Counting Oyster Seeds with Stardist2D: Comparative Model Evaluation and Raspberry Pi Deployment

Oyster Team-Spring 2025
Department of Mathematics, Louisiana State University

Introduction:

The Spring 2025 Oyster Team extended the automation project to increase the accuracy and accessibility of oyster seed counting through deep learning. This semester, we collaborated with the Louisiana Sea Grant Research Lab [1] specially with Dr. Sarah Bodenstein [2] to develop, evaluate, and deploy a StarDist2D-based model for counting 4-6 mm oyster seeds. Building on past work in the 0-2 mm and 2-4 mm ranges, the goal this semester was to create a robust model, compare it to previous versions, and deploy it in the field using a Raspberry Pi.

Figure 1: Oyster Farm (Top), Oyster (Below)

Data Collection:

Oyster seeds samples were collected, filtered and classified into three categories: 0-2mm, 2-4mm and 4-6mm. Then the images were captured. Louisiana Sea Grant lab [1] provided the dataset of these images. For Spring-2025, the special focus was on the dataset of 4-6mm size range. A total of 31 images were annotated with Fiji and the LabKit plugin, which allowed exact labeling of individual seeds. This annotated dataset formed the basis for training and evaluating the deep learning model.

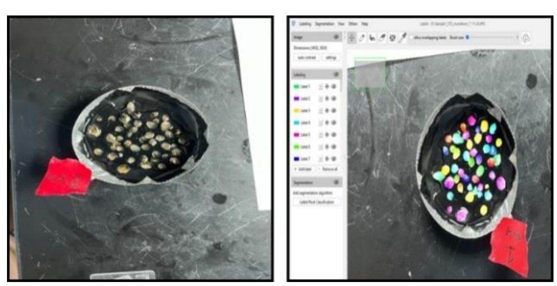


Figure 2: Original Oyster image (Left), Annotated image (Right)

Methodology:

The model for this semester was developed using StarDist2D [3] [4], a deep learning segmentation architecture that is specifically designed to identify star-convex shapes, such as oyster seeds. A U-net CNN is incorporated into model architecture, which generates two key predictions:

- **Pixel-wise object probability**, which calculates the likelihood that a pixel belongs to a seed.
- **Radial distances** (32 rays) between each center pixel and the seed's edge, capturing the seed shape.

To improve model generalizability and prevent overfitting, intermediate augmentation techniques were used which include rotations, flipping, and shifting. These modified the training set while maintaining biological accuracy.

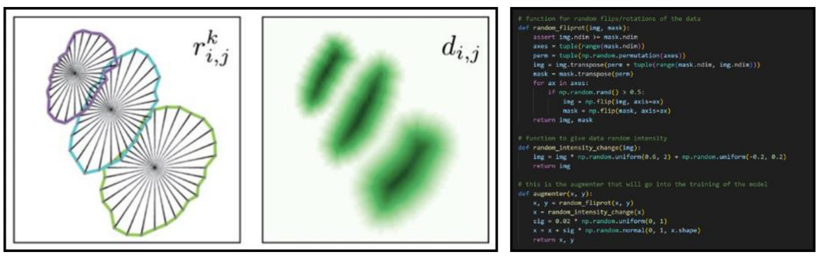


Figure 3: Star Convex Polygons (Left), Augmentation Function (Right)

Training:

The dataset was split 80-20 into training and testing subsets. The model was trained for both 300 and 500 epochs to measure performance and check for overfitting.

The model was trained with loss functions which included probability loss and distance loss, which contributed to improved segmentation accuracy.

- **Probability loss:** Probability loss ensures that each pixel is appropriately classified as either oyster seed or background. This is done using a binary cross-entropy formulation which enables precise detection of seed regions.
- **Distance Loss:** Distance loss is the difference between expected and ground-truth radial distances from the center of a detected seed to its boundary which is calculated in 32 directions. This component improves the model's capacity to accurately capture the form and size of each seed.

