalma, and Amadou Seidou Maiga. Shoreline detection using optical remote sensing: A review. *ISPRS International Journal of Geo-Information*, 8(2):75, 2019.
[3] Matthew Heberger, Heather Cooley, Pablo Herrera, Peter H Gleick, and Eli Moore. The impacts of sea-level rise on the california coast. 2009.
[4] Peter Ruggiero, George M Kaminsky, Guy Gelfenbaum, and Brian Voigt. Seasonal to interannual morphodynamics along a high-energy dissipative littoral cell. *Journal of Coastal Research*, 21(3):553–578, 2005.
[5] Charles Fletcher, John Rooney, Mathew Barbee, Siang-Chyn Lim, and Bruce Richmond. Mapping shoreline change using digital orthophotogrammetry on maui, hawaii. *Journal of Coastal Research*, pages 106–124, 2003.
[6] Michiru Miyamoto, Kunihiko Yoshino, Toshihide Nagano, Tomoyasu Ishida, and Yohei Sato. Use of balloon aerial photography for classification of kushiro wetland vegetation, northeastern japan. *Wetlands*, 24(3):701–710, 2004.
[7] Mike J Smith, Jim Chandler, and James Rose. High spatial resolution data acquisition for the geosciences: kite aerial photography. *Earth Surface Processes and Landforms*, 34(1):155–161, 2009.
[8] TW Ryan, PJ Sementilli, P Yuen, and BR Hunt. Extraction of shoreline features by neural nets and image processing. *Photogrammetric Engineering and Remote Sensing*, 57(7):947–955, 1991.
[9] Ron Li, Kaichang Di, and Ruijin Ma. 3-d shoreline extraction from ikonos satellite imagery. *Marine Geodesy*, 26(1-2):107–115, 2003.
[10] I-Chieh Lee, Liang Cheng, and Ron Li. Optimal parameter determination for mean-shift segmentation-based shoreline extraction using lidar data, aerial orthophotos, and satellite imagery. In *ASPRS Conference*, 2010.
[11] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and information conference*, pages 128–144. Springer, 2019.
[12] I-Chieh Lee. *Instantaneous shoreline extraction utilizing integrated spectrum and shadow analysis from LiDAR data and high-resolution satellite imagery*. The Ohio State University, 2012.
[13] Hannv Zhang, Qigang Jiang, and Jiang Xu. Coastline extraction using support vector machine from remote sensing image. *J. Multim.*, 8(2):175–182, 2013.
[14] Ali Masria, Kazuo Nadaoka, Abdelazim Negm, and Moheb Iskander. Detection of shoreline and land cover changes around rosetta promontory, egypt, based on remote sensing analysis. *Land*, 4(1):216–230, 2015.
[15] Soumia Bengoufa, Simona Niculescu, Mustapha Kamel Mihoubi, Rabah Belkessa, Ali Rami, Walid Rabehi, and Katia Abbad. Machine learning and shoreline monitoring using optical satellite images: case study of the mostaganem shoreline, algeria. *Journal of applied remote sensing*, 15(2):026509, 2021.
[16] T Chen and C Guestrin. Xgboost: a scalable tree boosting system. arxiv [cslg] 2016, 2018.
[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
[18] Philipp Koehn. Combining genetic algorithms and neural networks: The encoding problem. 1994.
[19] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. *Image Processing On Line*, 1:208–212, 2011.
[20] A Brad Murray, Sathya Gopalakrishnan, Dylan E McNamara, and Martin D Smith. Progress in coupling models of human and coastal landscape change. *Computers & geosciences*, 53:30–38, 2013.