MarshCover: A Web-based Tool for Estimating Vegetation Coverage in Marsh Images Using Convolutional Neural Networks

Lucas Welch
University of North Florida
1 UNF Drive, Jacksonville, FL 32224
n00797216@ospreys.unf.edu

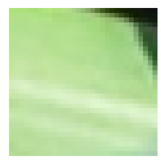
Xudong Liu
University of North Florida
1 UNF Drive, Jacksonville, FL 32224
xudong.liu@unf.edu

Abstract

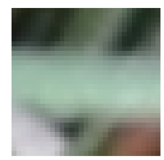
Marsh ecosystems are some of our most important, serving many crucial ecological functions. They are also rapidly changing, and it is vital for scientists to track these changes. This includes monitoring the health of marshes via estimating ground coverage by various grass species, a task that requires human labor to look at marsh images and manually estimate the coverage. Clearly, this task can be quite formidable. To automate this standard yet laborsome process, we develop a web-based system, called MarshCover, that automates the process of estimating vegetation density in marsh images using convolutional neural networks (CNNs). MarshCover, to the best of our knowledge, is the first such tool available to biologists that uses CNNs for marsh vegetation estimations. In order to select effective CNN models for our MarshCover server, we conduct extensive empirical analyses of three distinct CNNs, i.e., LeNet-5, AlexNet and VGG-16, to compare their performances on a public marsh image dataset. To this end, we address two classification problems for this paper: a *binary classification* problem classifying points as vegetated and unvegetated, and a *multiclass classification* problem that classifies points into either an unvegetated class or one of five different species classes. Our experiments identify the VGG-16 model as the best classifier to embed in MarshCover for both the binary classification problem and the full classification problem with a two model classifier (called two-shot). These two classifiers had accuracies on test data of 90.76% and 84% respectively. MarshCover is publicly available online.

Introduction

Maintaining healthy marsh ecosystems on the coast is vital because it serves many crucial ecological functions, such as offering habitats for animals and maintaining proper oxygen levels in the water. As marshes are wetlands composed of many species of grasses, their health can be assessed by monitoring these communities. Biologists often do this by manually estimating the percent coverage of various grass species in images taken over the marsh. However, this process can be labor-intensive. As a result, there have been collaborations between biologists and computer scientists creating computational means to automate the process of measuring vegetation coverage in marsh images. Recent work by Welch et al. (Welch et al. 2021; Welch and Liu 2021) presented a collected dataset of image snippets labeled by six classes: Unvegetated, Spartina alterfloris, Batis maritima, Juncus roemerianus, Avicennia germinans, and Sarcocornia perennis. Examples of snippets of each of these classes is shown in 1. This dataset was based on images provided by the Guana Tolomato Matanzas National Estuarine Research Reserve (GTMNERR) [1]. It also shows empirical results comparing two different binary classifiers, which were trained LeNet-5 (LeCun et al. 1988) and AlexNet (Krizhevsky, Sutskever, and Hinton 2012a) convolutional neural network (CNN) models.

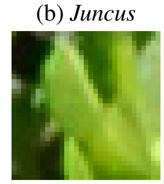


(a) *Spartina*

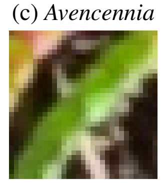


(b) *Juncus*

(c) *Avencennia*



(d) *Batis*



(e) *Sarcocornia*

(f) Unvegetated

Figure 1: Sample snippet images in our marsh dataset. Note the differences between the vegetated classes and the unvegetated class, which lacks obvious green regions.

The work by Welch et al. is only concerned with the binary classification problem— that is, deciding whether a marsh snippet is vegetated or not, and it does not present results on the full multi-class classification involving all six classes. In this paper, we demonstrate our findings on training CNN models to solve the full classification problem, us-

[1] `https://gtmnerr.org/`

ing the dataset by Welch et al. [2] Also, to the best of our knowledge, there is not a publicly available tool for the biologists to use for the purpose of measuring marsh vegetation coverage. As such, our work in this paper was created to fill this gap developing a webapp, which we call MarshCover, available for anyone to use to upload marsh grass images to obtain estimates of the species-specific vegetation density in the image. These are the two major contributions in our paper.

