In the Weeds: Automating Eelgrass Transect Insights through Computer Vision

Jannik Elsäßer
IT University of Copenhagen
Copenhagen, Denmark
jels@itu.dk

***Abstract*—This research employs computer vision to automate visual eelgrass (Zostera Marina) coverage estimation, aiming to enhance survey techniques. Using eelgrass transect data, the study introduces a two-step method: classifying valid/invalid eelgrass transect images and regressing eelgrass coverage on valid ones. Data preparation involves using Optical Character Recognition (OCR) to extract video over lay information, and then cross-referencing annotation data to create train and test datasets. Nine videos from six locations, with varying lengths and seasons, form the dataset.**

Machine learning experiments involve Convolutional Neural Networks (CNNs), Residual Neural Networks (ResNets), and Vision Transformers (ViTs). ViTs, despite initial challenges, outperform others, achieving a 63.3% accuracy in binary classification. Principal Component Analysis (PCA) reveals insights into model performance.

A unique aspect is the Label Studio setup for dataset annotation, overcoming challenges of human subjectivity. This platform, deployed on Azure, ensures a streamlined annotation process. Despite challenges, the study underscores machine learning's potential for efficient eelgrass analysis, contributing foundational resources for future endeavors in marine resource conservation.

***Index Terms*—eelgrass, computer vision, machine learning, environmental impact assessment, data science, Vision Transformer, CNN, ResNet, Label Studio, Azure, OCR, marine resource conservation, dataset annotation, principal component analysis (PCA), survey techniques, underwater video transects, marine ecology, convolutional neural networks, binary classification, regression, automation, image processing**

I. Introduction

The primary goal of this research project is to set a baseline method for estimating visual eelgrass (Zostera Marina) coverage using computer vision methods. The project aims to study and compare various methods for visually classifying eelgrass coverage using machine learning. By conducting this study, the project intends to advance the survey and environmental impact assessment techniques employed at the company and gain valuable insights into the performance of different model architectures and feature extraction methods. Ultimately, this knowledge will contribute to the development of a specialized method for automating the analysis of underwater video transects, with computer vision, at the company.

On a broader note, in the context of nature conservation, the automation of eelgrass coverage estimation holds great significance. [1] highlighted the value and vulnerability of underwater eelgrass meadows, which serve as critical coastal habitats. Eelgrass meadows help in carbon sequestration, provide protection and nurseries for juvenile fishes, and contribute to sediment stabilization [2]. Eelgrass has also been considered as one of the most important ecological indicators [3]. Therefore, by investigating and proposing methodologies for automating the estimation of eelgrass coverage, this study contributes to the conservation and management of this valuable marine resource.

This project is the author's 7.5 ECTS elective, *"Collaborating with Companies"* project in his 5th Semester BSc. in Data Science at the IT University of Copenhagen, Denmark. Associate professor Veronika Cheplygina supervised the project in the DASYA Group (Data Intensive Systems and Applications).

Repo available at https://github.com/jannik-el/InTheWeeds

II. Related Work

It is important to note that seagrass refers to various species of submerged flowering plants belonging to the order Alismatales, specifically those within the genera Zostera, with Zostera Marina or eelgrass being the specific seagrass analyzed in this study. The paper by [4] specifically trained on Zostera Marina (eelgrass), while [5] does not directly refer to the species of seagrass used. However, in the context of using computer vision, seagrass's characteristic leaf morphology, density variations contributing to distinctive underwater meadow structures, and consistent color spectrums suggest that methods applied to eelgrass can work on other seagrass species, and vice versa (see Figure 17).

There are currently only two papers directly related to this study: [5] with the title "*Looking for Seagrass: Deep Learning for Visual Coverage Estimation*", utilizing a dataset of around 6000 images, and [4] titled "*SeaGrassDetect: A Novel Method for the Detection of Seagrass from Unlabelled Underwater Videos*" whose dataset has not been made public.

In the approach taken by [5], the data was annotated with polygons, signifying the presence of eelgrass. During training superpixels were generated using SLIC (Simple Linear Iterative Clustering) or CW (Compact Watershed). Equal-sized superpixels were calculated, and features were extracted using CNNs (Convolutional Neural Networks), HOG (Histogram of Gradients), or LBP (Local Binary Patterns). These features were then input into a classifier that predicted eelgrass coverage values. Consequently color was used as an input for the model. However, this approach is considered flawed because color introduces bias relative to water depth, which

changes significantly with water depth due to the absorption and scattering of light rays. ML models with 3-channel inputs should not be used for predicting eelgrass coverage unless a form of color normalization is implemented, as highlighted by [6]. Color can still be used when predicting if the image is a valid eelgrass transect image however, since the task there is not to predict eelgrass but to predict if the image is a valid eelgrass transect image.

In "*SeaGrassDetect*", [4] argues that the method used in "*Looking for Seagrass*" is not an accurate representation of eelgrass. He points out that the ground truth labels in "*Looking for Seagrass*" were "roughly" annotated polygons, which labeled the separation between seabed and vegetation more than the actual seagrass coverage. Therefore, he suggests that his method of using line detection to extract the number of lines, in relation to the total length of these lines, better represents the "fine leaves of the plants" (stalks) and therefore the eelgrass coverage. This eelgrass stalk length and count is fed into a GMM and used to predict seagrass coverage.

