A Comparative Study of StarDist and Cellpose for 0-2mm Oyster Seed Detection

Oyster Team

Summer 2026

Sea Grant

Louisiana Sea Grant is an organization tasked with aiding in marine and coastal life conservation, protection, and research.

Figure: Oyster seed of size 0-2 mm: Previous dataset(left), New Dataset(Right)

Introduction

Project Goals in Summer 2026: The goal of the project is to:

- Develop a deep learning model to accurately count 0-2 mm oyster seeds
- Previous models achieved strong performance on 2-4 mm and 4-6 mm oyster seeds, but model accuracy was significantly lower for the 0-2 mm
- Train and test the dataset using **StarDist** and **Cellpose**.
- Compare both methods using counting accuracy, prediction quality, and count error.

Seed Collection

- Oyster seeds are collected and separated by size using filters.
- The size categories include 0–2 mm, 2–4 mm, and 4–6 mm.
- This semester's target category is **0–2 mm**.

Figure: Bottle and filtering process

Image Acquisition Setup

- Images were collected for 0–2 mm oyster seeds.
- The goal is to support a low-cost and portable imaging workflow.
- dataset size-30 Images

Figure: Team members

Augmentation

What is Data Augmentation? Data augmentation is the process of creating additional training samples by applying transformations to existing images.

Examples include:

- Rotation, Flipping and shifting
- Brightness adjustment
- Cropping and noise addition

Why is it Used? Data augmentation introduces additional variation into the training dataset.

It helps to:

- Reduce model overfitting, Improve generalization
- Increase training-data diversity and
- Reduce the need for new images

Main Idea: Augmentation helps the model learn from more diverse image variations without collecting a completely new dataset.

Augmentation

To mitigate issues like data scarcity and model overfitting, augmentation techniques can be classified into four categories:

- **Absent Augmentation**: No changes applied, providing a baseline comparison.
- **Basic Augmentation**: Simple transformations, including adjustments in brightness and contrast, to account for minor visual variations.
- **Intermediate Augmentation**: Adds techniques like rotation, flipping, and shifting to increase dataset diversity.
- **Advanced Augmentation**: Builds on prior methods, incorporating zoom and shear transformations to introduce further variability.

Augmentation

- **Selected Method**: (Intermediate Augmentation-90°Rotation)
  - Each selected training image was rotated by 90°.
  - No flipping, shifting, zooming, shearing, brightness adjustment, or noise was applied.
  - For segmentation, the same rotation was applied to both the image and its corresponding mask.

Figure: Augmentation functions applied to StarDist model

Stardist

Basics

- Uses a neural network, based on U-net, to identify star-convex polygons
- Probabilities are obtained for each pixel saying how likely a pixel is to be in a polygon
  - Probabilities found using the neural network
- 32 radial directions measured from a pixel to the boundary of the polygon
- Better at predicting cell shape than other image analysis methods

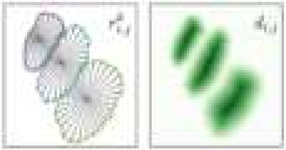


Figure: Star-Convex Polygons

Cellpose

Basics

- A deep learning model that automatically identifies and outlines cells in microscope images, trained across dozens of diverse biological datasets
- Supports fluorescence, brightfield, phase-contrast, and histology images without needing separate models for each
- Cellpose-SAM (2025) uses a large foundation model to improve accuracy and generalization

Cellpose-SAM

How It Works

- Rather than directly detecting cell boundaries, Cellpose predicts a vector flow field
  - This approach is more robust to irregular shapes and touching cells, as well as noise, blur, downsampling, and variable cell sizes
- The "SAM" component refers to Meta's Segment Anything Model
  - In Cellpose-SAM's, the SAM's decoder is removed and replaced with Cellpose's flow prediction system
- Can be fine-tuned on small custom datasets using standard consumer GPUs

