Quantifying Intertidal Zone Species Using Semantic Segmentation

Mitali Shah
Computer Science Department
CSU Channel Islands
Camarillo, CA USA

Geoffrey Dilly
Biology Department
CSU Channel Islands
Camarillo, CA USA

Kaylen Meeker
Biology Department
CSU Channel Islands
Camarillo, CA USA

Jason T. Isaacs
Computer Science Department
CSU Channel Islands
Camarillo, CA USA

***Abstract*—As anthropogenic impacts on marine ecosystems accelerates (e.g. warming, acidification, eutrophication, etc), it is essential to build robust datasets that establish biological baseline data and capture long-term trends in shifting species abundance and diversity. This data has traditionally been collected through continual revisits by skilled ecologists and taxonomists to long-term ecological monitoring sites. One novel technique developed by an intertidal ecology research group at California State University Channel Islands (CSUCI) builds 1m-wide photo-transects for the length of the tidal zone (20m from splash to low zone) at two sites on Santa Rosa Island. These photos are stitched together using software and offer high-resolution swaths of information at the island, taken twice a year. A machine learning technique, semantic segmentation, has been employed to automate the analysis of these large images, focusing first on a dominant algal species of rockweed *Silvetia compressa*. This automation will greatly reduce the time needed and human error involved in scoring and quantifying these transects. The study involves developing a convolutional neural network using transfer learning on a publicly available network.**

***Index Terms*—semantic segmentation, transfer learning, convolutional neural network**

1. Introduction

Ecological monitoring aims to track and identify causes of ecosystem shifts by measuring ecosystem state variables in space and time [1]. Ecological monitoring in rocky intertidal zones, helps to track the species assemblage and their biodiversity. This helps researchers to better understand the species, changes that occur in their assemblage over time and the factors that cause those changes, and to make informed decisions pertaining to the ecological balance and marine conservation [2]. Monitoring programs have traditionally focused on discrete measurements such as point intercept or transects. However, swath methods such as photo transects can provide more detailed information on the ecological diversity and abundance of an intertidal site. One such monitoring of the rocky intertidal zone of Santa Rosa Island is shown in the Figure 1. This figure shows the variation in the assemblage of species over a period of three years. During the winters of 2016 and 2017 the area was dominated by two species *Phragmatopoma californica* and *Phyllospadix sp.*; however, in winter 2018 *Phyllospadix sp.* was largely displaced by *Silvetia compressa*. *Phragmatopoma californica* was present in greater abundance in winter 2018 as compared to winter 2017.

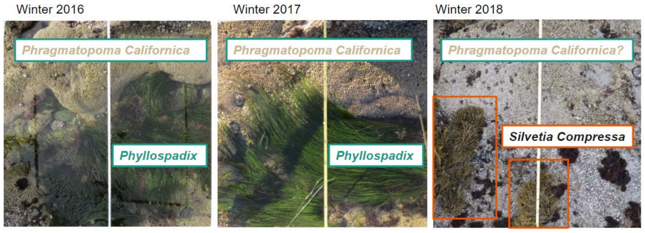

Winter 2016
Winter 2017
Winter 2018
Phragmatopoma Californica
Phyllospadix
Phragmatopoma Californica
Phyllospadix
Phragmatopoma Californica?
Silvetia Compressa

Figure 1. Changes in assemblage of species over time at Bechers Bay, Santa Rosa Island Intertidal Zone, CA.

Ecological monitoring can be done using point intercepts, vertical transects or photo transects. The point intercept method provides clustered data with good resolution of the species in specific randomly distributed plots in each zone (e.g. low, mid, high). The vertical transects method gives a low resolution view of the site but allows researchers to collect data evenly throughout the site. The photo transects method gives the highest amount of resolution of the site, but the data from this method can extremely difficult to process.

In the present study, photo transects are used as the source of information. Small sections of the region are captured in each image. These images collect an abundant amount of ecological data for a region, provide a snapshot in time of the state of the region, and quantify the presence of each species in the region. Quantifying images, allows researchers to establish a biological baseline and track any changes that occur in the species abundance and diversity over time.