However, it is also true to contend that his method does not more accurately represent eelgrass, as LSD not only detects eelgrass stalks but also the side of a diver's glove, a camera cable, the side of a boat, etc.

While there was an intention to implement both [4] and [5] methods, due to time and technical constraints, this proved not possible during the project. Therefore, this paper is not able to conclude on any of these issues.

III. Background

The core concept of this paper is to implement a computer vision model to 'watch' the videos and output eelgrass coverage. However, this computer vision model step raises multiple problems. For instance, the video camera may not always be facing the seabed, or the survey vessel may be off course, rendering the filmed footage worthless. Therefore, this paper proposes a two-step method to circumvent these issues.

1) Classification of valid/invalid eelgrass transect images
2) Regression of eelgrass coverage on only valid eelgrass transect images

It's important to note from the outset that there are two types of models and machine learning tasks, a detail that will be consistently referenced throughout the project.

The data in this project is provided by the company.

During an eelgrass video transect, an underwater camera is towed behind a survey vessel on a so-called "sled." This sled ensures that the camera is positioned at an angle toward the seabed and that the camera does not swing while being dragged through the current. The depth at which the sled is towed is monitored by a field technician or marine biologist on board the survey vessel. They are tasked with maintaining the sled at a certain height above the seabed, lifting the sled if there are any obstacles in its way, or lifting the sled out of the water and cleaning it if the camera gets covered.

Following the field data collection, a marine biologist would review the camera video footage in conjunction with an Excel spreadsheet containing timestamps and the survey vessel's location, making annotations specifically when notable changes for longer periods in eelgrass coverage were observed.

This methodology for eelgrass transect surveys is a predefined standard set up by Aarhus University [7].

In total, nine different videos were used from six different locations, varying between 45 minutes to 2 hours in length, from two different seasons.

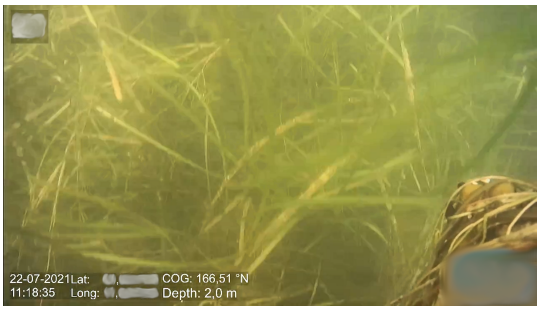

Fig. 1. Screenshot of an eelgrass transect video

IV. Data

A. Data Preparation

To facilitate machine learning applications, the initial step involved preparing the data into a usable format. The ultimate goal in data cleaning was to obtain images with associated eelgrass coverage labels.

To achieve this, an Optical Character Recognition (OCR) model, specifically Google's Tesseract, was used to extract overlay information from the video frames, which was then cross-referenced with the rows in the Excel spreadsheet.

Using this method enabled the extraction of an average of 84.4% of all potential data points from each video. Instances where data points were not extracted can be attributed to errors in the OCR extraction process. This was also partly intentional, as the extraction was programmed to validate the image extraction only when the date, time, latitude, and longitude all matched the Excel row. Therefore if any character in one of 4 columns was off it would not be matched to an image. This 4 class cross-referencing was especially important since the survey vessels were sailing in a course directly north/south, therefore meaning that latitude was only slightly changing, and in some cases due to a bug the timestamp would freeze and stay the same for several seconds.

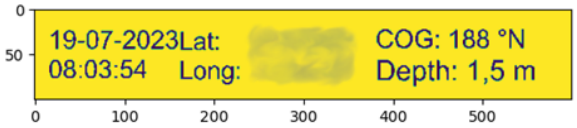

Fig. 2. Post-processing of eelgrass transect image for OCR, GPS position has been censored for anonymization purposes

To enhance reliability, given the inconsistent structure of Excel tables, the Excel data was also transitioned to an Azure SQL database, streamlining the process of value comparisons.

B. Train and Test dataset

The data used in this project is divided into two distinctly disconnected groups: the train and the test dataset.

The train dataset comprises all the images and labels extracted from the videos from one season. Within this dataset, there are six different transect locations, nine different videos, totaling 3445 images. These images were annotated by two different marine biologists, all from one season. Annotations were made while the video was running, sometimes at a faster speed than it was recorded. Therefore, they can be described as "roughly" annotated in the context of machine learning, since the annotation is based on a moving image, not the individual frame at which it was annotated. This means the data in this dataset is not optimally annotated for a machine learning task and is inaccurate based on a frame-by-frame analysis. The delay from impression to annotation was minimized using a mechanical slider fabricated by the company, which immediately inputs the eelgrass coverage into the Excel spreadsheets. The delay is also not necessarily something that needs improvement, as per the current setup standardized by Aarhus University [7], since a frame-by-frame annotation is not the desired outcome. The mechanical slider purely optimizes the annotation process, and as long as the delay is below 30 seconds, it was determined that it would not impact the model used to calculate the environmental impact, since the boat's maximum distance traveled in up to 30 seconds would be within the model's resolution.

The test dataset, on the other hand, is a specially created dataset for the validation of any model created using the train dataset. Since frame-to-label accuracy cannot be guaranteed with the train dataset, the test dataset was created using a custom annotation setup. This was done using Label Studio Community, which was then deployed as an App Service to Azure.