The rest of the paper is arranged as follows. We begin by discussing the three neural network models used in this paper. We then report our experimental results on training and testing multi-class classification models, including LeNet-5, AlexNet and VGGNet (Simonyan and Zisserman 2014) models, and our results suggest VGGNet as the best model to be embedded in MarshCover. Then in the following section we present our MarshCover including its front-end and back-end and demonstrate its effectiveness through a use case. Finally, we close with conclusion and future work.

Models Used

LeNet-5

LeNet-5 is a simple CNN that was designed and implemented for optical character recognition on the MNIST dataset of images of 32x32x1(LeCun et al. 1998). Because our images are of a different size, we adopt a slight variation of the original LeNet-5 model. There are two differences between our model and the original model: The first being the size of the first convolutional layer's receptive field and the other being the number of output classes (we use either 2, 5, or 6 depending on the classification problem while the original model used 10 output classes).

AlexNet

Similar to LeNet-5, AlexNet also is a convolutional neural network with numerous convolution and pooling layers(Krizhevsky, Sutskever, and Hinton 2012b). Because of the deeper nature of AlexNet, it is a more complex model than LeNet5. However, the size of the input image originally used for AlexNet was 244x244 with three color channels (Krizhevsky, Sutskever, and Hinton 2012b). Because our input image is only 33x33, many parameters like stride and receptive field size had to be adjusted. The hidden layers, much like LeNet-5, consist of an alternating pattern of convolutions and pooling layers. However, for AlexNet, max pooling is used instead of average pooling (Krizhevsky, Sutskever, and Hinton 2012b).

VGG-16

VGG-16—so named for its 16 convolutional layers—continues our trend of deeper and more complex networks. VGG-16 was designed as a more generalized model. Made originally for the Google ImageNet challenge, it has been widely used in various applications (Simonyan and Zisserman 2014). This is attributable to its depth and relative simplicity as well as its effectiveness. As such, it is often used to quickly train a familiar model on new image datasets for image identification problems (Simonyan and Zisserman 2014). While its architecture is much deeper than the previously discussed models, it is very similar in that it is composed of a series of alternating convolutional and max pooling layers. There is one such pooling layer every three convolutions, and the model concludes with a series of three fully connected layers. The model is shown for its original input size below in Figure 2. Because of its deeper nature, it is hoped that the VGG-16 model will be able to identify more complex features in the images, helping in identification.

[2] `http://unfail.ccec.unf.edu/marshdata.html`

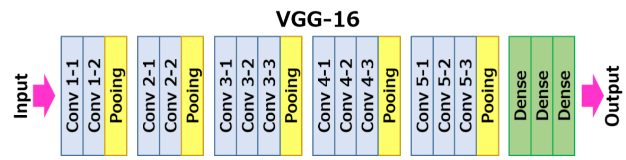


Figure 2: Diagram of original VGG-16 network. Note the 16 convolutional layers.

Model Selection

As noted by Welch et al. (Welch et al. 2021; Welch and Liu 2021), biologists have developed a systemic approach to vegetation density calculation by randomly picking a number of pixels in a marsh ground cover image (an example is shown in Figure 3), manually labeling sorting pixels into the aforementioned classes, and calculating the estimated vegetation coverage accordingly. Inspired by their process, we want to develop a web-based system that automates the estimation of vegetation coverage in marsh images. This provides an easy-to-use tool for biologists to get the general percent coverage of vegetation as a whole or as a more detailed percent coverage of each individual grass species in a marsh image. At the same time, accuracy should be maximized for these predictions by the neural network models. In order to achieve this, we compare and choose the best among three CNN architectures for the two estimators at hand. One estimator, which we refer to as the *binary estimator*, is to take as input a 33x33 snippet surrounding the chosen pixel and to predict whether the snippet is vegetated or not—whether or not the center pixel contains vegetation. The other estimator, which we refer to as the *full estimator*, is to take a snippet to predict which one of the six classes it belongs to.

