2019-12

Change Detection of Marine Environments Using Machine Learning

Orescanin, Mara S.

Monterey, California: Naval Postgraduate School

https://hdl.handle.net/10945/69925

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

CHANGE DETECTION OF MARINE ENVIRONMENTS USING MACHINE LEARNING

by

Prof. Mara Orescanin / Mr. Jeremy Metcalf

October 2019

Approved for public release; distribution is unlimited

Prepared for: HQMC Intelligence Department/US Coast Guard Research and Development Center. This research is supported by funding from the Naval Postgraduate School, Naval Research Program (PE 0605853N/2098). NRP Project ID: NPS-19-M020-A

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)***	**2. REPORT TYPE**	**3. DATES COVERED** ***(From-To)***
31-10-2019	Final Report	01-10-2018 – 30-09-2019

4. TITLE AND SUBTITLE
CHANGE DETECTION OF MARINE ENVIRONMENTS USING MACHINE LEARNING

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER 0605853N/2098

6. AUTHOR(S)
Prof. Mara Orescanin
Mr. Jeremy Metcalf

5d. PROJECT NUMBER
NPS-19-M020-A; W9A59

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
Naval Postgraduate School

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-19-M020-A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Postgraduate School, Naval Research Program
Topic Sponsor Lead Organization: HQMC Intelligence Department/US Coast Guard Research and Development Center
Topic Sponsor Name: Geospatial Intelligence Branch Deputy, Mr. Daniel Hannigan/Director of Research Partnerships/Public Affairs Officer, Dr. Joe DiRenzo
Topic Sponsor Contact Information: Daniel.g.hannigan@usmc.mil 860 271-2659/ Shannon.gorrell@usmc.mil / Joseph.DiRenzo@uscg.mil 860-271-2738

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Machine learning (ML), specifically deep learning (DL) using convolutional neural networks, is an increasingly powerful tool for classification of complex systems, including visual images and multispectral data. The focus of this study is to implement ML algorithms for 1) littoral environment classification and 2) change detection of littoral environments in order to rapidly assess changes in water quality, such as debris and oil slicks. Specifically, small unmanned aerial systems (UAS) were used to acquire data, including visual red-green-blue (RGB), RedEdge, near infrared (NIR) and infrared (IR), locally to Monterey Bay in order to train a deep neural network to recognize littoral environments on land, such as beach, marsh, and rocks, as well as water bottom type, such as sand, rock, and vegetation/algae. This database of images collected from UAS and small aircraft was used to assess coastal areas near ephemeral rivers that are known to seasonally breach thereby changing the littoral environment and water quality through erosion/removal of vegetation as well as sediment suspension. The remote sensing images were validated by site observations as well as morphodynamic observations of beach change in order to quantify the aerially observed changes. Results of ML model training indicate highly accurate (>90%) detection of littoral environments, both over land and littoral waters, without any need for image segmentation. These findings suggest models for targeted areas could be developed for rapid and accurate change detection post extreme events (hurricanes/tsunamis).

15. SUBJECT TERMS
Machine learning (ML), deep learning (DL), littoral environments, change detection, unmanned aerial systems (UAS), remote sensing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			
Unlimited	Unlimited	Unlimited		43	Prof. Mara Orescanin
					19b. TELEPHONE NUMBER ***(include area code)*** 831-656-3233

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Machine learning (ML), specifically deep learning (DL) using convolutional neural networks, is an increasingly powerful tool for classification of complex systems, including visual images and multispectral data. The focus of this study is to implement ML algorithms for 1) littoral environment classification and 2) change detection of littoral environments in order to rapidly assess changes in water quality, such as debris and oil slicks. Specifically, small unmanned aerial systems (UAS) were used to acquire data, including visual red-green-blue (RGB), red edge, near infrared (NIR) and thermal infrared (TIR), locally to Monterey Bay in order to train a deep neural network to recognize littoral environments on land, such as beach, marsh, and rocks, as well as water bottom type, such as sand, rock, and vegetation/algae (kelp). This database of images collected from UAS and small aircraft was used to assess coastal areas near ephemeral rivers that are known to seasonally breach thereby changing the littoral environment and water quality through erosion/removal of vegetation as well as sediment suspension. The remote sensing images were validated by site observations as well as morpho dynamic observations of beach change in order to quantify the aerially observed changes. Results of ML model training indicate highly accurate (>90%) detection of littoral environments, both over land and littoral waters, without any need for image segmentation. These findings suggest models for targeted areas could be developed for rapid and accurate change detection post extreme events (hurricanes/tsunamis).

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION AND RESEARCH OBJECTIVES

Landscape classification in coastal environments is a sub-discipline of large-scale landscape or land-use classification techniques involved in many areas of geospatial information analysis. Recently, ML algorithms have been implemented with heterogeneous landscapes to provide pixel-level classification of imagery (e.g. Buscombe and Ritchie, 2018; Maggiori, et al., 2017; Salamati, et al., 2012). However, the generation of ML models that are capable of pixel classification require carefully curated images with class delineations. While this method offers decent results for test images, it is not easily transferrable to datasets that have not been hand-labeled at the pixel level or have a wide range in field of view (or other image characteristics). In contrast to pixel-level classification, this study proposes traditional image classification for heterogeneous coastal environments in order to provide a model that is widely applicable to global coastal environments.

Aerial imagery, collected either by UAS or small aircraft, provides high-resolution (spatial and temporal) monitoring of coastal marine areas. This data can be used to identify and classify specific marine environments and visual changes in water quality, with the goal of creating a neural net that can then be used for assessment of marine environment changes. Specifically, local California coastal areas were used as proof-of-concept to identify regions including kelp forests, sandy bottom habitats, and reef/rocky bottom habitats. This approach could be applied to remote marine areas including Pacific National Monuments and missile test ranges with the intent of providing efficient management practices for these difficult-to-patrol areas. This project was a collaboration with the United States Coast Guard Research and Development Center.

The research questions are:

1. Can littoral environment and bottom type be determined by a neural network image feature extraction method?

Hypothesis 1: Provided sufficient data, deep neural networks are capable of identifying different littoral environments to high accuracy (>90%).

2. Does multispectral imagery significantly enhance deep learning classification of littoral waters?

Hypothesis 2: 5-band imagery enhances classification, but deep learning is also critical to fully classify littoral landscapes.

3. Can an algorithm be created that automates an assessment of change or damage at high-risk coastal assets?

Hypothesis 3: Using the trained neural networks from Hypothesis 1, it is possible to combine predictions from two identical regions but at different times into a single map indicating regions of greatest change.

B. BACKGROUND

With advances in machine learning algorithms, specifically deep learning, for image classification and analysis, the opportunity to augment physical processes-based research with computer-automated analysis can help identify global-scale patterns efficiently. For coastal landscape classification, with rapid change owing to physical forcing (waves, storm surge, precipitation/discharge from rivers) and a broad range in features for the same class, having the ability to target areas of similar change regionally will augment the statistical understanding of physical coastal processes. This automation would provide a method for rapid change detection post extreme event for disaster relief. However, there is not consistent understanding on how accurate deep learning methods are for highly variable datasets.

Given that CNN models are the best-performing models in large-scale image recognition tasks (Krizhevsky et al. 2012, Simonyan and Zisserman 2014, Han et al. 2016) it is desirable to apply them to image recognition tasks where the training datasets are of much smaller size such as the developed coastal dataset. Large annotated datasets in heterogeneous coastal landscapes do not exist, despite the multitude of images. An additional challenge is single class identification owing to the presence of overlapping class traits in the same image. Furthermore, several coastal landscapes, such as beaches, have few identifiable features that define the class as compared with other landscapes such as rocky coasts. Owing to limited annotated datasets, it is difficult to train a neural network from scratch, due to the large number of tunable parameters that modern CNN architectures like NASNet (Zoph et al. 2018) or ResNet (He et al. 2016) have.