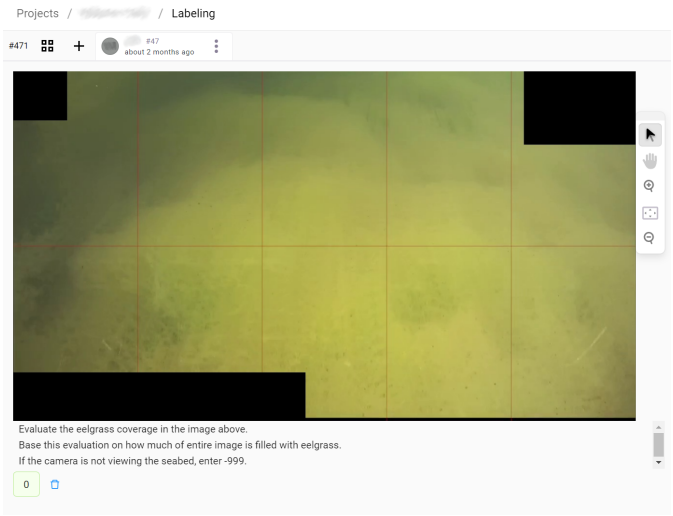

Fig. 3. Screenshot of Label Studio Custom Annotation Setup

Since Label Studio Community does not have persistent storage, a custom setup was created that used Azure blob storage containers for the input images, results, and the Label Studio database.

In total, the test dataset only consisted of 298 images, 152 of which were annotated by all annotators to complete an inter-annotator agreement. These images were extracted from the video using the same OCR cross referencing method as with the train dataset, since the videos had also been annotated in real time while on board the vessel.

C. Exploratory Data Analysis

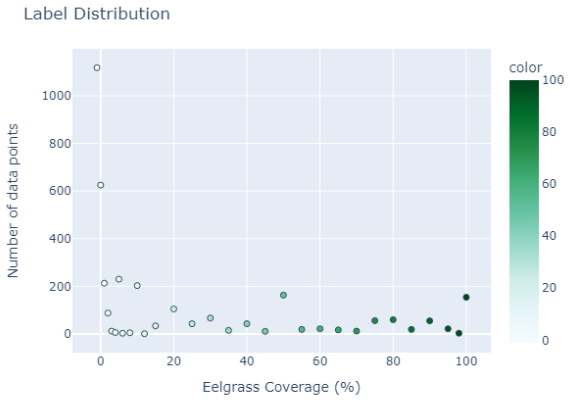

Fig. 4. Eelgrass Coverage Label Distribution (-1 is invalid image)

1) Train Dataset: As made clear in Figure 4, there was a significant imbalance within the eelgrass coverage classes. While eelgrass coverage is a continuous value, both annotators had a tendency to annotate discretely, which was expected, since their annotations should be an immediate estimation of the local coverage. While this imbalance seems incredibly significant, it is worth adding that in the context of the binary classification of predicting valid and invalid eelgrass transect images, this imbalance was much less significant, as seen in Figure 5.

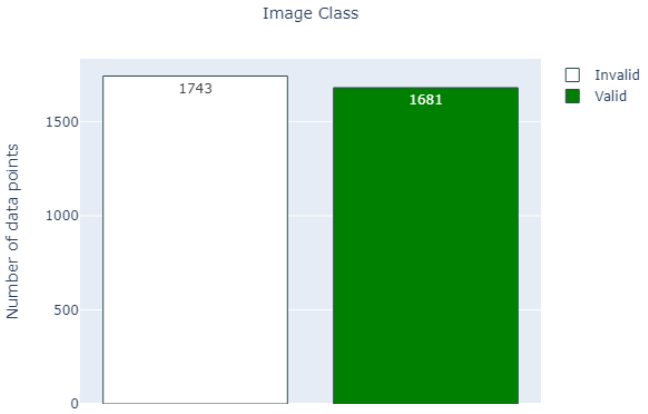

Fig. 5. Train Eelgrass Binary Image Class Distribution

This, in itself, was a very valuable finding, as it meant that more than half of all annotations made by the marine biologists were that the image was invalid. Therefore, purely by implementing a model that could classify and remove all the invalid images from the videos, this would significantly optimize time spent annotating.

2) Test Dataset: The test dataset contained 298 data points. These data points were all from one transect and from a completely different season than the train dataset. The marine biologists who annotated the dataset also commented that part of the transect differed significantly from what they were used to seeing, partly because the eelgrass was covered with sediment. Therefore, we believe this dataset serves as a great test dataset since it is completely separated from the train dataset by season and conditions, enabling us to better gauge if the model will predict well on unseen data. Interestingly, the distribution of eelgrass coverage labels was partially different in the context of valid and invalid images. As seen in Figure 16, the maximum label was not -1 (invalid image) like in the train dataset but 0. This becomes even clearer when you look at the valid/invalid class distribution in Figure 16.