Typically, these images are quantified by skilled taxonomists and ecologists, allowing for accurate identification of the species assemblage in each site. But quantifying

`mitali.shah900@myci.csuci.edu`, `geoffrey.dilly@csuci.edu`, `kaylen.meeker961@myci.csuci.edu`, `jason.isaacs@csuci.edu`

978-1-5386-7568-7/18/$31.00 ©2018 IEEE

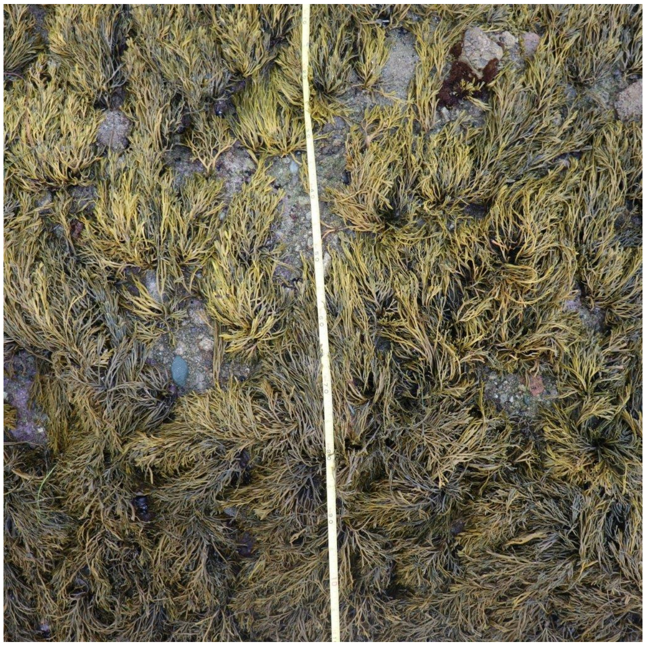


Figure 2. Sample image of algae species - *Silvetia compressa*.

huge amounts of data manually can be time-consuming and will likely lead to human error. In recent years, the machine learning research community has developed many techniques to address the problems that arise in manual quantification.

The goal of this work is to perform semantic segmentation on the photo transects containing a specific algae species, *Silvetia compressa*, as in Figure 2 found in two rocky intertidal zones in Santa Rosa Island - Bechers Bay and Skunk Point.

A brief description about the motivation of this work has been provided. The remaining sections are arranged as follows: Section 2 gives the background knowledge needed to understand the functionality of the present study. A thorough study and reference to techniques currently used and those used in the past are also included. Section 3 introduces the dataset and techniques used to label the training images. Section 4 describes the model architecture used in the present study. Section 5 demonstrates the implementation of the solution provided by the present study. Section 6 presents the results and a description of the metrics used to evaluate the system. Finally, Section 7 provides the concluding remarks and possible extensions of this work.

2. Related Work

Semantic segmentation is understanding the image at the pixel level, i.e., each pixel of the image is labeled with the object class it belongs to. Here an image is trained along with the image mask that contains the part of the image concerned (foreground, and the remaining is background).

Before deep learning, approaches like TextxonForest [3] and Random Forest [4] based classifiers were used for semantic segmentation. Fully Convolutional Networks (FCN) for Semantic Segmentation [5], was a Convolutional Neural Network (CNN) proposed for semantic segmentation. It was a dense network without a fully connected layer producing segmented images. Using a CNN for segmentation was not desirable because the pooling layers in the network increased the field of view and collected the information but discarded the location of the information. However, in semantic segmentation, the source of the information is necessary, so networks were evolved to tackle this problem such as the encoder-decoder network, the use of Conditional Random Fields (CRF), or the use of dilated convolutions.

In an encoder-decoder network, the encoder gradually reduces the spatial dimension with pooling layers and the decoder gradually recovers the object details (through the connections between encoder and decoder) and spatial dimension. There are usually short cut connections from encoder to decoder to help the decoder recover the object details better. CRFs are graphical models that smooth segmentation by observing that similar intensity pixels tend to belong to the same class. CRF post-processing is used after segmentation.