Coastal accretion and erosion, land use/cover with GIS and neural networks, and mapping of natural hazards and disasters have been extensively studied using satellite imagery and various analysis techniques (e.g. Mas, 2004; Maiti and Bhattacharya, 2009; Joyce et al. 2009, Natesan et al. 2015). Much of the landscape-based classification techniques are focused on segmentation of areas using machine learning that requires a pixel-level annotated dataset defining class features (Berlanga, et al., 2002; Tamassoki, et al. 2014; Hoonhut et al., 2015; Morgan, et al. 2015; Ballari, et al. 2016; Maggiori, et al., 2017). This is largely owing to the heterogeneity of landscape imagery as well as the variety within each class compared to discrete object-based classification (i.e. ImageNet, Krizhevsky et al. 2012). The technique of image segmentation addresses the issue of heterogeneity by separating the image into homogenous regions (e.g. Kuleli et al., 2011), and merging pixels into objects (also called superpixels) that are then classified based on different objects via the more error-prone way of classification using individual pixels (Lu and Weng, 2007). An example is accurate hybrid segmentation method (Buscombe and Ritchie, 2018), which combines the ability of deep convolutional neural networks to classify small regions in an image plus the use of fully connected conditional random fields for fine-grained localization pixel-level classification. This method has achieved high accuracy classification results of 88–98% (F1 scores) for five datasets using a different number of tiles applicable to large, spatially extensive landscapes (Buscombe and Ritchie, 2018) and can also be applied to large morphodynamic heterogeneous coastal areas. However, the method still relies on the segmentation of images into homogeneous landscape objects. The above-mentioned methods often require hand-annotated images depicting each class within a single image with bounded areas annotated by hand, which is labor-intensive and subjective.

Unmanned Aerial Systems (UASs) are highly-versatile low-cost platforms that can be used to map and monitor areas of environmental interest including cropland and forestry (Dunford et al. 2009, Rango et al. 2009, Turner et al. 2012), soil erosion (d'Oleir-Oltmanns et al. 2012), and coastal landscapes (Buscombe and Ritchie, 2018).

Combined with larger-scale aerial imagery efforts from manned aircraft, such as those performed by the USGS (Morgan et al., 2015), the amount of aerial imagery is sufficient to create a coastal database of landscapes spanning the coastlines of the entire United States that can augment databases of satellite imagery with both higher-resolution and higher frequency datasets than satellite observations.

However, most landscape imagery, even at these smaller scales, is not homogenous to one type of class: images regularly capture multiple classes and there is high variability within each class. Therefore, it is imperative to have an adequate number of training samples that are representative of each class (Hubert-Moy et al. 2001, Chen and Stow 2002, Landgrebe 2003, Mather 2004, Lu and Weng 2007). In order to test, the applicability of deep learning methods for landscape classification, over 10000 images of the U.S. West, East, and Gulf Coasts were hand classified into eight classes (samples shown in Figure 1). The imagery within each class was selected by the most dominant class present in the image, despite the presence of other classes within the same image.

C. PROJECT DESCRIPTIONS

1. Landscape Classification Using Various Neural Network Architectures

Here, thirteen convolutional neural network (CNN) deep learning models are developed for coastal landscape classification. The hypothesis of this study is that heterogeneous coastal landscapes can be used to train a deep neural network through transfer learning with a high degree of accuracy (>90% correct identification) without the need for image segmentation/pixel classification.

The ontology, or determination of class structure, was created based on NOAA's Environmental Sensitivity Index (ESI) for coastal landscapes along with considerations from general NOAA Land Use categories (NOAA, 2013). The ESI classification accounts for coastal material (fine sand to rocks) and slope (tidal flats to cliffs). Based off these general classes as well as examples presented in the data, one goal of this project is to test the ability of deep learning to distinguish classes that are useful to the larger community. However, in order for an ontology to be capable of using with deep learning models, it is necessary for each class to possess distinctive characteristics that identify the class.

Convolutional Neural Networks learn image features in their lower layers that are not domain specific, and therefore generalize to many datasets and domains (Yosinski et al., 2014). It is still not well-known whether this transferability of models trained on ImageNet provide similar accuracies when transferred across various domains (Kornblith et al., 2019). Here, pretrained ImageNet models are used with a transfer learning approach (Yosinski et al., 2014; Kornblith et al., 2019) to develop classification models on a newly developed coastal dataset. This approach is similar to that of Marmanis et al., 2015, who used pretrained CNN models on ImageNet for feature extraction from satellite imagery.

ImageNet pretrained model weights are available from multiple online resources and here pretrained models were downloaded through Keras/Tensorflow (Chollet, 2015; Abadi et al., 2016). In this study thirteen separate models are evaluated for feature transferability to coastal landscapes. Specifically, the two configurations of pre-trained models used for transfer learning to the target model are 1. Fixed-feature extractors (Donahue et al., 2014; Kornblith et al. 2019) and 2. Fine-tuning from ImageNet initialization (Agrawal et al., 2014; Kornblith et al. 2019).

The architecture for both transfer learning approaches used here, is to take a pre-trained network on ImageNet and copy it without the top classification layers (so called "bottleneck" built of hidden fully connected layers and a softmax layer) to the target network to initialize the new network. Previous classification layers are removed because they are task specific. The target network is then finalized by adding on top of the transferred convolutional layers two fully connected layers (rebuilding new "bottleneck", randomly initialized). The rectified linear unit (ReLU) is used for the activation function and adds nonlinearity to the layers, which is required in order to solve anything more complex than a linear fit. To control the overfitting dropout is used where 50% of neurons in the fully connected layers are active in each computation (Srivastava et al., 2014). A softmax layer of the size equal to the number of classes is added on the top of the architecture. In case of some architectures such as ResNet50, where there are no hidden fully connected layers on top of convolutional layers but a single softmax layer of class size, we remove only that last layer and add the same top layers.

The target network is developed using Keras with Tensorflow backend. Stochastic gradient descent (SGD) is used as an optimizer with learning rate, lr = 0.0001 and momentum of 0.9. Cross-entropy loss is minimized. If the target dataset is small and the number of parameters is large, fine-tuning may result in overfitting, therefore, training is stopped if the validation loss stopped decreasing after 30 epochs, an early stopping strategy (Goodfellow et al., 2016).

In the first set of transfer learning experiments, similarly to Kornblith et al., 2019, the transferred convolutional layers are treated as fixed feature extractors, which means that during training, gradients do not update weights in these layers (they are “frozen” and only the newly added layers that are randomly initialized are allowed to update weights). In the second set of transfer learning experiments, trained models from the first experiment are additionally retrained by allowing convolutional weights to be updated as well as the classification layers on the top of the target architecture, a process called fine tuning. The learning rate is kept constant across the layers in order to have a consistent approach amongst multiple architectures tested.

From each coastal category class 850 images were used for target model training. A further 200 images from each distinct class were used for the validation set during training to determine the accuracy of the training phase and to monitor the overfitting. There were 100 randomly selected images from each coastal class bin that were set aside and were not part of the training/validation phase. These unseen images were the test image set and yielded unbiased machine results depicting the binned coastal classes.

Two primary ways of combating the overfitting are by using dropout in classification layers and by data augmentation (Krizhevsky et al. 2012). Here, data augmentation was used during training phase on the training data set only. Data augmentation techniques were randomly applied to images as a part of the training pipeline and consisted of following techniques: horizontal reflections, image shifts, zooming and rotation.

2. Littoral Bottom Type Classification Using Multispectral Imagery and Neural Networks

A database of imagery with different bottom types was needed in order to be able to process and to be able to send to a neural network for training. Carmel River State Beach (CRSB) was selected as the site to conduct the collection of images, as there are a variety of different bottom types in the area. CRSB is located south of Monterey Bay, and is in the southern portion of Carmel Bay (Figure 4). As can be seen from the figure, there are natural rock formations on the northern point near Carmel-by-the-Sea and on the southern portion of Carmel Bay adjacent to the area where the Carmel River meets the beach. Kelp is generally always present in the area, as it is a staple of the littorals of the west coast of the United States. As noted previously, the presence of kelp indicates that there are rocks beneath, as kelp overwhelmingly adheres to rocky substrates the best, although there are cases of kelp growing from, and adhering to sand.