D. Inter-annotator Agreement

The intraclass correlation metric was utilized to gauge the degree of agreement or disagreement between annotators involved in labeling the dataset. By calculating ICC, the aim is to quantify the consistency in the annotations provided by different annotators, thereby offering valuable insights into the robustness and reliability of the annotation process. This statistical measure is instrumental in understanding the extent to which annotators are aligned in their interpretations and evaluations, forming a crucial component of the broader analysis and interpretation of the research findings.

$$ICC = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_e^2}$$

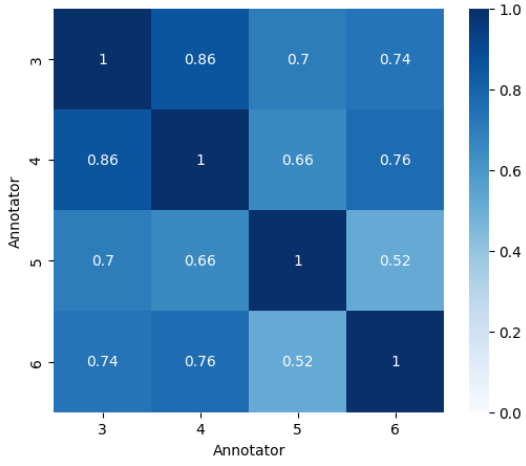

Fig. 6. Intraclass Correlation Confusion Matrix

The ICC values, ranging from 0 to 1, indicate the proportion of variance in the data that can be attributed to true differences between annotators relative to the total variance. Notably, the ICC between annotators 3 and 4 is high (0.857), suggesting a substantial agreement in their annotations. Annotators 5 and 6 exhibit a lower ICC of 0.519, indicating a moderate level of agreement. Additionally, the ICC between annotators 3 and 5 (0.704) and annotators 4 and 6 (0.762) falls in between, signifying a moderate to substantial level of agreement. These findings imply that there is variability in the agreement levels among annotators, with some pairs demonstrating higher consistency than others. Understanding such inter-annotator agreement is crucial for interpreting the reliability of the annotation process and refining the consistency of annotations across the dataset.

The variability in the Intraclass Correlation Coefficient (ICC) values among annotators (3, 4, 5, and 6) implies a level of disagreement in their dataset annotations. The ICC values, reflecting agreement levels between annotators, highlight that some annotator pairs do not exhibit a consistent interpretation of the data. In the context of a machine learning model, these divergences among annotators can pose challenges to both training and evaluation processes.

V. Machine Learning Experiments

A. Convolutional Neural Network

Employing computer vision methodologies, a Convolutional Neural Network (CNN) was implemented to address the binary task of predicting valid and invalid transect images, based on RGB images. The CNN comprises three convolutional layers using the Rectified Linear Unit (ReLU) activation function, with an input shape of (64, 64, 3). This architecture is well-suited for computer vision tasks, leveraging its ability to automatically learn hierarchical features from visual data. The chosen input shape strikes a balance between computational efficiency and preserving essential spatial and color information. The deployment of the CNN in this context underscores its efficacy in discerning intricate patterns within RGB images, thereby enhancing the accuracy and efficiency of ecological monitoring processes.

B. Vesselness Filter Testing and Feature Engineering

In exploring optimal representations for eelgrass through feature engineering, the application of a Vesselness filter, specifically the Frangi Filter, was investigated. Despite initial ineffectiveness due to border-related issues, subsequent attempts with line enhancement and edge detection tools, such as the hessian image filter, proved effective in capturing eelgrass stalks.

A Hessian filter is a computational technique used in image processing to analyze the local structure and detect key features such as edges, corners, and blobs within an image. It operates by applying the Hessian matrix to each pixel in the image, where the elements of the matrix represent the second-order partial derivatives of the image intensity with respect to spatial coordinates. The eigenvalues of this matrix convey

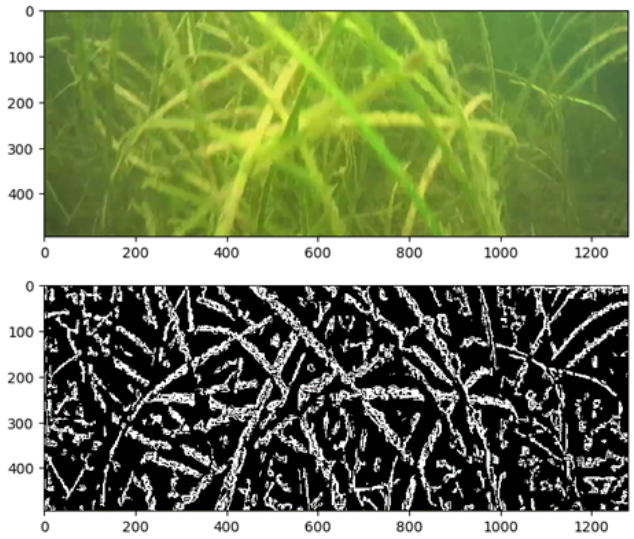

Fig. 7. Hessian filter applied to eelgrass image

information about the local curvature of the image intensity surface.

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

In practice, the Hessian filter is particularly adept at identifying regions with distinct shapes and structures. For instance, regions with high positive eigenvalues indicate edges, while regions with high negative eigenvalues often correspond to corners or blob-like structures. By leveraging these local curvature properties, the Hessian filter proves valuable in tasks such as feature extraction, object recognition, and image segmentation within the field of computer vision.

As seen above in Figure 7, this filter does well in bringing the stalks of the eelgrass into the foreground of the image, and should theoretically provide a great background for the model to learn eelgrass as a feature from.