FCN and SegNet [6] were two initial encoder-decoder architectures. Since then several networks have been developed specifically for semantic segmentation. Multi-Scale Context Aggregation by Dilated Convolutions [7] and DeepLab [8] were based on dilated convolutions that performs convolution operations with a modified (wider) kernel. U-Net [9] is an encoder-decoder network that has demonstrated success working on a small number of bio-medical images. DeepLab v3 [10] has also demonstrated success in semantic segmentation by modifying the previous versions of DeepLab and using DenseCRF post-processing.

Training of all the above-mentioned networks required huge datasets except U-Net. But in many domains very few data samples are available for training. The problem with a small dataset is that it leads to over-fitting and reduces the accuracy of the network. To overcome this problem data augmentation, dropout [11] and transfer learning [12] were evolved.

Transfer learning is used to take the knowledge learned in one model and apply it to another task. This helps to use existing networks without worrying about the large dataset and computational power required to train the network. There are three major transfer learning scenarios:

- CNN as a fixed feature extractor: In this method a CNN pre-trained on an existing dataset is used. The last fully-connected layer is removed, and the remaining network is treated as a fixed feature extractor for the new dataset.
- Fine-tuning the CNN: This method, not only involves replacing and retraining the top layers of the CNN, but also fine-tuning the weights of the pre-trained network. All layers can be fine-tuned or a higher level portion of the network is fine-tuned while keeping the earlier layers fixed.
- Pre-trained models: Since it takes time to train a

CNN, some people release the model weights of their CNN trained on the state-of-the-art datasets which can be used by others on their datasets.

CNN features are more generic in the early layers and more original dataset specific in the later/higher layers. The selection of the transfer learning depends on various factors, but the size of the dataset and its similarity to the original dataset are the most important ones.

- If the new dataset is small and similar to the original dataset then using CNN as feature extractor is beneficial to avoid over-fitting.
- If the new dataset is large and similar to the original dataset then fine-tuning is used.
- If the new dataset is small and different from the original dataset then it is better to train the SVM classifier using activations from earlier layers.
- If the new dataset is large and different from the original dataset then fine-tuning partially-completely is appropriate.

Several networks have been successfully created using transfer learning [13], [14] and [15].

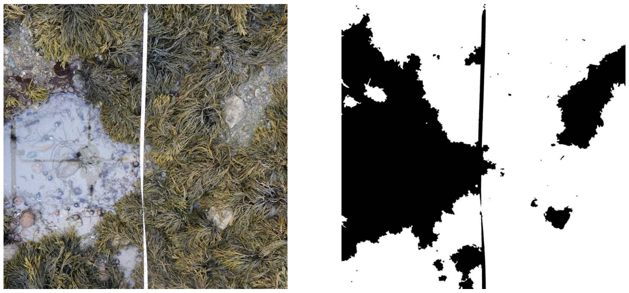
3. Dataset

This section provides details about the datasets used in the current study. It contains information like where and how the images were captured and the preprocessing techniques used on the images for machine learning.

The images of various species were collected from photo transects at two rocky intertidal zones in Santa Rosa Island - Bechers Bay and Skunk Point. These images were captured using an SLR camera. Each image is an RGB image with a resolution of approximately 3400×3400. The phototransects comprise eleven total swaths per site per season, each 20 m long and spaced 3 m away from one another. A rig of 1 m × 1 m was used to capture a single image and 58 such images were captured every 35 cm in order to create 65% overlap. These images were later restitched to create a 1 m × 20 m image of the entire photo transect. All images were captured during low tide, and preferably in daylight hours. However, the low tides in these zones can occur during the night. Due to this, some images were captured using flash photography and with this variability in the lighting conditions different colors of the same species were observed.

The images contain nine different species namely *Mytilus californianus*, *Silvetia compressa*, *Phragmatopoma californica*, *Phyllospadix sp.*, *Endocladia murcata*, *Ulva sp.*, *Anthropleura sola* and Red Algae. The present research study focuses on *Silvetia compressa*. A total of 592 images having dominant species as *Silvetia compressa* were taken. These images contain *Silvetia compressa* along with other species.