The training for the binary estimator is straightforward. For the full estimator, because we note a high level of imbalance (over 65% of the snippets are bare), we consider the following two implementations, namely, one-shot and two-shot classifiers. The *one-shot* classifier, as the name suggests, runs a multi-class classifier to predict one out of the six classes. On the other hand, the *two-shot* classifier first runs the binary classifier. Then, if the binary classifier classifies the snippet as vegetated, it runs a separate species classifier to predict which of five species present in the marsh the snippet belongs to. Clearly, these two estimators we use in our system call for experimenting with the three CNNs

Figure 3: An example marsh ground image by GTMNERR

across three classification problems, i.e., binary, one-shot, and species classifiers. Afterward, the two-shot classifier simply is a combination of the binary and species classifiers.

Convolutional neural networks, since their inception in the 1980s with the design of LeNet, have been seen extensively solving image classification problems. Using various modules such as the convolutional layers and pooling layers, a slew of different CNN architectures have been proposed. Due to the complexity of the images, we decided to use the already-available dataset as well as the snippet strategy used in Welch et al. as opposed to semantic image segmentation. In order to choose which models will be embedded into our system, MarshCover, we choose to experiment with three CNN architectures: LeNet-5, AlexNet, and VGGNet, which have been proven to be effective in solving problems similar to ours. To this end, we carry out cross validation experiments with these CNNs for our Binary, Species, and One-Shot classifiers.

10-fold Cross-Validations

In order to find the best model or models to use in MarshCover, we executed a total of nine 10-fold cross validations (CVs). We executed 10-fold CVs for each combination of topology (LeNet-5, AlexNet, and VGG) and number of output classes (binary, species, and one-shot). In order to do this, the dataset was split in a stratified manner on all six classes into a training and test set. The training and test set are allotted 80% and 20% of the dataset respectively. After obtaining the training set, we then split it into 10 buckets, each containing 10% of the training set. In order to perform the 10-fold CV from here, the 10 buckets from the training set are iterated through, each one being used as a validation set for the model being trained, the other nine buckets being combined into a training set. After training a fresh model for each bucket, the results are compared, and the model with the best performance on its validation set is chosen for each of the nine cross-validations. A summary of the results for this are contained in Table 1. Because of the large volume of data, these results are a summary, including only the best results from each cross-validation. The models are trained for 120 epochs (iterations of training through the training set) and a batch size of 1 using stochastic gradient descent and a learning rate of 10^{-4}.

According to these results, the VGG-16 models have the best test accuracies surpassing LeNet-5 and AlexNet for Binary and Species classifiers. And for the One-Shot classifier, LeNet-5 is best.

While the 10-fold CVs inform of the best models to use of each type, it is still uncertain whether a single model (One-Shot) or a pipeline of two models (Binary followed by Species) is best. The following experiment, however, addresses this issue.

One-Shot vs. Two-Shot

Now that the best-performing models have been identified in cross validation, it is important to know which of the two potential estimators is most accurate. In order to do this, the accuracies of the one- and two-shot estimators must be compared. While obtaining traditional accuracy (also known as top-1 accuracy) where the model's most confident prediction is compared to the labeled ground-truth, top-n wanted to be checked (accuracies from top-1 to top-5 in this case). To compute top-n accuracies, we need to get the output probability distributions over the six classes. This is straightforward for the One-Shot classifier. However, for Two-Shot, we have to define it as follows.

We now use b to denote unvegetated, s Spartina, j Juncus, m Batis, p Sarcocornia, and a Avicennia. For a Two-Shot classifier model M, let i be a snippet fed to M, $\langle u, v\rangle$ the binary output vector produced by the binary classifier, $\langle s, j, m, p, a\rangle$ the multi-class output vector by the species classifier. Clearly, the probability of i being unvegetated by model M is $P_i^M(Unv) = \frac{u}{u+v}$, and the probability of s being vegetated by model M is $P_i^M(Veg) = \frac{v}{u+v}$. Now, we define the probabilities of s being one of the five species, Spartina, Juncus, Batis, Sarcocornia, and Avicennia, to be $P_i^M(s) = P_i^M(Veg) * \frac{s}{s+j+m+p+a}$, $P_i^M(j) = P_i^M(Veg) * \frac{j}{s+j+m+p+a}$, $P_i^M(m) = P_i^M(Veg) * \frac{b}{s+j+m+p+a}$, $P_i^M(p) = P_i^M(Veg) * \frac{c}{s+j+m+p+a}$, and $P_i^M(a) = P_i^M(Veg) * \frac{a}{s+j+m+p+a}$, respectively.