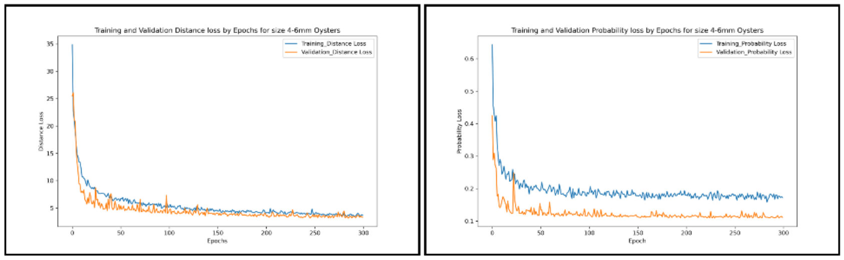


Figure 4: Probability Loss (Left), Distance Loss (Right) for 300 epochs

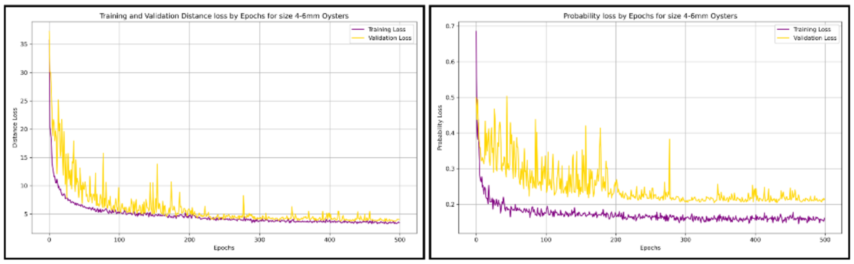


Figure 5: Probability Loss (Left), Distance Loss (Right) for 500 epochs

Training showed convergence of both loss components, especially in the first 50 epochs. The half dataset (15 images) showed a decrease of probability loss from 0.6 to 0.2 and distance loss stabilizing. The whole dataset (31 images) showed similar trends, however overfitting increased after 300 epochs. The training-validation loss gap remained less, which indicates stable model generalization.

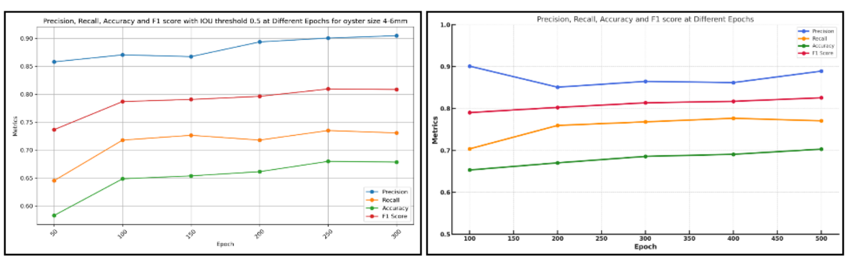
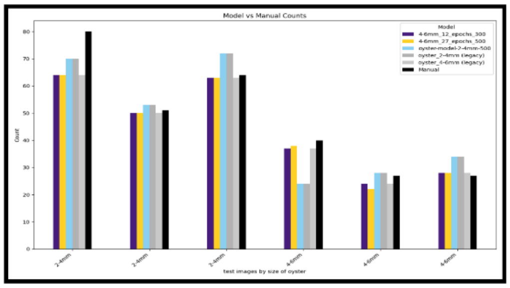


Figure 6: Precision, F1 score, Recall, Accuracy of two models- 15(Left), 31(Right)

Model Performance:

The model was trained on 31 annotated images using an 80–20 split. It had 98.48% accuracy and 1.0 MAE (Mean Absolute Error) at 300 epochs. Overfitting caused accuracy to decline to 93.34% and MAE to rise to 2.0 after 500 epochs. Throughout the training, precision, recall, and F1 score improved. Training with 300 epochs yields high model performance. When training was done till 300 epochs, the result showed good model performance. For evaluation the IoU (Intersection over Union) threshold of 0.3 was chosen.



Model	2-4mm Acc% (MAE)	4-6mm Acc% (MAE)	4mm Acc% (MAE)
4-6mm_12_epochs_300	87.31% (10.12)	**98.48% (1.0)**	**93.34% (2.0)**
4-6mm_27_epochs_500	88.22% (9.62)	93.94% (4.0)	91.67% (2.4)
oyster-model-2-4mm-500	**93.24% (5.12)**	87.88% (8.0)	82.45% (5.8)
oyster_2-4mm (legacy)	93.24% (5.12)	87.88% (8.0)	82.45% (5.8)
oyster_4-6mm (legacy)	87.31% (10.12)	98.48% (1.0)	93.34% (2.0)

MAE = Mean Absolute Error (average absolute deviation in shell counts)

Figure 7: Model vs Manual Count

Graphical User Interface (GUI):

A graphical user interface [5] was developed to support the detection of oyster seeds measuring 4–6 mm. The interface enables users to upload images, select trained models, and view annotated predictions alongside the total seed count, all within a single platform. In a specific test case, the model predicted 70 seeds for an image, while the actual count was 66, demonstrating a high level of accuracy and practical reliability.