A total of 150 images were annotated out of which 100 were used as training and validation datasets and remaining were used as ground truth images to analyze the predicted mask of the test images. Annotated images (images containing only the foreground object i.e. *Silvetia compressa*) were created. Out of the training images only 85 images and their labels (annotated images) were used for training and 15 images and their labels were used for validation. The entire dataset was randomly split into training and validation datasets while training the model. Figure 3 shows an example of a labeled image. The left image is the original image and the right image contains the segmented image which is a binary image with white pixels indicating foreground (*Silvetia compressa*) region and black pixels indicating the background region. For creating image labels, the Image Segmenter App in Matlab Image Processing Toolbox was used.



(a) The original image containing Sivetia Compressa.

(b) The resulting annotated image.

Figure 3. An example of the image labeling process.The annotated image created using Image Segmenter App. The white pixels indicate foreground region and the black pixels indicate the background region.

The Image Segmenter App provides many different ways to annotate an image. The Graph-cut method was used in the present study. Graph-cut is a semi-automatic segmentation technique used to annotate images into foreground and background elements. To mark foreground and background elements, lines called scribbles are drawn. Based on the scribbles, the image software completes the segmentation.

The segmented image might have some imperfections, so morphological tools like dilation and erosion were used to fix the imperfections and to create a well-defined border. The segmented binary image was then stored and used as a label while training the model.

4. Model Architecture

The present study uses the existing U-Net model architecture and performs fine-tuning. U-Net was developed to perform semantic segmentation on microscopy images. As mentioned earlier, U-Net is an encoder-decoder architecture with skip connections. It consists of an encoder (contracting) path and a decoder (expansive) path.

The left part follows a typical CNN architecture with repeated application of two 3×3 convolutional layers, each having an ReLU activation function, and followed by a 3×3 pooling layer with the max pooling operation having a stride of 2. Downsampling operations are performed in this part. Downsampling is a max pooling layer which is an operation that summarizes each neighborhood of 2×2 neurons with

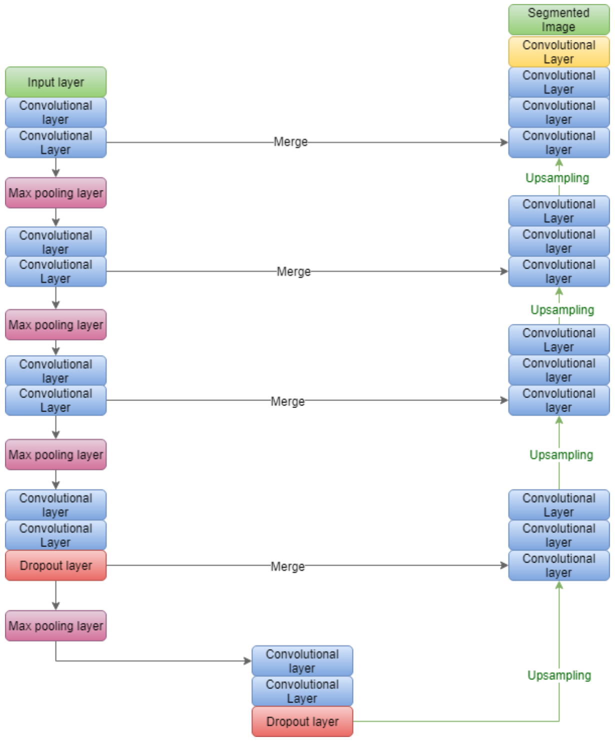


Figure 4. A fine-tuned U-Net architecture for performing semantic segmentation.

its maximum value thereby reducing the dimension of the data by a factor of 4.

Each step in the expansive part has two 3×3 convolutional layers followed by upsampling operations that double the output layer's image dimension by repeating each neuron's value twice. The skip connections that are used are operations that merge the output of last convolutional layer of each step at the downsampling part with the output of convolutional layer with the same resolutions at the upsampling part.

The U-Net architecture was fine-tuned, as shown in Figure 4, for the present study as mentioned below:

- An RGB input image of size 512×512 was used.
- No data augmentation was performed.
- Dropout layers were added to avoid over-fitting caused in small training datasets.
- Instead of a SGD optimizer an Adam optimizer was used.

5. Implementation

This section contains details about performing training on the model discussed in the previous section. The training was performed on a machine with an Operating System (OS) - Ubuntu 16.04 and a Graphical Processing Unit (GPU) - Nvidia GTX 1080Ti installed. Python code running on the Keras framework with a Tensorflow backend was used.