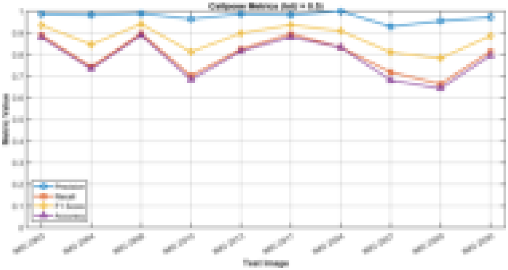
Figure: Predicted Vector Flow Fields

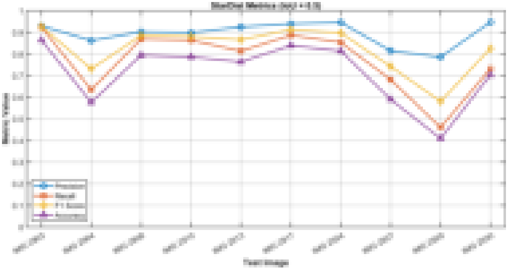
Figure: Final Segmentation Masks

Cellpose vs Stardist : Segmentation Approaches

- StarDist models each cell as a star-convex polygon
  - Well-suited for round or convex objects
  - Performance degrades significantly for elongated, irregular, or heavily overlapping cells
- Cellpose-SAM learns a continuous flow field that doesn't rely on or assume any particular shape or structure
  - Handles irregular, elongated, and densely packed cells with greater consistency
  - Improves its ability to work on new data by being pre-trained on a large amount of data

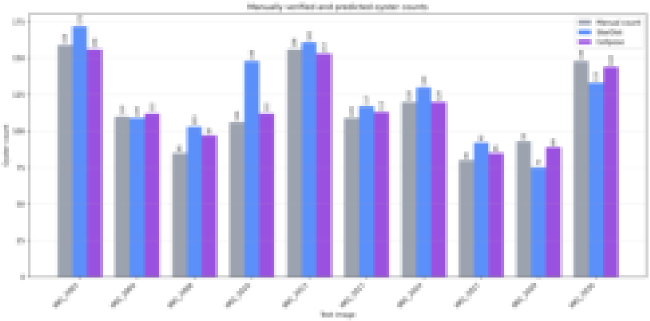
Evaluation Metrics



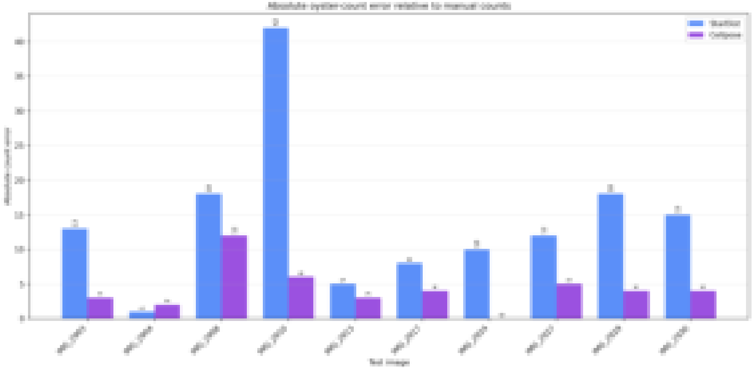


Manual vs. Predicted Counts

Cellpose and StarDist instance counts vs. manual reference counts, full test set



Absolute Count Error

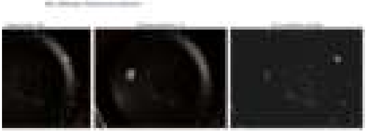


- Cellpose achieved lower count error on most test images.
- StarDist showed greater variability across the dataset.

Cellpose Vs StarDist : Performance Comparison

Lowest object-recall of the whole test set for StarDist

Cellpose overlay



StarDist: ground truth (left) vs. prediction (right)

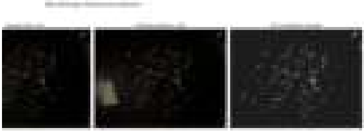
	Ground truth	Detected	Recall
Cellpose	128	89	69.5%
StarDist	128	75	58.6%

On this frame StarDist posts its lowest recall of the ten-image test set (58.6%), while Cellpose though not flawless recovers noticeably more of the annotated oyster seed.

Cellpose Vs StarDist : Performance Comparison

StarDist nearly matches ground truth; Cellpose under-detects

Cellpose overlay



StarDist: ground truth (left) vs. prediction (right)

	Ground truth	Detected	Recall
Cellpose	154	112	72.7%
StarDist	154	148	96.1%

Here StarDist recovers 96.1% of annotated oyster seed (near-perfect), while Cellpose has its second-lowest recall (72.7%) of the test set, missing a substantial fraction of small/dim objects.