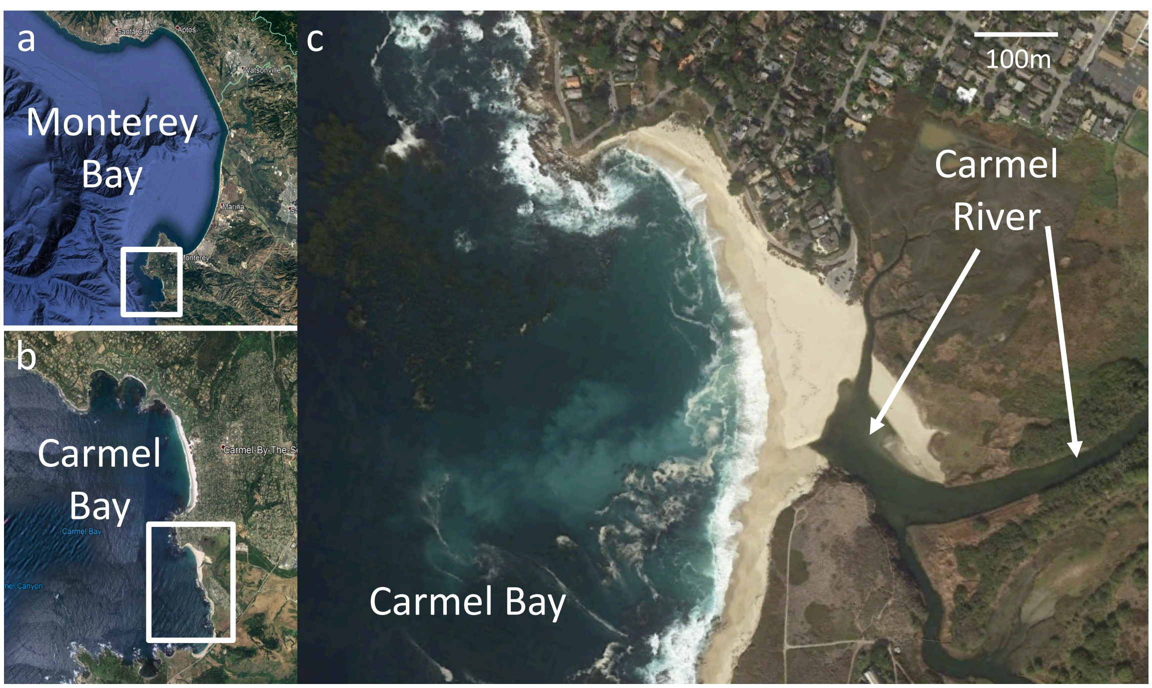

Figure 1. Carmel River State Beach, the site of image acquisition with relative location to Monterey and Carmel Bays (a,b)).

Over the course of two days, over 1200 images were captured of the area just offshore of CRSB. The Da-Jiang Innovations (DJI) Inspire 1 UAV was the platform used to acquire the images, and it was outfitted with the MicaSense RedEdge-M (RE-M), a professional 5-band multispectral camera that was designed for the agriculture industry to conduct precise vegetation mapping. The RE-M captures images in the Blue (475 nm center, 20 nm bandwidth), green (560 nm center, 20 nm bandwidth), red (668 nm center, 10 nm bandwidth), red edge (717 nm center, 10 nm bandwidth) and near-IR (840 nm center, 40 nm bandwidth) spectral bands. Flights with the RE-M were flown with the camera at near-nadir and with a 75% forward overlap.

The raw imagery from the RE-M is not registered, meaning that for every image capture, there are 5 panchromatic images, one image for each band (Green, Blue, Red, NIR, RE). Due to the RE-M having 5 lenses, that each capture one of the five bands, in different locations on the camera, each of the five images (of the one image) are collected at different geometry. Interactive Data Language (IDL), a scientific programming language also developed by Harris Geospatial Solutions, Inc., was used to register the images. IDL is the programming backbone of ENVI.

ENVI provides many ways to classify image data, including supervised and unsupervised methods. Supervised classification methods used training data to train the algorithm. The training data (within ENVI) consists of regions of interest (ROIs) that consist of similar groups of pixels that the user has identified. The supervised classification algorithms available are Maximum Likelihood, Minimum Distance, Mahalanobis Distance, Spectral Angle Mapper, and Support Vector Machine, including others. Unsupervised classification does not use training data to train the algorithm, and for this type, ENVI has 2 methods available, ISOData and K-Means statistical classifiers.

These are all viable options, but since the goal was to use deep learning and more specifically, a CNN, the ENVINet 5 model (that uses TensorFlow) was chosen to be the algorithm to train the images on. In order to create the training data, the images were hand labeled to identify the various features across each image and to build up a large enough library of each class of bottom type and feature of interest.

Initially, only the bottom types of sandy bottom, rocky bottom, and kelp were considered for classification, but further discussion led to the additional classifiers of sand, above ground rock and swash zones. This is due to the fact that since research is focused on the littorals and the imagery is so close to the shore, classification of the sand, above ground rock and swash zone would reduce the false positive identification of bottom types in the swash zone and sand areas of the beach. In essence, a full picture classification of the littoral area including the beach was determined to be the most beneficial to get a complete picture of the area.

To start the labeling process, the 2-D scatter plot tool within ENVI was utilized. This brings up a 2-D scatterplot which displays all of the pixels that are contained in the image on a 2-D plot, with 2 bands of the image as the x and y axes. In the figure below, Red is plotted as the x-axis and Blue is plotted as the y-axis. Different configurations were used to identify different areas, for example if trying to classify kelp, choosing NIR as the x-axis or y-axis would easily distinguish the pixels that contained kelp as kelp has the highest reflectance values of all other classification categories (sand, above ground rock, swash zone, sandy bottom, rocky bottom) in the NIR range.

In the 2-D scatter plot, groups of pixels are able to be circled, and these pixels then belong to a class which is differentiated by color. The corresponding pixels in the image are simultaneously given the same color as the pixels that were circled on the scatter plot, giving the user instant feedback on whether or not the circled pixels in the scatter plot need to be adjusted. For instance, if it is recognized that two different classes were labeled the same color in the image (i.e. sand and rocky bottom), this would indicate that the circle of pixels drawn in the scatter plot was too liberal and needs to be tightened.

Classification continued in this manner until all images were labeled. Once labeled, each individual class, within each image, was first saved as a ROI in a .xml file. This would facilitate the next step; creating labeled rasters from these ROIs and training the neural network.

In ENVI, the TensorFlow based model can be trained using either a label raster (the saved ROIs in .xml format) or a classification image, which can be obtained from the unsupervised or supervised classification methods mentioned above. The label rasters built from the ROIs contain all of the bands from the image as well as a binary mask that labels areas where the desired feature's presence is indicated by a value of 1 and where there is absence of the feature, a value of 0 is assigned. Before the model can start to train on the label rasters, it must be initialized with the desired parameters.

The first parameter to define is the patch size. A patch is a specified portion or "chip" of the image that is sent to the model for training. The larger the patch size, the faster training can occur because the model can get through each image quicker. The default patch size is 572 pixels, and can be made larger, but is limited by the size of the memory on the users' graphics card.

The next parameter is to indicate the number of bands in the image that the model will classify. For this study, we are interested in seeing how well the model can classify 5-band imagery compared to 3-band imagery, so we essentially will be running all of the model runs twice. We will run 5-band models that will be identify the classes of sand, swash zone, above ground rock, rocky bottom, and sandy bottom, then will run 3-band models to identify the same classes and compare.

Important to note is that even with the same training rasters as inputs to the model, successive model runs will not yield the same results. This is due to the inherent randomness of deep learning and convolutional networks. Just as the human brain contains millions of neurons creating infinite combinations of paths to-and-from dendrites, neural networks are fashioned in the same manner, and no two model runs will be identical. ENVI has an additional feature that is helpful regarding these additional parameters that allows you to randomize them if you are unsure about the best values to input.

The parameters discussed above are the only mandatory parameters that you must input, the rest are optional to input a value for, as ENVI will put default values in for them unless the user specifies otherwise. For this study, the default values were used for the number of epochs (25), patch sampling rate (16), number of patches per epoch (300), and number of patches per batch (9).

There were a total of 12 models created, 6 for each of the classes (sand, swash zone, above ground rock, sandy bottom, rocky bottom, kelp) utilizing all 5-bands of the imagery and another 5 models for each of the classes using just 3 bands (Blue, Green, Red). The output of the model is a class activation raster, which is a greyscale image in which the color of the pixels corresponds to the probability that the feature is present, with white being a positive match and areas where there is no match are black.

The models were constantly refined and retrained. By taking the class activation raster (the output of the model) and comparing it to the input raster, areas correctly identified the feature were put into a new ROI, saved into a .xml, and sent into another model run so that the accuracy of the model increased with each iteration.