Keras: Keras is a high-level neural networks Application Programming Interface (API), written in Python enabling fast experimentation of various machine learning techniques, and runs on top of either TensorFlow, Theano or Microsoft Cognitive Toolkit (CNTK), which are software libraries for machine learning. Keras provides:

- Easy and fast prototyping through a user friendly interface, modularity and extensibility.
- Support for convolutional neural networks, recurrent networks and their combination.
- CPU and GPU compatibility.

Different approaches were tried to tune the learning rate hyperparameter. The first try was using the original U-Net model without dropout. The original U-Net model used SGD optimizer. The training evaluation results using this dataset can be seen in Figure 5. From the figure, it is seen that the training accuracy and the validation accuracy was low for semantic segmentation. Also, the training and validation loss was high.

Next, dropout layers were added added and evaluated. The training evaluation results can be seen in Figure 6. From the figure, it is seen that there was no change in the training accuracy and the validation accuracy. Also, the training and validation losses are high. Then the modifications mentioned earlier i.e. addition of two dropout layers and using an Adam optimizer were performed.

For the training of U-Net for segmentation the final tuning included an Adam optimizer with learning rate of $1e-4$. The learning rate was decided based on the training loss. The higher learning rates were causing the model to converge faster, and the model was also over-fitting. The training loss was decreasing, and accuracy was increasing whereas there was very slow change in validation loss and accuracy. The smaller learning rates caused the model to converge very slowly and the difference in the optimal training loss with the selected learning rate and the one with small learning rate was small.

Training was carried on until the stopping criteria was met. Initially the number of epochs was used as a stopping criteria but the model was not trained optimally i.e., training continued even after the accuracy started decreasing or loss started increasing. So, to avoid this situation the EarlyStopping class in Keras was used. With EarlyStopping, validation loss is monitored and if there is no change or increase in the validation loss for specified number of epochs then the training stops. Model checkpoint was added to save best weights. The model and weights were saved and used for testing the segmentation model. It took between 30 and 40 minutes to train the model for segmentation.

When training the model with training and validation datasets it is important to analyze the accuracy and loss of training versus validation datasets. Binary cross entropy was used as the loss function and the accuracy metric were used to evaluate the model during the training process. Figure 7 shows the accuracy and loss graph of the selected model for semantic segmentation. For a well trained model, not only should the accuracy be as high as possible and the loss

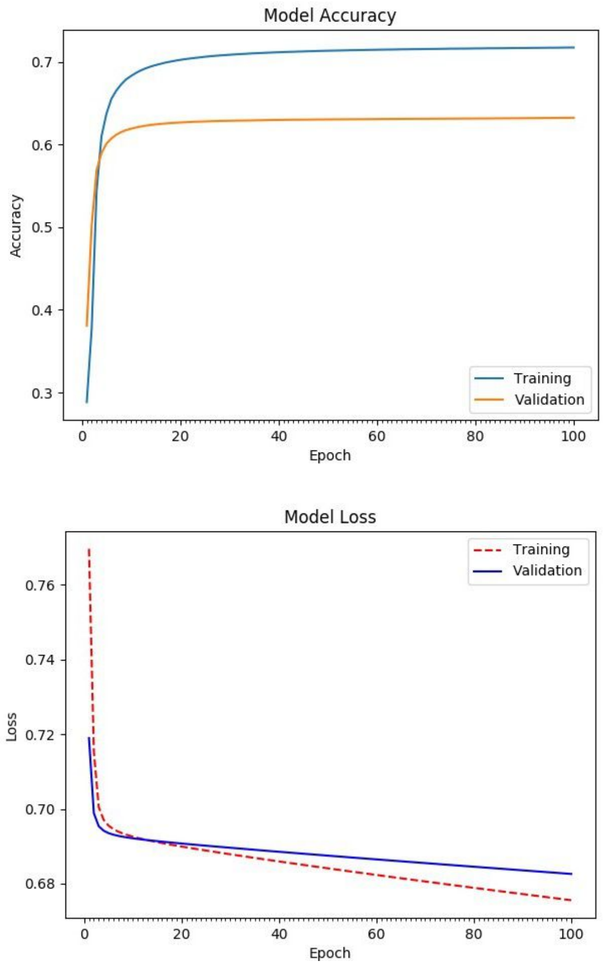


Figure 5. Results of training evaluation of U-Net model using SGD optimizer and without dropout layer.The top graph shows the accuracy vs epoch for training and validation datasets and the bottom graph shows the loss vs epoch for training and validation datasets.

be as low as possible, but also the difference between the training and validation accuracy and loss should be as small as possible. Figure 7 shows that the difference was low and the performance of this model was good.