Therefore, for any snippet i sent to a Two-Shot classifier model M, we obtain a probability distribution over the six labels of Unvegetated, Spartina, Juncus, Batis, Sarcocornia, and Avicennia; such distribution, denoted by $P_M(i)$, is given by the vector $\langle P_i^M(Unv)$,$P_i^M(s)$, $P_i^M(j)$, $P_i^M(m)$, $P_i^M(p)$, and $P_i^M(a)\rangle$.

Now that $P_M(i)$ has been defined, the top-n accuracy is straightforward as follows. Given a test set, T, of n snippets $\{i_1, ..., i_n\}$, each i_j labeled by a ground-truth g_j, and a model (one- or two-shot classifier), M. We first obtain, for snippet i_j, a vector $\{l_{j1}, ..., l_{j6}\}$ of the six labels ranked by their corresponding probabilities obtained in P. We can then define the top-n accuracy, $TopAccuracy(M, T, n)$, where

Table 1: Summary table of the nine 10-fold cross-validations. Below are the training, validation, and test results for the best model from each 10-fold CV.

	LeNet-5			AlexNet			VGG-16		
	Training	Validation	Test	Training	Validation	Test	Training	Validation	Test
Binary	98.18	90.05	89.3	98.94	90.37	90.52	99.21	91.14	**90.76**
Species	76.98	68.85	67.31	96.04	72.84	71.87	89.09	75.81	**75.37**
One-Shot	84.5	82.58	**82.35**	96.25	82.9	82.31	97.87	82.05	81.32

$1 \leq n \leq 6$, as:

$$\frac{|\{i_j \in T : g_j \in \{l_{j1}, ..., l_{jn}\}\}|}{|T|} \tag{1}$$

The top-n results are listed in Table 2. As can be seen, VGG has the best top-1 test accuracy when comparing Two-Shot models. On the other hand, AlexNet has a better top-1 test accuracy when comparing the One-Shot models. These patterns continue beyond top-1 accuracy to encompass all top-n values (1-5). Additionally, VGG's Two-Shot estimator has the best top-n accuracy when comparing all six pipelines. Therefore, the **VGG-16 Two-Shot** classifier will be used as the full estimator in the MarshCover program, and the **VGG-16 Binary** classifier will be used for the binary estimator.

MarshCover

In this section, we discuss MarshCover, the web-based system that we implemented to measure vegetation coverage in marsh grass photographs. In Figure 4 we show an overview of MarshCover that includes front-end and back-end and how data flow between the two. A prototype system also has been developed and deployed, and is available for public access at `http://139.62.210.148/MarshCover`.

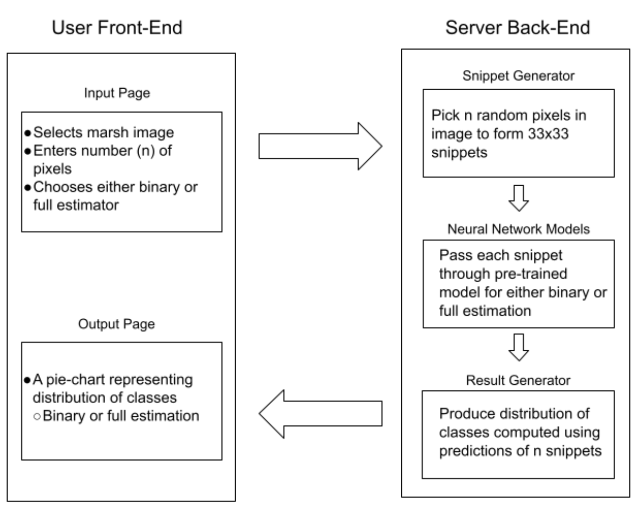


Figure 4: Diagram depicting the interaction between the front-end and back-end

The front-end for MarshCover is fairly straight-forward. When a user accesses the site, they are prompted to select a marsh ground image, enter the number n of points for which snippets are to be generated, and choose either binary or full estimator to perform vegetation density analysis. Once this input form is submitted, the user will be directed to the result page presenting a pie chart denoting binary or full distribution of the classes in the image, as well as a table tallying each snippet, its point coordinate in the image, and its predicted label.