Once the models were trained to a 98% accuracy level, the model was used to classify the images in our testing folders, which the models had never been trained on. The results of these model runs are described in the Section II.

3. Littoral Change Detection

Datasets for change detection are generally collected from airborne or space-borne assets. Change detection tends to be application specific, which poses a problem for researchers in the subject field. Some researchers augment their datasets by creating simulated imagery through bootstrapping. Bootstrapping is generating data from real data so that it has the same spectral characteristics. Some Siamese networks are trained to work with pixels through color differencing and some of the networks are trained to pick up on spectral bands. The networks trained to use spectral bands are of great use to the coastal classification research community because certain bands separate the land from the water as was stated earlier. Another trick performed in research is to augment datasets by rotating and flipping images. The images are rotated in increments of 90 degrees and the flipped horizontally and vertically.

There were several datasets found during the research for this thesis. Most of the datasets left by researchers are datasets that have to do with urban environments such as the SZTAKI and Onera Satellite datasets. Some researchers also stated that they used Google Earth images because they can view different time periods.

Siamese neural networks are special in that they have two input streams instead of one. This means that the architecture can take in two inputs at the same time to perform a task. In the case of this research the inputs will be images, but there are examples of other uses such as signature verification. On a historical note, the signature verification was developed back in the early 1990s at AT&T Labs. For a Siamese network, it is up to the researchers to determine when the network streams are combined before the desired final output is reached. The final output of the Siamese networks can be a simple binary answer or it can be a difference image showing the difference between images from different time periods.

The approached to training Siamese networks can be either supervised or unsupervised. In the supervised approach the networks are trained on a dataset specific to an application. In unsupervised learning the network has no training set and the goal is to extract features from the images and compare them, both approaches have their advantages.

The first example of using a Siamese network for change detection is a network used for detecting helicopters in images. The dataset used for research was augmented by bootstrapping. Bootstrapping was also used to increase performance of the network. "Bootstrapping was performed by collecting all the misclassified samples after one pass through the network reusing them along with new samples to continue training [the] network" (Rahman, Vasu, Van Cor, Kerekes, & Savakis, 2018). Constructing the Siamese network consisted of joining two open sourced VGG16 networks by a custom built two-layer decision network. The output for the change detection scheme consisted of a simple Boolean answer indicting whether or not a helicopter was detected in the target image. The goal of the study was to determine if a Siamese network could be trained using simulated imagery.

Another example of a network with a binary output is research about areas impacted by tsunamis. Two images are compared in a Siamese network and the intended output is to determine if buildings are washed away from the storm or not. The network output can then be overlaid on a map which would ultimately paint a picture for how much damage was created by the storm. Binary outputs are not the only thing that Siamese neural networks can produce. Some networks produce difference images that display the change between the two images.

Siamese networks can produce difference images by extracting features from the images and comparing them. The approach to compare features is a step up from the binary pixel difference maps. A similarity score can be generated between images to give a better metric for how much they changed.

With regard to landscape change detection many researchers construct difference images and then empirically apply other functions to the difference image to clean it up and produce an optical output. Several teams also make use a loss function to optimize their feature extraction techniques (Arabi, Karoui, & Djerriri, 2018) (Zhan, Fu, Yan, Sun, Wang, & Qiu, 2017).

Change detection is the process of identifying differences in the state of an object or phenomenon by observing it at different times (Singh, 1989). In our particular problem we are not attempting to identify specific objects in a picture but the entire picture itself – a coastal landscape.

This study builds upon a the models trained in section 1. The coastal image dataset is a set of orthomosaics (images compiled from several hundreds of UAS imagery) that were taken at Carmel River State Beach. This thesis will conduct a survey of Siamese open-source neural networks to find an optimal neural network that will maximize the change detection algorithm. The use of an open-source network is beneficial because only the input and output layers of the network will need to be trained. The layers in between are left alone. This thesis will not train a brand new neural network from scratch specific to the problem set.

Current research in the field of change detection attempts to discover differences in images and report those differences back to an expert for verification. This research differs because the goal is for the change detection results to tell something about how the classification of the coastal environment changed.

II. RESULTS

A. LANDSCAPE CLASSIFICATION USING VARIOUS NEURAL NETWORK ARCHITECTURES

Results of model accuracies from the two sets of experiments are summarized in Table 1. The 13 network models used in this study are reported with top-1 accuracies on ImageNet validation set ranging from 0.704 to 0.825 (ImageNet Keras, Table 1). These networks included older VGG architectures (Simonyan and Zisserman, 2014) and modern architectures such as Inception architectures (Szegedy et al., 2016; Szegedy et al., 2017) , Xception (Chollet, 2017), ResNet (He et al. 2016), MobileNets (Howard et al., 2017; Sandler et al., 2018), DenseNets (Huang et al., 2017) and NASNets (Zoph et al. 2018). The Keras-provided achieved accuracies (column 2, Table 1) are not identical to the best reported accuracies from the original authors of the models unless they have made their models publicly available in the different formats (e.g. Keras, PyTorch). As an example, ResNet50 reported Top-1 Accuracy on ImageNet Validation set (He et al. 2016), is 0.793 where the model checkpoints used here, as available on Keras application repository Chollet 2015, achieved reported accuracy of 0.749. In order to have an established starting point validation was run on all of the pretrained model checkpoints to verify the results reported on the Keras webpage and confirm the values in the Table 1 for Top-1 ImageNet Accuracy on the ImageNet released validation dataset ILSVRC2012.

Fixed feature extraction experiments on the coastal dataset provided results that varied in top-1 accuracy from 0.608 to 0.888 (Table 1, Experiment I). The two best performing models were Visual Geometry Group's 16 (VGG16) model and VGG19. These older architectures are 16-layer and 19-layer convolutional neural networks (Simonyan and Zisserman, 2014). Interestingly, the best performing architectures on ImageNet validation set InceptionResNetV2 and NasNetLarge, with deeper architectures (572 for InceptionResNetV2 and unreported for NasNetLarge), were the least accurate on the coastal test dataset. The worst performing model was MobileNetV2 (88 layers deep) that achieved an accuracy of 0.608.

In the second set of the experiments the models were fine-tuned (Table 1, Experiment II). In this case, the best performing model is DenseNet121 with 0.946 top-1 accuracy on CostalData test set and the model with lowest accuracy is MobileNetV2 with 0.905 accuracy. However, relative difference in the accuracies achieved after the fine-tuning is only 4.3% suggesting that all of the used architectures transfer well to the CostalData set. This is in contrast to the Experiment I where the relative difference in accuracies was 28.6%.

The result of these two experiments are normalized confusion matrices (Figure 2), where the blue diagonal squares show the accuracy of the model using the testing images sets for each coastal class (only shown for the best fixed-feature and best fine-tuned models for example). Perfect accuracy would be a diagonal matrix with all 1.0 values on the diagonal. The effect of fine-tuning on DenseNet121 is shown by an increased diagonality in the confusion matrix (compare Figure 2a with 2b). This shows that by allowing the features to be fine-tuned, the model better-learns representations that are specific to the coastal dataset. This is consistent with newer architectures learning hierarchical representations within datasets more efficiently. In contrast, for older models, such as VGG16, fine tuning does not greatly improve model accuracies, suggesting that these older architectures that are not as deep do not have the capability of learning highly complex feature representations.