C. Residual Neural Network

A Residual Neural Network (ResNet) was implemented, known for its efficacy in feature extraction from image data. The decision to employ ResNet was driven by its computational efficiency compared to a CNN, resulting in significantly reduced training times. The ResNet model implemented was pretrained on ImageNet using the ResNet50 architecture proposed by [8], which is currently well-known for being one of the most leading computer vision algorithms.

The realization of a two-step model emerged during ResNet implementation. Initially conceived for both training and inference, the two-step model was later streamlined into distinct binary and continuous models. Both models utilized a ResNet for image representation, with a classification head predicting valid or invalid eelgrass transect images in the binary model and a regression head predicting eelgrass coverage in the continuous model. The adoption of a unified model architecture for both tasks, employing a sigmoid function for values between 0 and 1, facilitated a consistent approach.

Residual Networks (ResNets) demonstrated effectiveness in processing time-series visual data due to their architecture, addressing challenges in training deep neural networks. Considering the input data's temporal nature, a time-series training data split was implemented.

Due to hardware limitations, it was necessary to implement transfer learning.

D. Transfer Learning

Transfer learning, leveraging a model trained on one task for a related second task, was employed to enhance model performance in scenarios with limited labeled data. A pretrained ResNet50 model, facilitated faster convergence and partially improved performance in predicting eelgrass coverage.

The methodology involved fine-tuning the pretrained model's parameters on the target task, utilizing its learned features as a foundation. This approach is advantageous, particularly when collecting extensive labeled data for the target task is impractical, as is the scenario in this project.

E. Vision Transformer

An experimental exploration into Vision Transformers, based on architecture proposed by [9], was conducted for its potential performance benefits in the context of eelgrass prediction. Transfer learning principles were applied using a pretrained Vision Transformer from the transformers Python library. Patterns and relationships captured by the transformer were subsequently input into a fully connected Feedforward Neural Network (FFNN), culminating in a sigmoid output layer providing final class probabilities.

VI. Results

A. Valid/Invalid Eelgrass Image Model Results

TABLE I
BINARY TASK RESULTS

Model	Filter	Best Epoch	Accuracy
CNN	None	111	0.619
CNN	Hess	147	0.618
ResNet50	None	2	0.537
ResNet50	Hess	80	0.542
ViT	None	46	**0.633**
ViT	Hess	5	0.509

In the Binary Task Results table (Table 1), a comprehensive evaluation of three distinct machine learning models—CNN, ResNet50, and ViT—is presented across binary classification tasks, considering variations in the presence of a Hess filter. The "Best Epoch" column captures the epoch at which each model achieved its peak accuracy during training. Notably, the Convolutional Neural Network (CNN) demonstrated a peak accuracy of 61.9%, achieved at epoch 111, while ResNet50 reached its maximum accuracy of 53.7% at epoch 2. Interestingly, the Vision Transformer (ViT) outperformed both CNN

and ResNet50, attaining a noteworthy accuracy of 63.3% at epoch 46. However, it's crucial to highlight that the ViT model with the Hess filter and the ResNet50 model without the filter yielded suboptimal results, with accuracies of 50.9% and an 53.7%, respectively. These lower accuracies suggest potential challenges or limitations in specific configurations, warranting further investigation into the impact of filtering mechanisms on model performance. In general it can be said that the transfer learning approach was very unsuccessful.

Fig. 8. CNN (Green) vs ViT (Red) Train Accuracy Curve

Figure 8 illustrates the noteworthy observation that the Vision Transformer demonstrates accelerated dataset learning in comparison to the Convolutional Neural Network (CNN). Specifically, the Vision Transformer manifests improved accuracy by epoch 26, contrasting with the CNN's requirement of over 100 epochs to initiate effective dataset learning. Subsequently, the CNN exhibits a swifter learning rate than the Vision Transformer. This phenomenon, however, is attributable to the CNN's susceptibility to overfitting the training dataset, evident in the consistent rise in accuracy without a corresponding enhancement in test dataset accuracy.

B. Eelgrass Coverage Model Results

TABLE II
REGRESSION TASK RESULTS

Model	Filter	L1 Loss
CNN	None	0.186
CNN	Hess	0.187
ResNet50	None	0.189
ResNet50	Hess	0.204
ViT	None	0.186
ViT	Hess	0.188

The results of the regression task are summarized in Table II. The table presents the performance of different models under two filter conditions, namely "None" and "Hess," with the evaluation metric reported as L1 Loss. The CNN model demonstrates a marginally lower L1 Loss with no filtering (0.186) compared to the case where the Hess filter is applied (0.187). For ResNet50 both models do not provide any performance improvement. The Vision Transformer (ViT) model exhibits a slightly improved performance with no filtering (0.186) compared to the Hess filter condition (0.188). These findings suggest that, across the models evaluated, the impact of the Hess filter varies, with some models experiencing a slight increase in L1 Loss when the filter is applied, while others, such as CNN, exhibit a negligible difference. The nuanced effects of filtering on different models highlight the importance of considering both model architecture and filter type in regression tasks.

As also becomes clear in Figure 9, even the best performing model the Vision Transformer with no filter, does not succeed at learning any distinctiveness between the regression classes.

