

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

Oyster Team – Summer 2026

Department of Mathematics, Louisiana State University

Introduction

The Summer 2026 Oyster Team continued the automated oyster-seed counting project in collaboration with the Louisiana Sea Grant Research Lab [1]. Accurate oyster-seed counts are important for monitoring seed growth, mortality, stocking density, and nursery production. However, manual counting can be time-consuming, particularly when working with very small oyster seed [10].

Previous phases of the project developed an automated image-based oyster-seed counting system using StarDist. The published study demonstrated strong performance for larger oyster-seed size classes, particularly 2–4 mm and 4–6 mm seeds, and showed that software-based counts could provide an efficient alternative to manual counting [10]. However, the 0–2 mm size category remained a major challenge. In the previous study, high-quality images of 0–2 mm seed suitable for model development could not be obtained using the imaging setup available at that time [10].

For the Summer 2026 project, the focus therefore shifted specifically to the 0–2 mm size category. In the images collected for the current project, individual seeds were extremely small and could appear in dense groups with weak visual separation, making both manual counting and automated instance detection more challenging.



Figure 1: Oyster Research and Demonstration Farm (Top), oyster nursery environment (Bottom Left), and oyster specimen (Bottom Right).

To address the limitations associated with the 0–2 mm category, two deep-learning instance-segmentation approaches, StarDist and Cellpose/Cellpose-SAM, were evaluated. StarDist represents objects using star-

convex polygons [2], whereas Cellpose uses a flow-based representation for instance segmentation [3]. Cellpose-SAM further extends the Cellpose framework using a pretrained foundation-model architecture [4].

The main goals of this project were to evaluate both methods on the 0–2 mm oyster-seed dataset, compare their detection and counting performance, examine prediction quality and count error, and incorporate the new models into the existing oyster-counting graphical user interface.

Data Collection, Image Acquisition, and Preparation

For the Summer 2026 project, the team collected and prepared image data specifically for the 0–2 mm oyster-seed category. The oyster-seed images were captured using a mobile phone under controlled imaging conditions. A dark background was used to provide clear visual contrast between the oyster seeds and the surrounding area.



Figure 2: Oyster seed samples (Left) and a representative 0–2 mm oyster-seed image captured using a mobile phone under controlled imaging conditions (Right).

After image acquisition, the oyster seeds in each image were carefully counted by hand to establish manual reference counts. Manual counting was particularly challenging and time-consuming because the seeds were only 0–2 mm in size and many individual seeds could appear together within a single image. These manually verified counts were later used as reference values for evaluating the counting performance of StarDist and Cellpose.

For instance-level evaluation, ground-truth annotations were also prepared to identify individual oyster seeds in the images. Particular care was required when seeds were closely spaced, clustered, dim, or touching. The annotations provided the reference objects needed to compare the individual instances detected by the two models.

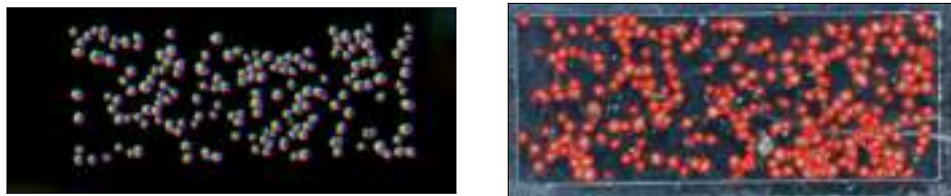


Figure 3: Representative ground-truth oyster-seed annotations used for instance-level model evaluation.

The final dataset contained 116 images with a resolution of 4032×3024 pixels in JPG format. Of these, 106 images were used for model development and training. The remaining 10 images were kept

completely separate as a held-out test set. These held-out images were not used during training and were reserved for the final evaluation and comparison of StarDist and Cellpose.

Methodology

Two deep-learning instance-segmentation approaches were used in this project for detecting and counting 0–2 mm oyster seeds: StarDist and Cellpose/Cellpose-SAM. Although both methods identify individual objects, they use different representations for instance segmentation. StarDist represents objects using star-convex polygons [2], whereas Cellpose uses predicted spatial flows [3].

StarDist

StarDist is a deep-learning-based instance-segmentation method that represents individual objects using star-convex polygons. The model uses a U-Net-based neural-network architecture to process an input image and predict object information at the pixel level [2].

For each pixel, StarDist predicts an object probability together with radial distances from that pixel toward the boundary of the corresponding object. These radial distances are measured along a fixed set of directions and together form a star-convex polygon that approximates the shape of an individual object [2].

In the configuration used in this project, 32 radial directions were used for representing oyster-seed objects.