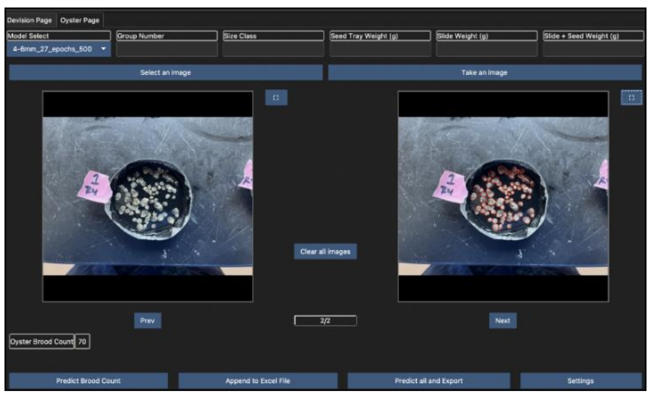


Figure 8: GUI Prediction of Oyster image (4-6 mm)

Raspberry Pi Deployment:

To support real world use, the trained model and GUI were effectively deployed on a Raspberry Pi device. The Raspberry Pi was selected because of its portability, minimal cost, and compatibility with Linux-based software.

This deployment included creating a precompiled binary for the GUI which could be installed with a single click. For field use a touchscreen and battery pack were added. Benchmarking revealed that the system could process and annotate one image in approximately 60 seconds, three times faster than hand counting.

A 3D-printed casing was created to contain the gadget, and a video instruction [5] was created to help with setup, making the system usable by non-technical users in hatchery and research environment.

Figure 9: Raspberry Pi implementation

Future work:

- Enhance the usability of the GUI interface and resolve minor functional issues.
- Increase model detection to all oyster seed sizes (0–6 mm).
- Finalize and describe the Raspberry Pi configuration for broader user adoption.
- Adjust the design of the 3D-printed housing according to the comments made during the trial run.
- Prepare a research article that outlines the methodology, presents the results, and describes the deployment framework.
- Create detailed user documentation for replication and maintenance.

Acknowledgement:

We would like to thank:

- Dr. Peter Wolenski and Dr. Nadejda Drenska for guiding and supporting us.
- Dr. Sarah Bodenstein, Post Doc, Louisiana Sea grant research lab, for her collaboration.
- Elizabeth M. Robinson and the Michael C. Voisin Oyster Hatchery for research support.
- Gowri Priya Sunkara, Dow Draper for their assistance with machine learning.
- Nikkos Svoboda for introducing and supporting the use of the *Chaos* computing system.
- LSU Department of Mathematics for infrastructure and project support.

Oyster team-Spring 2025:

Department of Mathematics Advisors:	• Dr. Peter Wolenski • Dr. Nadejda Drenska
Team Members:	• Sanjida Nupur • Mark C Herpin • Maxwell T Shroyer
GUI Team Members:	• Dow Draper • Paul Yeon
Deployment Lead (Raspberry Pi)	• Maxwell T Shroyer

References:

[1] L. S. Grant, "Oyster Research Lab," n.d.. [Online]. Available: https://www.laseagrant.org/outreach/aquaculture/oyster-research-lab/lab-staff/.

[2] D. S. Bodenstein, "Personal Website," n.d.. [Online]. Available: https://www.sarahbodenstein.com.

[3] U. W. M. B. C. &. M. G. Schmidt, "Cell Detection with Star-Convex Polygons," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2018*, 2018.

[4] M. S. U. H. R. S. K. &. M. G. Weigert, "Star-Convex Polyhedra for 3D Object Detection and Segmentation in Microscopy," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[5] L. DeVision, "GUI Development Repository," n.d.. [Online]. Available: https://github.com/LSU-Devision/GUI/.