6. Results

This section consists of the results obtained after testing the model on test datasets and the details about how the model was evaluated. A confusion matrix describes the performance of a classification model on the test datasets. It consists of four different combinations of predicted and actual values. Table 1 shows the four different combinations.

- True Positive (TP): When the predicted value is positive and the actual value is also positive.
- False Positive (FP): When the predicted value is positive but the actual value is negative.
- True Negative (TN): When both, the predicted value and the actual value are negative.

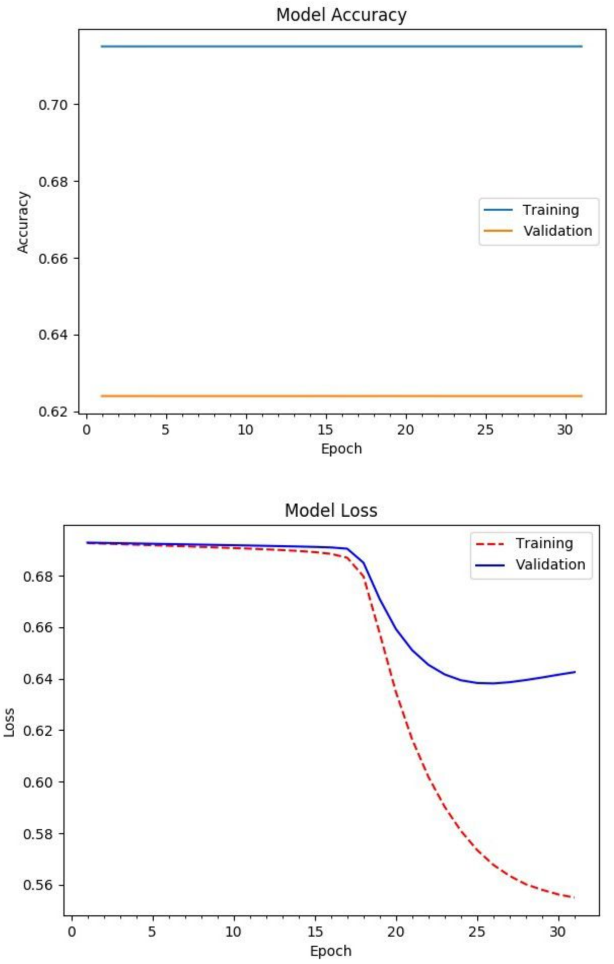


Figure 6. Results of training evaluation of U-Net model using SGD optimizer and with dropout layer. The top graph shows the accuracy vs epoch for training and validation datasets and the bottom graph shows the loss vs epoch for training and validation datasets.

- False Negative (FN): When the predicted value is negative but the actual value is positive.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

TABLE 1. CONFUSION MATRIX USED FOR EVALUATION.

This table helps to find accuracy, recall, precision, F-score, etc. In the present study, the Sorensen-Dice coefficient and accuracy were the metrics used to evaluate the model on test datasets. The Sorensen-Dice coefficient is a statistical metric used for comparing the similarity between two images. The *Sorensen-Dice coefficient* is given by:

$$\text{Sorensen-Dice coefficient} = \frac{2TP}{2TP + FP + FN} \tag{1}$$

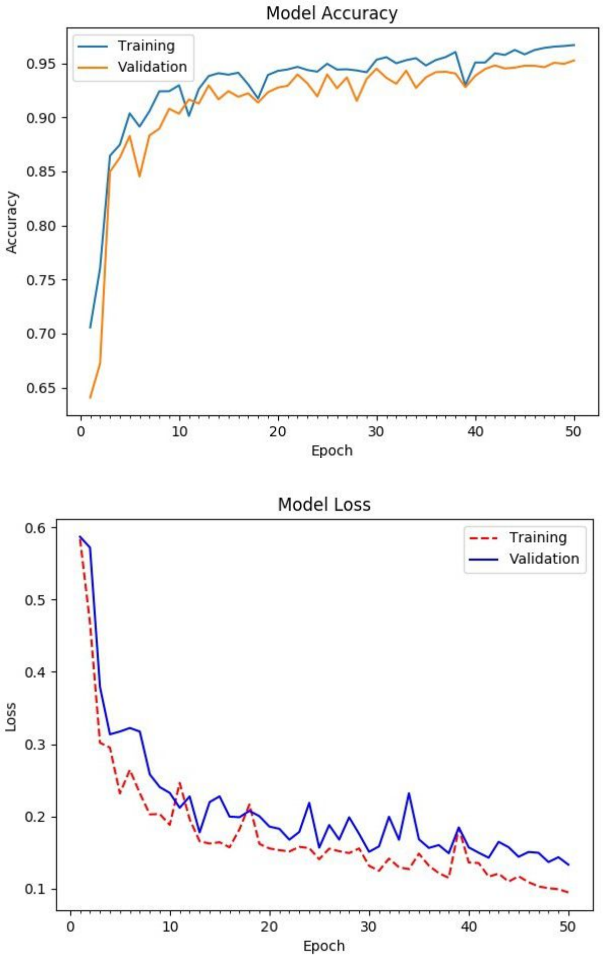


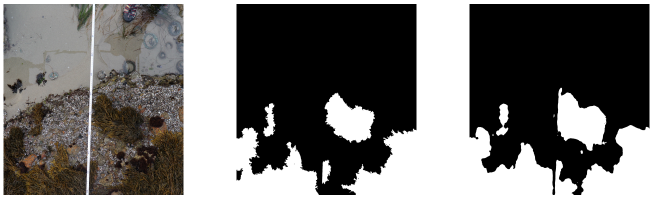
Figure 7. The accuracy and loss graph of the selected model. The top graph shows the accuracy vs epoch for training and validation datasets and the bottom graph shows the loss vs epoch for training and validation datasets.

and *accuracy* is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

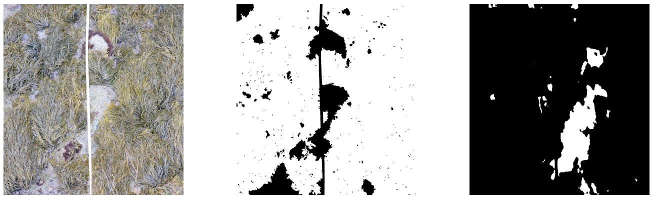
A test dataset of 50 images was used. These images were annotated manually, and the annotated images served as ground truth images for evaluation. The testing was performed using the model with training accuracy of 97.56% and validation accuracy of 95.24% and training loss of 0.098 and validation loss of 0.152.

Initially the test images were not pre-processed so the images that were bright or taken using flash were not segmented well. Figures 8 and 9 show the results of the model on unprocessed test images. Figure 8 shows that for the image taken under normal lights the model segments properly. Figure 9 shows that for the images that are bright, the model does not segment properly. So, two pre-processing techniques - histogram equalization and contrast limited adaptive histogram equalization (CLAHE) were tried.



(a) Original Image (b) Labeled Image (c) Predicted Mask

Figure 8. Result of unprocessed test image under normal lighting conditions.



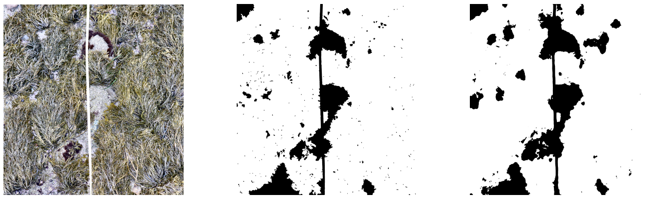
(a) Original Image (b) Labeled Image (c) Predicted Mask

Figure 9. Result of unprocessed test image under flash lighting conditions.