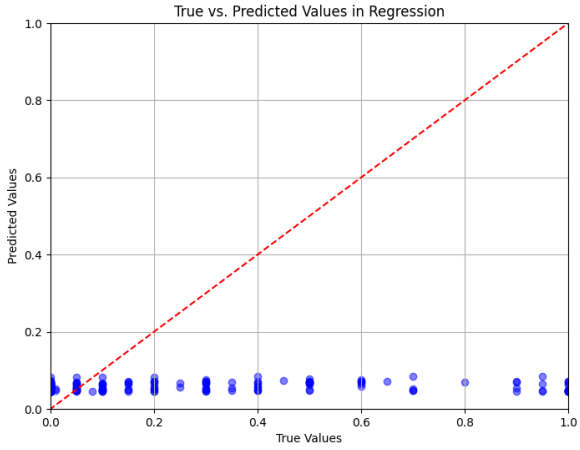

Fig. 9. ViT True vs Predicted Values Epoch 433

C. Principal Component Analysis (PCA) for Model Performance Analysis

It was decided to employ Principal Component Analysis (PCA) to better understand the poor results produced by the machine learning experiments. Both ResNet and ViT serve as powerful architectures for capturing intricate patterns within image datasets. The application of PCA to their output features enables the identification of major sources of variance and potential redundancies or noise impacting model performance.

PCA transforms the original feature space into a new set of uncorrelated variables, known as principal components, reducing dimensionality while retaining essential information. Visualizing these principal components provides insights into the factors contributing to suboptimal model performance. This approach offers a nuanced perspective on feature interactions, potentially uncovering patterns or discrepancies not immediately evident through standard performance metrics.

As becomes clear in Figure 10, the results of the PCA analysis revealed a striking outcome, indicating a lack of visual discernibility between the various regression classes. Despite the rich and complex feature representation provided by the Vision Transformer, the PCA results suggest that, in the reduced-dimensional space, the inherent characteristics that distinguish between regression classes may not be readily apparent.

Similar outcomes were noted across all other models and filters employed. For instance, the Vision Transformer model applied to the binary classification task also fails to produce

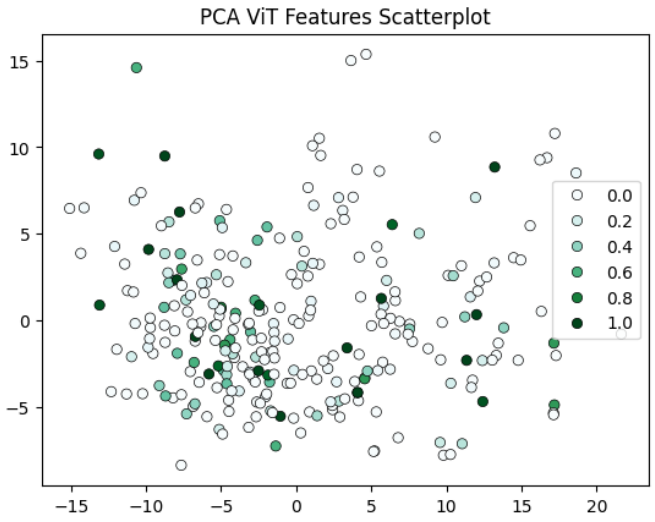

Fig. 10. PCA on test dataset, ViT feature extraction, eelgrass coverage as Label

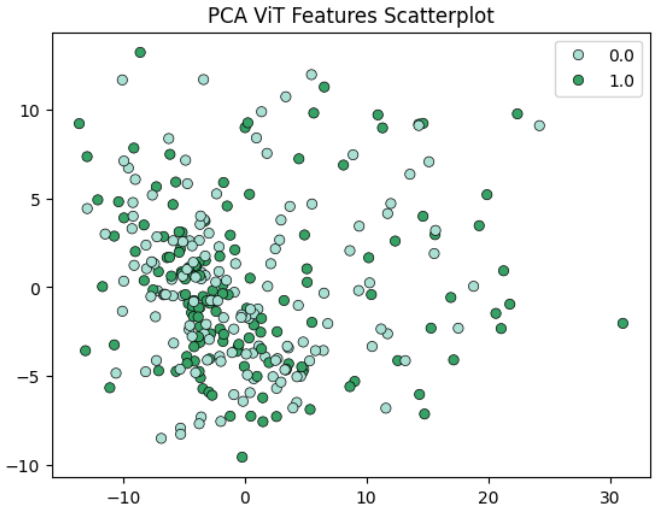

Fig. 11. PCA on ViT feature extraction on the binary task

a discernible visual distinction between the binary classes, as seen in Figure 11.

VII. Discussion

The outcomes indicate that none of the models effectively address the assigned tasks, prompting an exploration into the underlying factors contributing to this result.

One potential explanation considered is the discrepancies in the test dataset features or output classes compared to the training dataset. The annotation methodology for the test data differs significantly, using Label Studio instead of the cross-referencing video and Excel tables approach. However, this hypothesis is dismissed, as the observed shift in class distributions primarily involves two classes—invalid image and 0% eelgrass coverage — both of which significantly influences the binary task, yet fail to explain the lack of observed performance discrepancy.