GUI Prediction v1.6 Updates

Changes in v1.6

New Features

- Cellpose & StarDist **0–2 mm** models added
- All 4 models **load automatically**
- **NVIDIA CUDA GPU** support
- Live **GUI progress bar** during prediction
- GPU↔CPU **cache bug** fixed

Known Issues

- Weights **not bundled** (1.19 GB $>$ GitHub limit)
- **AMD GPU** not yet supported

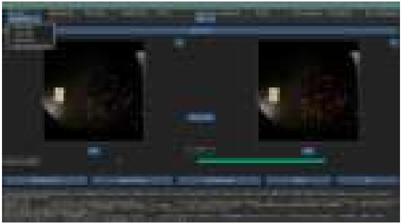


Figure: Cellpose 0–2 mm prediction from Image 2903 (156 seeds detected)

GUI Prediction Pros & Cons

Cellpose vs StarDist for Oyster Seed Counting

Cellpose (PyTorch + SAM2)

Advantages

- Better at detecting **irregular 0–2 mm** seeds in images
- More accurate on **densely packed** seeds
- SAM2 adapts well to **varying seed appearances**
- More **robust to image noise** and lighting variation

Disadvantages

- Weights require **manual setup** (1.19 GB)
- Needs **NVIDIA GPU** for practical prediction speed
- **Very slow on CPU** for oyster seed images

StarDist (TensorFlow)

Advantages

- **Lightweight** no large weight files needed
- Runs reliably on **CPU** for image prediction
- Quick and **easy to set up** for new users
- Sufficient for seeds with **rounder shapes**

Disadvantages

- **0–2 mm seeds are irregular** star-convex assumption often fails
- Less accurate when seeds are **overlapping** in the image
- Counting accuracy **drops** with varied seed appearances

Recommendation: Use StarDist for CPU-only machines; use Cellpose with NVIDIA GPU for highest seed counting accuracy.

Future Work

- Increase the 0–2 mm dataset size and make the models robust.
- Testing the GUI with the models in Grand Isle Lab
- Integrate the best-performing model with the Raspberry Pi setup.
- Dedicated **GUI team** recommended
- Weight size **may reduce** in future versions
- **AMD** and CPU support planned

Acknowledgments

We would like to thank

- Dr. Peter Wolenski, Dr. Nadejda Drenska for guiding and supporting us.
- The Department of Mathematics for foreseeing the utility of "Chaos" and the subsequent purchase.
- Elizabeth M. Robinson,Director, Louisiana Sea Grant Research Lab and Michael C. Viosin Oyster Hatchery for giving us an opportunity to apply the Machine learning Algorithm on the Oysters project.
- Nikkos Svoboda, Computer Analyst for introducing and making us familiar with the workings of the high-performance computing system "Chaos".

Acknowledgements

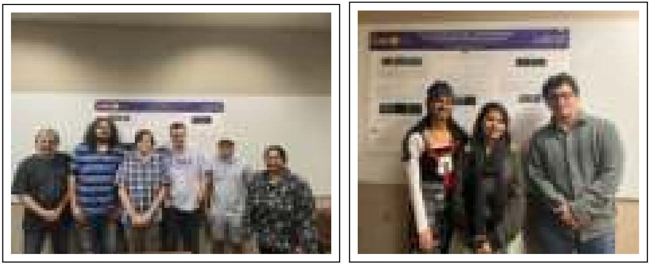


Figure: Previous Oyster Teams

References

References:

- Schmidt, U., Weigert, M., Broaddus, C., Myers, G. (2018). Cell Detection with Star-Convex Polygons. Medical Image Computing and Computer Assisted Intervention MICCAI 2018.
- Cellpose-SAM: superhuman generalization for cellular segmentation. Marius Pachitariu, Michael Rariden, Carsen Stringer bioRxiv 2025.04.28.651001; doi: https://doi.org/10.1101/2025.04.28.651001
- https://github.com/LSU-Devision/Summer-2024
-
- https://www.laseagrant.org/outreach/aquaculture/oyster-research-lab/lab-staff/about/
-
- https://github.com/LSU-Devision/GUI
- https://github.com/LSU-Devision/Louisiana-Sea-Grant.

Team Members

Department of Mathematics Advisors:

- Dr. Peter Wolenski
- Dr. Nadejda Drenska

Team Members:

- Sanjida Nupur
- Tasnim Tanha Mim
- Esrat Nur
- Briana Cox
- Emmanuel Asante
- Gowri Priya Sunkara
- Asad Swati
- Tre' Henson

Thank you for listening!