To provide a user-friendly webapp that integrates Python programming, we develop MarshCover on the Django (Django Software Foundation) framework. Below we demonstrate MarshCover via a user case.

When accessed, MarshCover shows an interface as shown in Figure 5, where the user is prompted on the purpose of the webapp as well as instructions on how to use it.

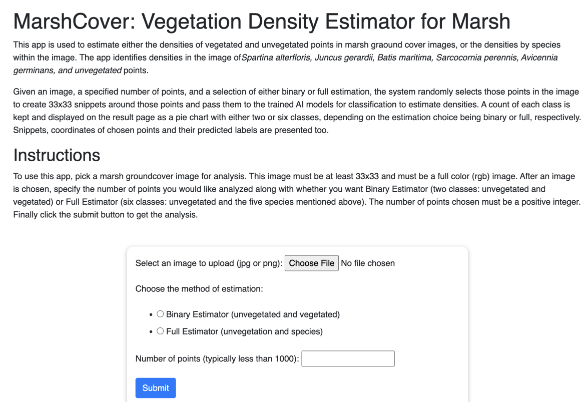


Figure 5: MarshCover main page

The user can upload any marsh image for estimation. Suppose she uploads the example image in Figure 3. After Binary or Full Estimator is selected and 500 is entered as the number of points, or snippets, to use for the analysis, the webapp sends all the information to the back-end server to run the pre-trained VGG-16 models to gather results. Specifically, the server generates n 33x33 snippets around the randomly chosen n pixels. Then, it passes every snippet through the pre-trained model for either Binary or Full classification. As aforementioned, the Binary Estimator uses the VGG-16 Binary classifier, and the Full Estimator the VGG-16 Two-Shot classifier. Finally, the server produces the resulting distribution of classes using the predictions of the n snippets.

We include the output page in Figure 6 for running binary estimation of the sample image in Figure 3. The cor-

Table 2: Top-n results for each of the best pipelines. One-shot consists of a single model with 6 output classes while the two-shot pipelines consist of a binary classifier and species classifier Two-Shot together.

	LeNet-5		AlexNet		VGG-16	
n	One-Shot	Two-Shot	One-Shot	Two-Shot	One-Shot	Two-Shot
1	82.35	80.49	82.31	82.86	81.32	**84**
2	92.52	90.94	91.65	92.87	90.55	**93.48**
3	**96.66**	95.25	95.3	96.16	94.72	96.45
4	**98.69**	97.22	97.4	97.87	97.33	97.8
5	**99.57**	98.18	98.6	99.05	98.67	98.76

responding output page if the user selected full estimation is shown in Figure 7. We note that our webapp not only provides the estimated vegetation density distribution over the classes, but also a point-by-point breakdown of the results, including images of the selected snippets, coordinates of their central pixels, and their predicted labels.