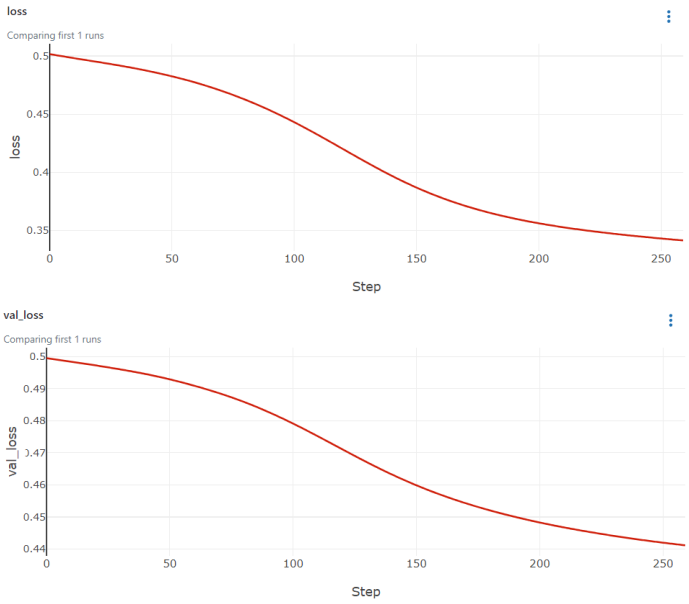

Fig. 12. Best Binary task ViT Model Loss Plots

Furthermore, the loss plots of the most proficient model for the binary task, the Vision Transformer, depict a textbook-like loss function for both the training and test datasets (see Figure 12). This observation suggests that the model effectively generalizes to unseen data without succumbing to overfitting on the training dataset.

Consequently, the deduction follows that the hypothesis positing insufficient data as a contributing factor, implying an inadequate quantity of data for training the model on previously unseen data, and the hypothesis of a too diverging test dataset, can be dismissed.

An alternative explanation to consider is the converse phenomenon—underfitting. Underfitting occurs when a model falls short of capturing the inherent patterns in the data, resulting in poor generalization. The potential occurrence of underfitting is suggested by the models' inability to effectively address the assigned tasks. This deficiency may be ascribed to a model lacking the capability to comprehend and represent the inherent complexities within the dataset. The disparities between the test and training datasets, coupled with the failure to enhance task performance, serve as indicators of underfitting. Notably, certain models exhibit significant difficulty surpassing a 60% accuracy threshold in the binary task, suggesting an inability to grasp the intricacies of the valid/invalid transect image features.

Moreover, insufficient feature engineering could contribute to these issues. The dismissal of model complexity as a contributing factor is grounded in the successful learning of the data by both the "standard" CNN and the Vision Transformer, as evidenced by the respective loss curves.

In addition to examining the factors contributing to the suboptimal performance of machine learning models, it is crucial to explore the extent to which humans can reliably assess eelgrass coverage levels. The variability in the definition of eelgrass coverage among annotators is evident from the inter-annotator agreement and Intra-Class Correlation (ICC) values. Given that the training dataset was annotated by annotators with the highest variability (annotators 5 & 6), it becomes apparent why the dataset might pose significant challenges for different models, considering potential substantial divergences in eelgrass coverage definitions.

Discussions among annotators revealed that the definition of eelgrass coverage is subjective and varies based on individual experiences. For instance, an annotator's perception of a 5% change in coverage might differ based on their recent exposure to 100% coverage versus 0% coverage. Notably, one annotator highlighted significant observable changes between 0% and 20% coverage but perceived no substantial differences between 50% and 95% coverage.

This variability in defining eelgrass coverage is mirrored in existing literature, as is seen in the differences between [4] and [5] approaches.

It is important to note that this does not mean that the current annotation system is in-accurate, only that it is unsuited for machine learning tasks. [7] standard methodology for doing visual eelgrass transects surveys is designed in the context of producing an estimation of the eelgrass coverage in a large area, and not designed in the context of analysing the eelgrass coverage in that specific frame, or time and place, as is the goal of the computer vision models.

Despite these challenges, a significant realization emerges: employing machine learning algorithms or computer vision models not only saves time and enhances efficiency but also mitigates subjectivity inherent in human annotation and the varied definitions of eelgrass coverage.

VIII. Conclusion

In conclusion, this research project focused on automating the estimation of visual eelgrass (Zostera Marina) coverage through machine learning models. Through the application of Principal Component Analysis (PCA) and analysis of machine learning model performance, the study revealed insights into the representation of the test dataset and the efficiency gained by using the Label Studio setup for transect analysis.

The findings suggest the need for further exploration to determine optimal methods for describing eelgrass coverage features. The project, accompanied by its dataset and repository, stands as a foundational resource for future research in the field, addressing issues of reproducibility and enabling others to utilize the data for their own machine learning experiments. Despite challenges related to subjectivity in human annotation, the project highlights the potential of machine learning algorithms and computer vision models to save time, enhance efficiency, and mitigate variations in defining eelgrass coverage.

In summary, the study contributes to the understanding of eelgrass transect analysis, providing a basis for future endeavors in the automation of analysis methods that can further contribute to the conservation and management of this essential marine resource.

Acknowledgment

Many thanks to Veronika Cheplygina for supervising me throughout the project and providing valuable guidance. Many thanks as well to all individuals at the company who helped me with the annotation and all marine biology and data related questions.

References

[1] M. Gonciarz, J. Wiktor, A. Tatarek, P. Węgleński, and A. Stanković, "Genetic characteristics of three baltic zostera marina populations**this work was financially supported by the project: 'ZOSTERA: Restoration of ecosystem key elements in the inner puck bay'.," vol. 56, no. 3, pp. 549–564.