Table 1. Summary of model accuracies

Model	**top-1 accuracy ImageNet Keras (reported)**	**top-1 accuracy Experiment I (fixed features)**	**top-1 accuracy Experiment II (fine-tuned)**
Xception	0.790	0.689	0.919
VGG16	0.713	0.888	0.919
VGG19	0.713	0.843	0.938
ResNet50	0.749	0.794	0.921
InceptionV3	0.779	0.749	0.940
InceptionResNetV2	0.803	0.655	0.933
MobileNet	0.704	0.765	0.920
MobileNetV2	0.713	0.671	0.905
DenseNet121	0.750	0.774	0.946
DenseNet169	0.762	0.824	0.939
DenseNet201	0.773	0.779	0.944
NasNetMobile	0.744	0.608	0.939
NasNetLarge	0.825	0.670	0.908

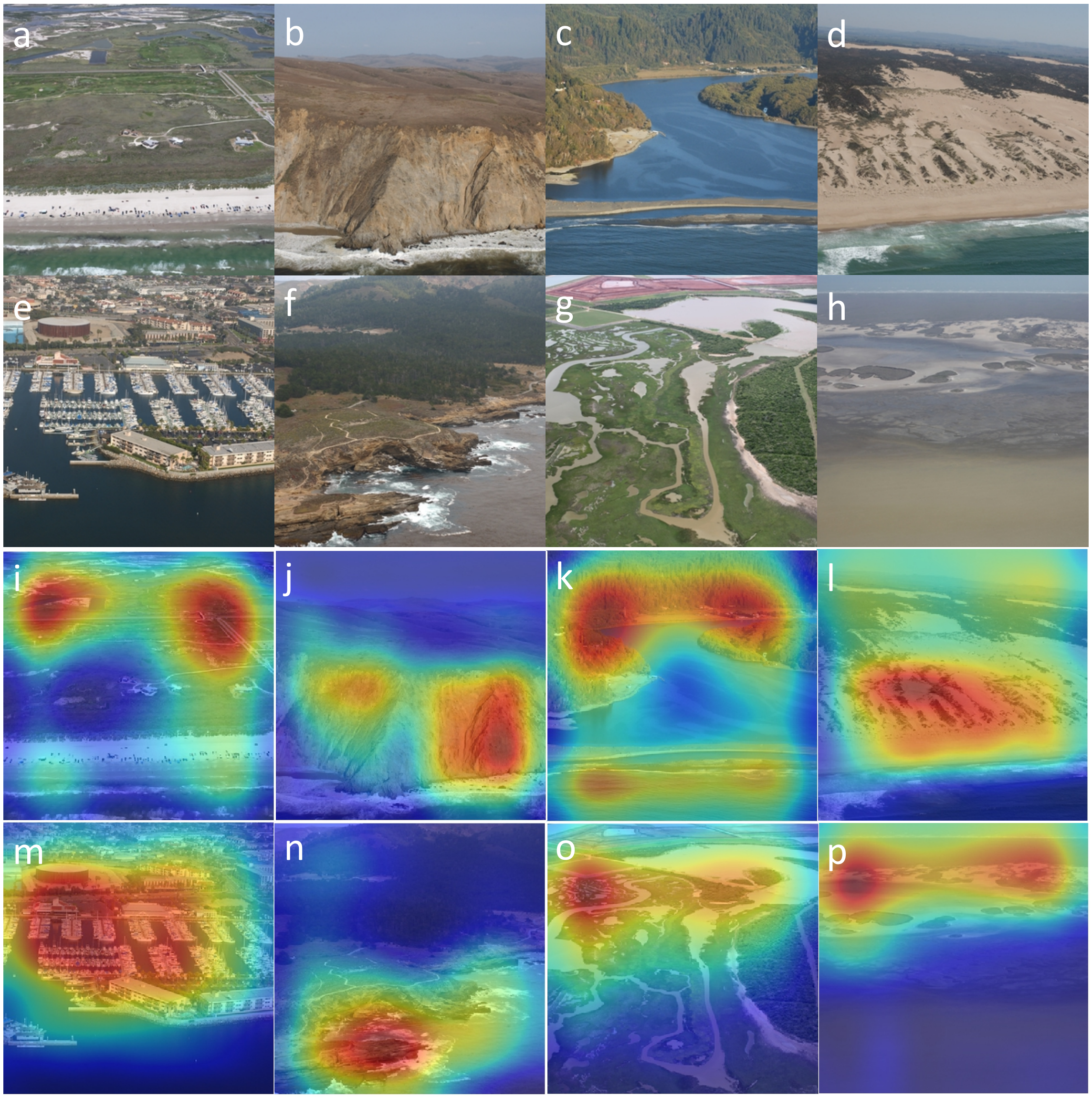

Figure 2. Class examples from test dataset: a) Beach (sandy), b) Cliff, c) Coastal waterway, d) Dune, e) Man-made structure, f) Rocky coast, g) Marsh, h) Tidal flats. Panels i-p show the same images in a-h, but with the Grad-CAM heatmap (red = high, blue = low) overlaid from fine-tuned DenseNet121, depicting regions in each image that specify where the class exists. All probabilities for i-p were above 0.99 for the represented class.

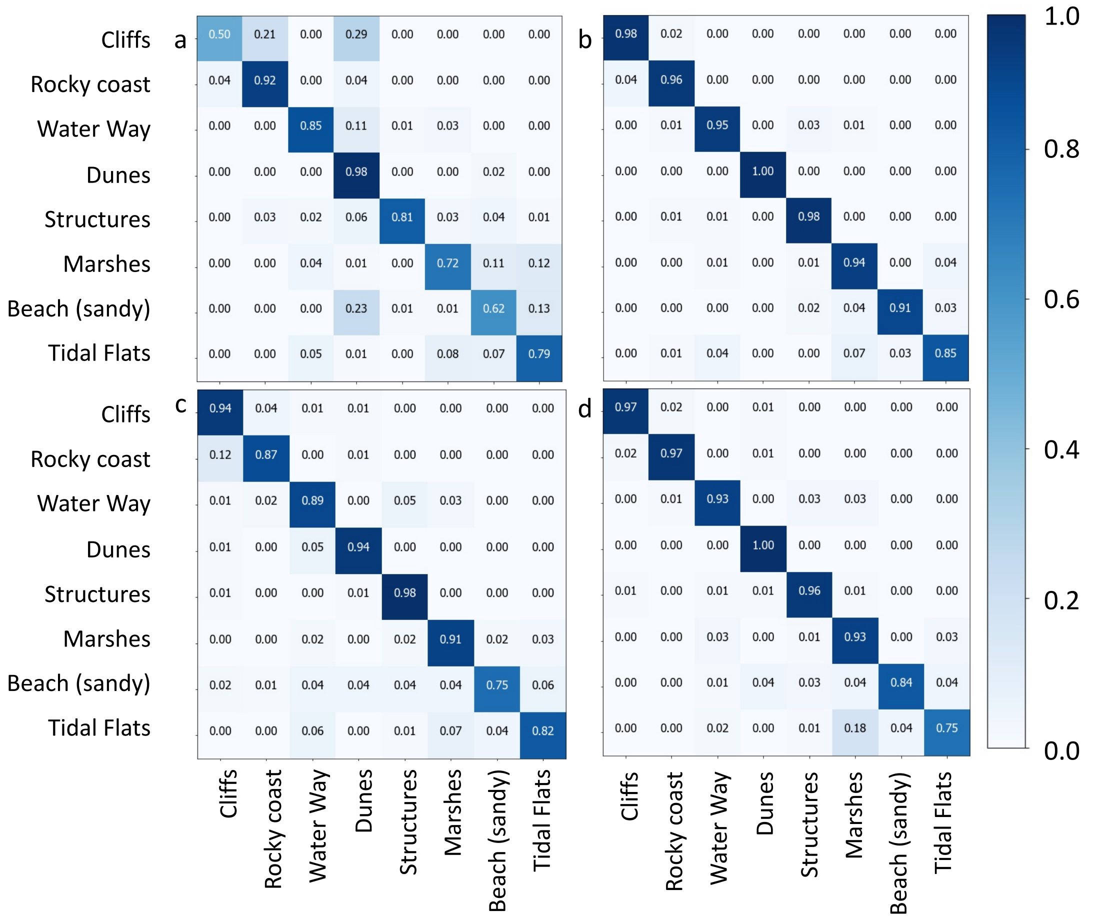

Figure 3. Confusion Matrices. Top (a, b) are DenseNet121 with fixed feature (a) and fine-tuned (b). Bottom (c, d) are VGG16 with fixed feature (c) and fine-tuned (d). This matrix is normalized by the number of images in each class to denote a percent accuracy. All classes had 100 images to test.

Another way of highlighting class features is by clustering algorithms such as t-SNE (Maaten and Hinton, 2008) that take multi-dimensional feature representations and projects this information into a 2D space such that clustering of images with similar features shows class similarity. For the coastal dataset, Figure 3 shows the t-SNE clustering method for the same models as shown in Figure 2, DenseNet121 and VGG16. Both fixed feature clustering for DenseNet121 and VGG16 (Figures 3a and 3c, respectively) show considerable spread over all eight classes, although there is still distinct class similarity. For VGG, there is more class overlap, indicated by non-uniform cluster space. The effects of fine-tuning are best shown by DenseNet121, similar to in the confusion matrices (Figure 2a and b), there is obvious class distinction in the fixed-feature model that is enhanced in the fine-tuned model (compare Figure 3a with 3b).