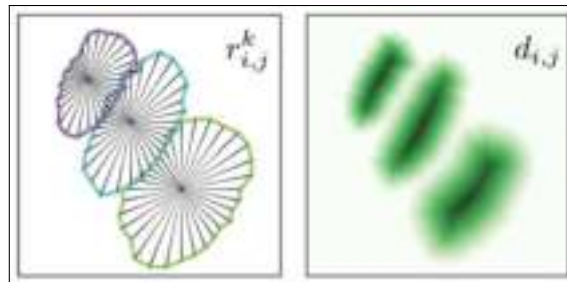


Figure 4: Illustration of the star-convex polygon representation used by StarDist for instance segmentation.

During prediction, pixels with high object probabilities provide candidate object locations. Their corresponding radial-distance predictions are used to reconstruct candidate star-convex polygons, which are then processed to produce separate object instances [2].

This representation allows individual objects to be identified separately rather than treating all foreground pixels as one region. The star-convex representation is particularly useful for objects that can be approximated by star-convex shapes [2].

In this project, a custom StarDist model was trained using the oyster-seed images so that the model could learn the visual characteristics of the 0–2 mm seeds. Because oyster seeds can also appear irregular, touching, or closely spaced, StarDist was compared with Cellpose, which uses a different representation for instance segmentation.

Cellpose / Cellpose-SAM

Cellpose is a generalist deep-learning instance-segmentation framework originally developed for biological image segmentation [3]. Instead of describing objects using a predefined polygonal shape, Cellpose predicts spatial vector flows associated with pixels belonging to individual objects.

The predicted vector field is used to group pixels into separate object instances. This flow-based representation allows Cellpose to segment objects with different shapes without requiring a fixed geometric representation [3].

This characteristic was relevant to the 0–2 mm oyster-seed dataset, where individual seeds can appear irregular, closely spaced, or touching.

The project also evaluated Cellpose-SAM. Cellpose-SAM extends the Cellpose approach using a pre-trained foundation-model architecture and retains the flow-based instance-segmentation formulation [4].

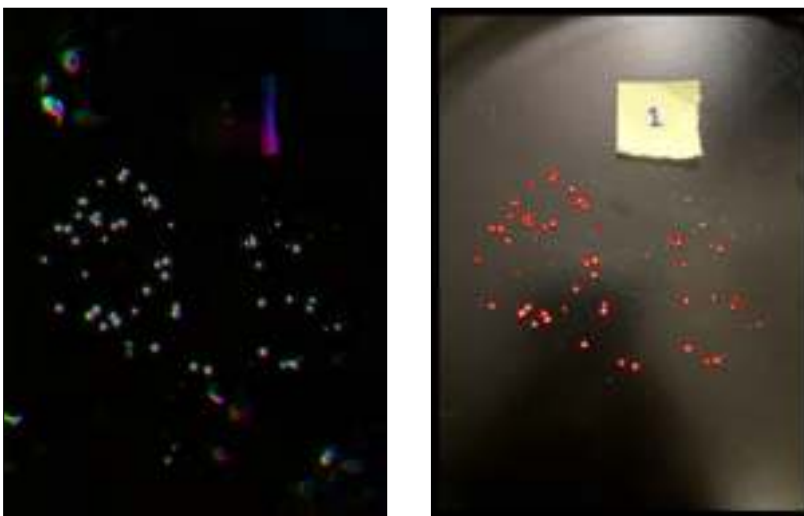


Figure 5: Predicted vector flow field (Left) and the resulting Cellpose segmentation masks (Right).

Pretrained Cellpose models can be adapted to specialized image datasets through fine-tuning [3]. This allows pretrained representations to be adapted to a specific image domain while retaining the flow-based segmentation framework.

In this project, Cellpose was applied to the 0–2 mm oyster-seed dataset and evaluated alongside the custom StarDist model.

Data Augmentation

Data augmentation generates additional variation from existing training images by applying transformations to them. It is widely used in deep-learning image analysis to increase training-data diversity, reduce overfitting, and improve model generalization [5, 6].

Image augmentation methods can include both photometric and geometric transformations. Common examples include brightness and contrast adjustment, rotation, flipping, translation, zooming, and shearing [5, 6, 7].

During this project, augmentation approaches were considered in four categories: absent, basic, intermediate, and advanced augmentation.

Table 1: Data augmentation categories considered during model development.

Augmentation Category	Description
Absent Augmentation	No changes applied; the original images were retained as the baseline.
Basic Augmentation	Brightness and contrast adjustments.
Intermediate Augmentation	Rotation, flipping, and shifting transformations.
Advanced Augmentation	Zoom and shear transformations.