After trying histogram equalization, the results were good. The model was able to segment all the test images properly. Figure 10 shows the results of the model on test images after histogram equalization.

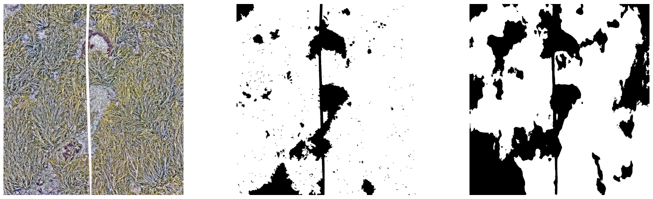
Adaptive histogram equalization (CLAHE) did a good job segmenting most of the test images however the results with histogram equalization were better. Figure 11 shows the results of the model on test images after CLAHE. On comparing the right image of Figure 10 and Figure 11 it is seen that the result after histogram equalization is more similar to the ground truth image(annotated image). There were some pixels that were incorrectly segmented as *Silvetia compressa*. But the overall results of the segmented images were good. So, histogram equalization was performed on all the test images before testing.

The segmented results of these pre-processed images were then evaluated by calculating the confusion matrix for each image. Using the confusion matrix, Sorensen-Dice coefficient was calculated for each image. Table 2 shows the confusion matrix created using the average count true



(a) Post Histogram Equalization Image (b) Labeled Image (c) Predicted Mask

Figure 10. Result of test images after histogram equalization.



(a) Post CLAHE Image

(b) Labeled Image

(c) Predicted Mask

Figure 11. Result of test images after adaptive histogram equalization.

positive, true negative, false positive and false negative pixel values of 50 test images. The resulting average value of Sorensen-Dice coefficients for the 50 test images is 0.9453. This value indicates that on an average the predicted mask and the annotated image are 94.53% similar. The accuracy for each image was also calculated, using the values from confusion matrix in Table 2. The average accuracy is 94.51% which is close to the validation accuracy.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	124229	8828
	Negative	5554	123533

TABLE 2. Confusion matrix for the segmentation model.

7. Conclusions and Future Work

The major motivation of this work was to apply semantic segmentation to identify a specific algae species - *Silvetia compressa*. It was found that fine-tuning an existing model achieved a good accuracy for this application. The main goal was to reduce the time required to analyze the images for ecological monitoring and to make it less prone to human error. This was achieved by creating an automated tool to find the assemblage of *Silvetia compressa* by implementing semantic segmentation.

The semantic segmentation model was able to provide good results, but there were a few images where a few pixels misclassified. Improvements in training and preprocessing can be done to achieve more accurate results. There are other species that are present in the two rocky intertidal zones of Santa Rosa Island. The created models should be tested on the new species and a multi-class classification and segmentation model should be implemented to identify these species.

References

[1] N. G. Yoccoz, *Ecological Monitoring*. Encyclopedia of Life Sciences, 2012, ISSN 1476-9506.s, doi:10.1002/9780470015902.a0023571.

[2] S. Airamé, J. E. Dugan, K. D. Lafferty, H. Leslie, D. A. McArdle, and R. R. Warner, "Applying ecological criteria to marine reserve design: a case study from the california channel islands," *Ecological applications*, vol. 13, no. sp1, pp. 170–184, 2003.

[3] J. Shotton, M. Johnson, and R. Cipolla, "Semantic Texton Forests for Image Categorization and Segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008*. IEEE, 2008, pp. 1–8.

[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011*. Ieee, 2011, pp. 1297–1304.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for Biomedical Image Segmentation," in *International Conference on Medical image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How Transferable are Features in Deep Neural Networks?" in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3320–3328. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969197

[13] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught Learning: Transfer Learning from Unlabeled Data," in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 759–766.

[14] A. Quattoni, M. Collins, and T. Darrell, "Transfer Learning for Image Classification with Sparse Prototype Representations," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on 2008*. IEEE, 2008, pp. 1–8.

[15] C. Douarre, R. Schielein, C. Frindel, S. Gerth, and D. Rousseau, "Transfer Learning from Synthetic Data Applied to Soil-Root Segmentation in X-Ray Tomography Images," *Journal of Imaging*, vol. 4, no. 5, p. 65, 2018.