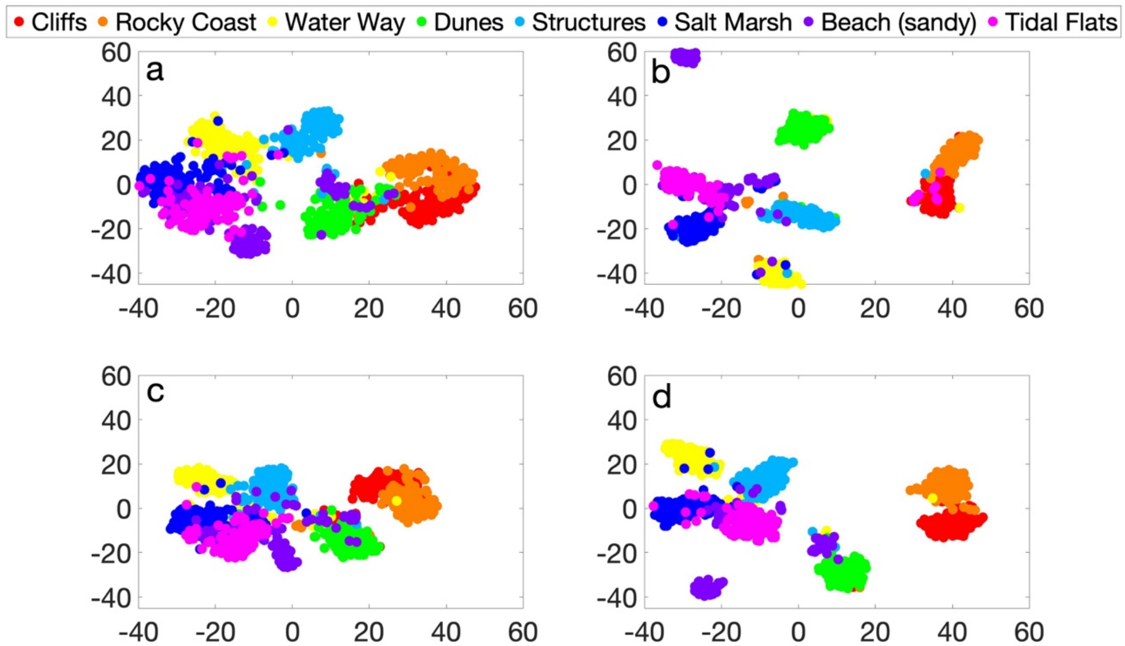

Figure 4. Clustering results using the t-SNE algorithm. Similar models to Figure 2, with a) DenseNet121 fixed feature, b) DenseNet121 fine-tuned, c) VGG16 fixed feature, and d) VGG16 fine-tuned. Low dimensional embeddings of CoastalData test set using t-SNE on the penultimate layer of networks in a), b), c) and d) for 8 classes. Proximity shows similarity of each image, represented by a single dot from the test dataset of 100 images per class.

In order to provide visual explanations for classification decisions of the trained models (go beyond black-box) a Gradient-weighted Class Activation Mapping (Grad-CAM, Selvaraju et al. 2017) algorithm is applied to DenseNet121 and VGG16 for the eight classes (Figure 1i-p). The Grad-CAM application to both of the models produces a localization heatmap of important regions in the image for predicting each of the classes.

The results indicate that using deep neural networks, specifically through transfer learning, is an effective method to classify heterogeneous coastal landscapes without the need for semantic segmentation. Despite the heterogeneity of the training dataset and its relative small size compared to ImageNet (10000 images vs. over 10 million images), there is enough difference between each coastal class implying correct identification is possible. This method shows potential for automatic classification of coastal landscapes, which can help increase efficiency for identifying areas of change. Specifically, large datasets with high overlap collected from UAS or airplane could be rapidly classified to show regional changes in coastal landscape.

Similar performance degradation between the best reported ImageNet top-1 accuracies and the performance of publicly available model checkpoints of the best reported architectures such as DenseNets and ResNet on fixed feature extraction tasks was recently reported by Kornblith et al., 2019. This investigation focused on the correlation between the ImageNet accuracy and transfer accuracy with fixed features and found low correlation between the two similar to this study (compare top-1 accuracy ImageNet and Experiment I of Table 1). It was demonstrated, on the example of Inception models, that the some of the choices made in the training process of modern architectures such as: the absence of scaling parameter for batch normalization, use of label smoothing, use of dropout, and the presence of auxiliary classifier head are detrimental to transfer accuracy. However, older style architectures (VGG16, VGG19), do not use any of the above-mentioned features of the modern training architectures and are expected to better transfer learn in the fixed feature extraction configuration.

One of the focuses of this study was to use an ontology that would provide meaningful classification results, with the goal that the enormous unlabeled aerial imagery datasets could be easily sorted. Therefore, one possible assessment on model skill is diversity of features between separate classes. Considering the confusion matrices (Figure 2) and clustering of features (Figure 3), there is evidence that most classes remain distinct, such that CNNs are capable of determining differences. Both clustering and confusion matrices indicate some classes with a higher likelihood for misclassification, based on feature similarity. For example, "coastal cliffs" and "rocky coast" are often similar in both classification metrics. The class of "Beach (sandy)", a landscape that is inherently without strong features, there is a tendency to mis-classify. This is likely owing to other areas of the image containing feature-rich areas that could contain another class. Evidence of strong class features is indicated by similar heat maps (Figure 1i-p) for varying models, such as "rocky coast", "man-made structures", and "marsh" in contrast to classes with weaker features, such as "Beach (sandy)". It should be noted that there is slight variance between the heatmaps for various models, but the general trend exists that each class can be identified.

In addition, the motivation for single labeling heterogeneous landscapes for use in deep learning is more robust to perturbations in pixel-level fluctuations seen in semantic segmentation methods. This implies that this method would minimize that sensitivity to changes at specific locations and identify global-scale class features.

B. LITTORAL BOTTOM TYPE CLASSIFICATION USING MULTISPECTRAL IMAGERY AND NEURAL NETWORKS

1. Results of Year 1

This project is currently in progress and is the thesis of LCDR Ash Mielke in the Oceanography Department at NPS (graduating December 2019). To date, a new dataset of multispectral (visible, near infrared, and RedEdge) data has been curated and spectrally analyzed. The hypothesis to be tested is that multispectral datasets provide critical signals that can help with deep learning classification of various landscapes. As an example, Figure 5 shows how the near infrared and RedEdge bands are critical for enhancing vegetation, such as kelp, within the nearshore.