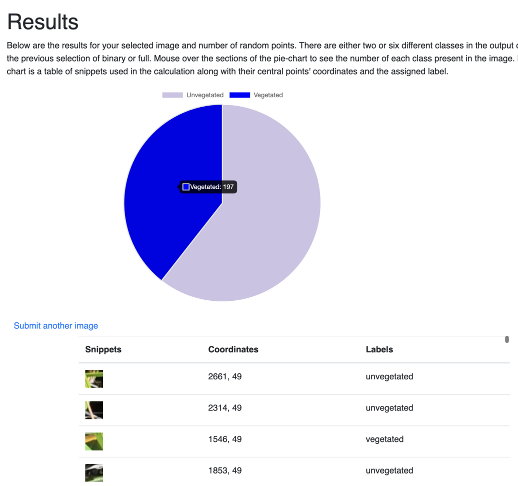


Figure 6: Binary estimation output page for input image in Figure 3

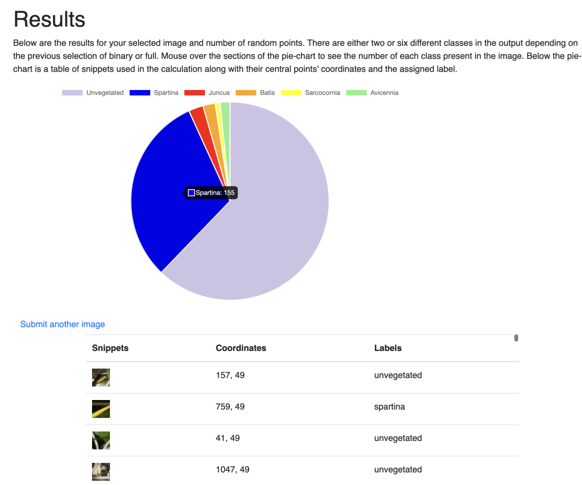


Figure 7: Full estimation output page for input image in Figure 3

Conclusion and Future Work

In this work, we studied the problem of estimating vegetation coverage in marsh ground cover images using convolutional neural networks. Our contributions are of two folds. First, we provided an extensive empirical analysis of three CNNs, i.e., LeNet-5, AlexNet and VGGNet, to compare their performances to solve both the binary and the multi-class classification problems on a marsh dataset. During our experiments, it was found that the Two-Shot pipeline model using VGG-16 is the most accurate with a test accuracy of 84%, and that the Binary VGG-16 model outperforms others with a test accuracy of 90.76%, on par with the preliminary results reported by Welch et al. (Welch et al. 2021). Second, we developed and deployed a prototype web-based tool, MarshCover, for the biology community to use to automatically to obtain vegetation coverage. This, to the best of our knowledge, is the first such tool available to biologists that uses CNNs for estimations.

For the future work, there are many new types of CNN models to try such as ResNet (He et al. 2016), Vision Transformer (Dosovitskiy et al. 2020) and DenseNet (Huang et al. 2017). Moreover, models of these types pre-trained using ImageNet are available and we plan to apply transfer learning (Bozinovski and Fulgosi 1976) to further train these models on the marsh dataset. With these more versatile models, accuracy could be improved beyond our current state-of-art. We also plan to conduct user case studies with biologists for usability of our web-based tool. Feedback from biologist users will be key to improve our developed prototype. For instance, a batch uploader could be very helpful for scientists who have a myriad of images to analyze. Additionally, we would like to implement a file output that is able to record each point and its label, so the biology researchers are able to find the points that were labeled by the program, allowing for tighter quality control.

References

[Bozinovski and Fulgosi, 1976] Bozinovski, S., and Fulgosi, A. 1976. The influence of pattern similarity and transfer learning upon the training of a base perceptron b2. In *Proceedings of Symposium Informatica*.

[Django Software Foundation,] Django Software Foundation. Django.

[Dosovitskiy et al., 2020] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[He et al., 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*, 770–778.

[Huang et al., 2017] 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*, 4700–4708.

[Krizhevsky, Sutskever, and Hinton, 2012a] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012a. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

[Krizhevsky, Sutskever, and Hinton, 2012b] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012b. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

[LeCun et al., 1988] LeCun, Y.; Touresky, D.; Hinton, G.; and Sejnowski, T. 1988. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann.

[LeCun et al., 1998] LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

[Simonyan and Zisserman, 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Welch and Liu, 2021] Welch, L., and Liu, X. 2021. Measuring vegetation density in marsh grass photographs using deep neural networks (student abstract). In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press.

[Welch et al., 2021] Welch, L.; Liu, X.; Reddivari, S.; Umapathy, K.; and Kahanda, I. 2021. Vegetation coverage in marsh grass photography using convolutional neural networks. In *Proceedings of the 34th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*. University of Florida Press.