[2] J. W. Fourqurean, C. M. Duarte, H. Kennedy, N. Marbà, M. Holmer, M. A. Mateo, E. T. Apostolaki, G. A. Kendrick, D. Krause-Jensen, K. J. McGlathery, and O. Serrano, "Seagrass ecosystems as a globally significant carbon stock," vol. 5, no. 7, pp. 505–509. Number: 7 Publisher: Nature Publishing Group.

[3] S. Cooper, A. Schmidt, and J. Barrell, "DOES EELGRASS (zostera marina) MEET THE CRITERIA AS AN ECOLOGICALLY SIGNIFICANT SPECIES,"

[4] S. Sengupta, B. K. Ersbøll, and A. Stockmarr, "SeaGrassDetect: A novel method for the detection of seagrass from unlabelled underwater videos," vol. 57, p. 101083.

[5] G. Reus, T. Moller, J. Jager, S. T. Schultz, C. Kruschel, J. Hasenauer, V. Wolff, and K. Fricke-Neuderth, "Looking for seagrass: Deep learning for visual coverage estimation," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–6, IEEE.

[6] Y. Li, H. Lu, J. Li, X. Li, Y. Li, and S. Serikawa, "Underwater image descattering and classification by deep neural network," vol. 54, pp. 68–77.

[7] . P +4560919639, "AU ecoscience - marint fagdatacenters gældende tekniske anvisninger."

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition."

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale."

Appendix

The Company

The companies name has been redacted due to confidentiality reasons.

Hardware Used

All experiments were trained on a Dell Precision 5560 (32GB RAM, i7-11800H) using a 4GB Nvidia Quadro T1200 for all pytorch CUDA tasks.

Extra Figures

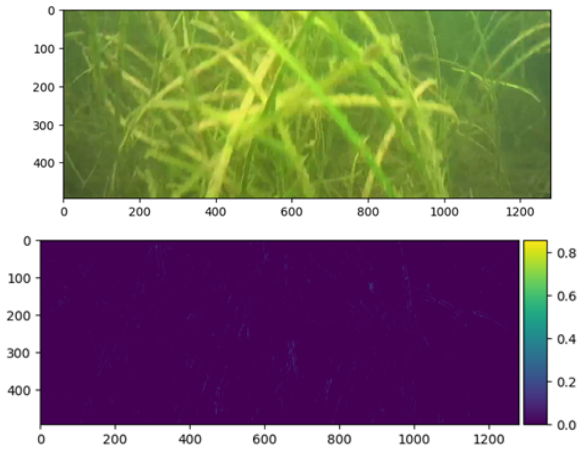

Fig. 13. Frangi Filter

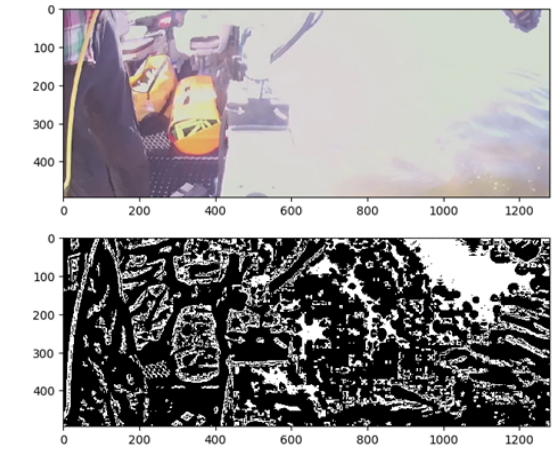

Fig. 14. Hessian filter applied to invalid image

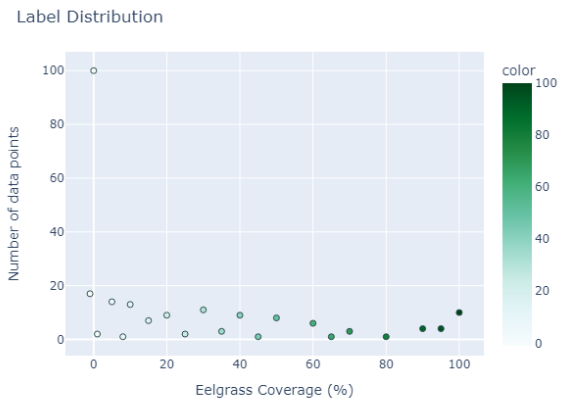

Fig. 15. Test Dataset Eelgrass Coverage Label Distribution (-1 is invalid image)

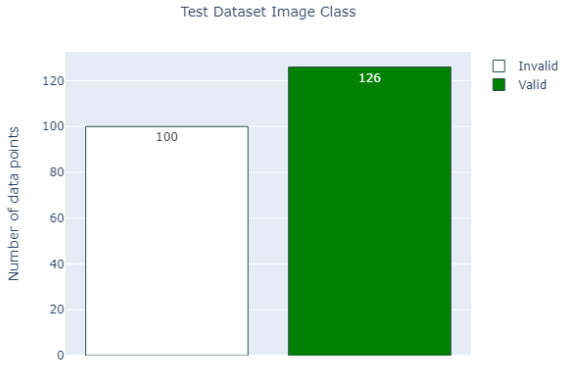

Fig. 16. Test Eelgrass Binary Image Class Distribution

Fig. 17. Eelgrass (Top), Manatee Grass (Middle) and Neptune Grass (Bottom)