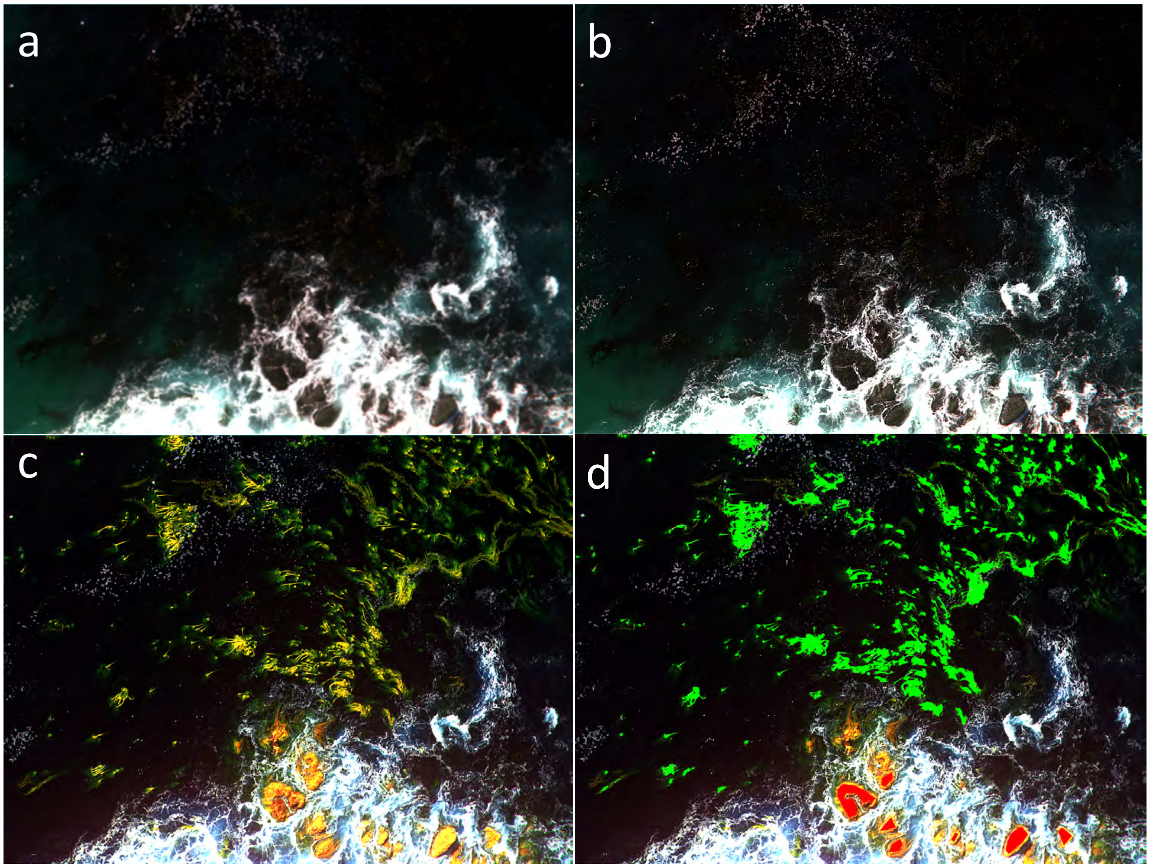

Figure 5. Coastal littoral example off Carmel River State Beach showing areas of rocks, kelp and sand. a) raw image RGB (Red, Green, Blue), b) sharpened image, c) false colored with bands Red, Near Infrared, and RedEdge, d) raster shapes of kelp (green) and vegetated rock (red).

2. Ongoing Work (Student Thesis)

LCDR Mielke will continue class segmentation using the 5-band data collected at Carmel River State Beach. His goal is to use the ENVI platform to train individual neural network models for the classes: kelp, rock, sand, and breaking wave. This is an iterative process, and will result in individual models capable of predicting likelihood of each class.

C. LITTORAL CHANGE DETECTION

1. Results of Year 1

This project is currently in progress and is the thesis of Capt. Theodore Ayoub in the Computer Science department at NPS (graduating March 2020). To date, large orthomosaics for Carmel River State Beach are used as the dataset showing coastal change owing to beach physical processes breaching the beach (Figure 6 a versus 6 b). Each orthomosaic is larger in scope than the aerial imagery used to train the neural networks (section A), so each is segmented into scaled images on the same scope as the training dataset. This creates a moving filter, and each sub-filter is sequentially tested using both VGG16 and DenseNet121 for comparison. The results for each class are stored as a percentage class likelihood, creating grayscale pixelated maps of the same dimensions as the original images, where white is 100% likely and black is 0% likely that the class is represented.

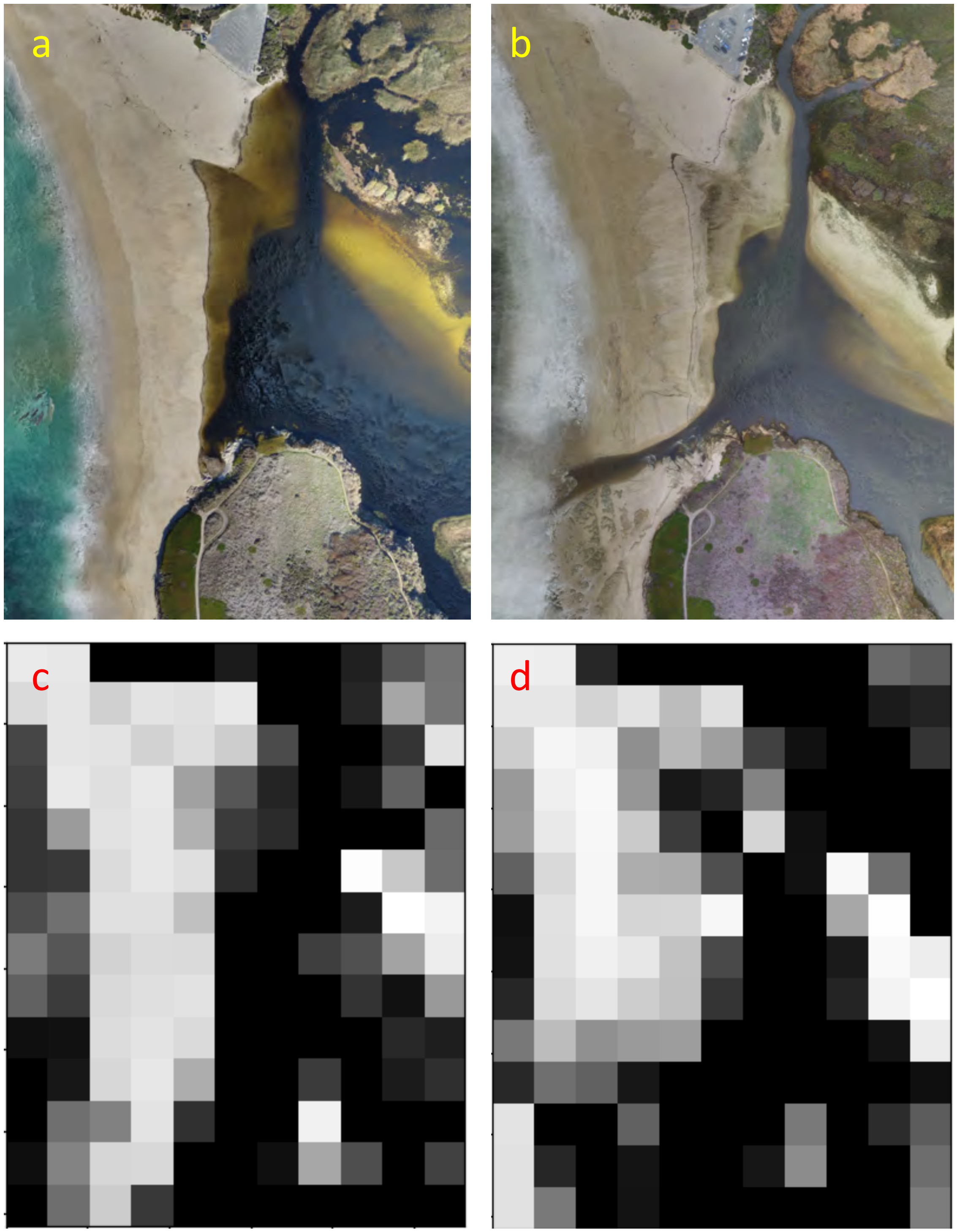

Figure 6. Change detection for Carmel River State Beach using VGG16. Orthomosaics are from a) December 2017 and b) January 2018. North is vertical and scale from top to bottom is 1km. The back lagoon overtopped the beach on January 11, 2018 creating the river to the bottom half of the picture in b). Class prediction maps (white = 100% likely, black = 0% likely) for the tidal flats class (wet sand).

From December to January, there is a noticeable pattern for tidal flats (wet sand) that matches the overall shape of the beach during each time. In particular in the southwest quadrant (bottom left), it is possible to see a change away from tidal flat to another class, consistent with the location of the River removing part of the beach.

2. Ongoing Work (Student Thesis)

During the remainder of his thesis, Capt. Ayoub will convert the neural network architectures into a Siamese network framework to quantify the difference between the class predictions during December and January at Carmel River State Beach (i.e. assess relative change between Figures 5c and 5d). The goal is to formulate an approach that will account for meaningful morphodynamic change (river breaching) in contrast to situational changes owing to variations in color, location of cars/etc., or smearing of breaking waves, to name a few.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SUMMARY

A. LANDSCAPE CLASSIFICATION USING VARIOUS NEURAL NETWORK ARCHITECTURES

It was possible to use heterogeneous coastal landscapes to train deep neural networks through transfer learning with a high degree of accuracy (>90% correct identification) for 13 CNNs without the need for image semantic segmentation. Furthermore, the effects of fixed feature (Experiment I) versus fine-tuning (Experiment II) in CNN architectures suggests that there is adequate class distinction using NOAA's ESI landscape classification ontology. However, for smaller datasets (100s of images) it is best to use fixed feature extraction (and hence older architectures) to reduce the tendency of overfitting. By developing CNN models without segmentation requirements, it will be possible to provide rapid coastal landscape assessments for large aerial image datasets, which will show both longer-term evolution as well as rapid changes post extreme events.

B. LITTORAL BOTTOM TYPE CLASSIFICATION USING MULTISPECTRAL IMAGERY AND NEURAL NETWORKS

Multispectral data (RGB, RedEdge, and Near Infrared) was collected by UAS at Carmel River State Beach during summer 2019. This dataset has been annotated into class-specific categories of kelp, sandy bottom, rocky bottom, above ground rocks, sand, and breaking waves to provide meaningful labels to littoral landscapes. The goal is to provide a deep learning method for identifying coastal zone obstacles for littoral operations. It is expected that multispectral data augmented with feature extraction using neural networks will provide the best classification capabilities over coastal waters. This project is the thesis of LCDR Ash Mielke, who will graduate from the Oceanography Department at NPS in December 2019.

C. LITTORAL CHANGE DETECTION

The curated coastal landscape database presented in section A is being used to develop a Siamese neural network architecture for change detection at Carmel River State Beach. The goal of this project is to have the model only detect change that is morphologically relevant to the coastal zone: identify beach breaching versus expected changes owing to time of day/time of year. To date, the trained models have been tested on a moving window of two large orthomosaics (UAS large images from multiple smaller images). These class expected values show change between closed and breached beaches, and are currently being developed by Capt. Teddy Ayoub for his thesis. Capt. Ayoub will graduate in March 2020 from the Computer Science department at NPS.

D. RECOMMENDATIONS FOR FURTHER RESEARCH

Given the preliminary success of multiple deep learning architectures for classification, a longer-term goal is to establish an easy-to-use and automated methodology to detect change and damage to coastal high-impact assets. By creating an deep learning-driven assessment of coastal change, a time-efficient and consistent method will be created capable of assessing any coastal feature and performing a change detection to determine fundamental changes in environment.

Specifically, it is recommended that an expansive dataset be curated to expand upon the California dataset so that model capabilities are not region specific. This will require further testing/training of neural network architectures, but will help solidify accuracies and transferability.

LIST OF REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation* ({OSDI} 16) (pp. 265-283).

Agrawal, P., Girshick, R. and Malik, J., 2014, September. Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision* (pp. 329-344). Springer, Cham.

Buscombe, D. and S.C Ritchie, 2018: Landscape classification with deep neural networks. *Geosciences*, **8**,(244), 1–23.

Chen, D. and D.A., Stow, 2002: The effect of training strategies on supervised classification at different spatial resolution. *Photogrammetric Engineering and Remote Sensing*, **68**, 1155–1162.

Chollet, F., 2015. Keras.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

d'Oleire-Oltmanns, S., I. Marzoff, K.D. Peter, and J.B. Ries, 2012: Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco. *Remote Sensing*, **4**, 3390–3416.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T., 2014, January. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655).

Dunford, R., K. Michel, M. Gagnage, H. Piegay, and M.L. Tremelo, 2009: Potential and constraints of unmanned aerial vehicle technology for the characterization of Mediterranean riparian forest. *Int. J. Remote Sens.*, **30**, 4915–4935.

Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. MIT press.

Han, S., H. Mao, and W.J. Dally, 2016: Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *ICLR* 2016.

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

Hoonhut, B.M., M. Radermacher, F. Baart, and L.J.P. van der Maaten, 2015: An automated method for semantic classification of regions in coastal images. *Coastal Engineering*, **105**, 1–12.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

Hubert-Moy, 2001: A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote Sensing of Environment*, **75**, 174–187.

Hugenholtz, C.H., and K. Whitehead, 2014: Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: a review of progress and challenges. *J. Unmanned Veh. Syst.*, **2**, 69–85.

Joyce, K.E., S.E. Belliss, and S.V. Samsonov, 2009: A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Progress in Physical Geography*, **33(2)**, 183–207.

Kornblith, S., Shlens, J. and Le, Q.V., 2019. Do better imagenet models transfer better?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2661-2671).

Krizhevsky, I. Sutskever, and G.E. Hinton, 2012: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp.1097-1105.

Kuleli, T., A. Guneroglu, F. Karsli, and M. Dihkan, 2011: Automatic detection of shoreline change on coastal Ramsar wetlands of Turkey. *Ocean Engineering*, **38**, 1141–1149.

Landgrebe, D.A., 2003: Signal Theory Methods in Multispectral Remote Sensing, Hoboken, NJ: John Wiley and Sons. 528pp.

Lu, D., and Q. Weng, 2007: A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, **28:5**, 823–870.

Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2016). Convolutional neural networks for large-scale remote-sensing image classification. IEEE Transactions on Geoscience and Remote Sensing, 55(2), 645-657.

Maiti, S. and A.K. Bhattacharya, 2009: Shoreline change analysis and its application to prediction: A remote sensing and statistics based approach. *Marine Geology*, **257**, 11–23.

Marmanis, D., M. Datcu, T. Esch, and U. Stilla, 2015: Deep learning earth observation classification using ImageNet pretrained networks. *IEEE: Geoscience and Remote Sensing Letters, Vol. 13.* No. **1**, January 2016.

Mas, J.F., 2004: Mapping land use/cover in a tropical coastal area using satellite sensor data, GIS and artificial neural networks. *Estuarine, Coastal and Shelf Science*, **59**, 219–230.

Mather, P.M., 2004, Computer Processing of Remotely-Sensed Images: An Introduction, 3rd edn., Chichester: John Wiley & Sons. 442pp.

Morgan, K.L.M., Krohn, M.D., and Guy, Kristy, 2015, Post-Nor'Ida coastal oblique aerial photographs collected from Ocean City, Maryland, to Hatteras, North Carolina, December 4, 2009: U.S. Geological Survey Data Series 930 http://dx.doi.org/10.3133/ds930

Natesan, U., A. Parthasarathy, R. Vishnunath, G. Edwin Jeba Kumar, and V.A. Ferrer, 2015: Monitoring long term shoreline changes along Tamil Nadu, India using geospatial techniques. *Aquatic Procedia*, **4**, 325–332.

NOAA. (2013). Shoreline Assessment Manual. 4th Edition. U.S. Dept. of Commerce. Seattle, WA: Emergency Response Division, Office of Response and Restoration, National Oceanic and Atmospheric Administration. 73 pp + appendices.

Rango, A., A.S. Laliberte, J.E. Herrick, C. Winters, K.M. Havstad, C. Steele, and D.M. Browning, 2009: Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management. *J. Appl. Remote Sens.*, **3**, 1–15

Salamati N., Larlus D., Csurka G., Süsstrunk S. (2012) Semantic Image Segmentation Using Visible and Near-Infrared Channels. In: Fusiello A., Murino V., Cucchiara R. (eds) Computer Vision – ECCV 2012. Workshops and Demonstrations. ECCV 2012. Lecture Notes in Computer Science, vol 7584. Springer, Berlin, Heidelberg

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510-4520).

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 618-626).

Simonyan, K. and A. Zisserman, 2014: Very deep convolutional networks for large-scale image recognition. *ICLR* 2015.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), pp.1929-1958.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Tamassoki, E., H. Amiri, and Z. Soleymani, 2014: Monitoring of shoreline changes using remote sensing (case study: coastal city of Banddar Abbas). *IOP Conf. Series: Earth and Environmental Science*, **20**, 1–7.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320-3328).

Zoph, B., Vasudevan, V., Shlens, J. and Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697-8710).

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Research Sponsored Programs Office, Code 41
   Naval Postgraduate School
   Monterey, CA 93943

4. HQMC Intelligence Department
   Geospatial Intelligence Branch Deputy, Mr. Daniel Hannigan
   Daniel.g.hannigan@usmc.mil
   860 271-2659

5. US Coast Guard Research and Development Center
   Director of Research Partnerships/Public Affairs Officer, Dr. Joe DiRenzo
   Joseph.DiRenzo@uscg.mil
   860-271-2738