For this project, intermediate augmentation using a 90-degree rotation was selected. Rotation is a commonly used geometric augmentation technique that introduces orientation variation while preserving the essential structure of the image [5, 6].

Each selected training image was rotated by 90 degrees. Because this project involved instance segmentation, the corresponding annotation mask was transformed using the same rotation. Applying the same geometric transformation to both an image and its associated mask preserves the spatial correspondence between the input image and its ground-truth segmentation [7].

No flipping, shifting, zooming, shearing, brightness adjustment, or artificial noise was included in the selected augmentation procedure. Therefore, the final augmentation method introduced additional orientation variation without applying more extensive geometric or photometric changes to the oyster-seed images.

Evaluation Metrics and Test-Set Performance

The StarDist model was trained for 500 epochs using the prepared training images and corresponding annotations. Training and validation losses were monitored during model development. The 10 held-out test images were kept completely separate from model training and were used only for the final evaluation of StarDist and Cellpose. Both models were evaluated on the same test images to ensure a consistent comparison.

For instance-level evaluation, predicted oyster-seed objects were compared with the corresponding ground-truth annotations using Intersection over Union (IoU). IoU measures the overlap between a predicted region and its corresponding ground-truth region and is commonly used for evaluating object detection and segmentation performance [8]. An IoU threshold of 0.5 was used for the final comparison.

Based on this matching criterion, each prediction was classified as a true positive (TP), false positive (FP), or false negative (FN). A true positive represents a predicted oyster seed that correctly matches a ground-truth object. A false positive represents a predicted object without a correct ground-truth match, while a false negative represents a ground-truth oyster seed that was missed by the model.

Precision, recall, F1 score, and instance accuracy were then calculated to evaluate the detection performance of the two models. Precision and recall are commonly used measures for evaluating detection performance, while the F1 score provides a combined measure of precision and recall [9].

Precision measures the proportion of predicted oyster seeds that correctly match ground-truth objects:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

Recall measures the proportion of ground-truth oyster seeds that are successfully detected:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

The F1 score is the harmonic mean of precision and recall:

$$F1 = \frac{2(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}}.$$

Instance accuracy was calculated as:

$$\text{Instance Accuracy} = \frac{TP}{TP + FP + FN}.$$

These four metrics were calculated for both StarDist and Cellpose across the same 10 held-out test images at an IoU threshold of 0.5. The image-level performance of the two models is shown in Figure 6.

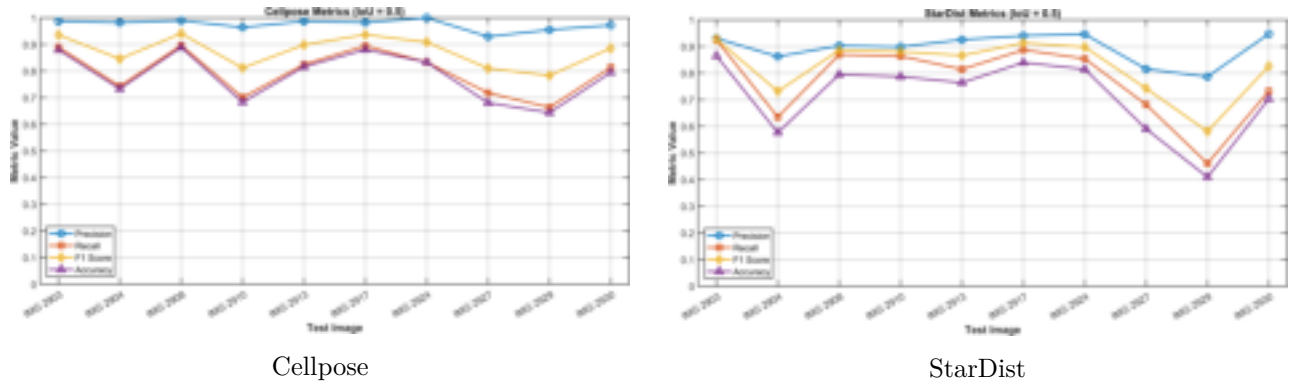


Figure 6: Precision, recall, F1 score, and instance accuracy for Cellpose (Left) and StarDist (Right) across the 10 held-out test images at IoU = 0.5.

Both models achieved strong performance across several test images, although their performance varied between individual cases. Cellpose showed stronger overall average performance across the four instance-level evaluation metrics, while StarDist achieved particularly strong results on some individual test images. The overall numerical comparison of the two models is presented in the Quantitative Results section.

Quantitative Results

Manual vs. Predicted Counts

In addition to instance-level evaluation, the total number of detected objects in each image was compared with the manually verified reference count.

Both models produced reasonable count estimates across the held-out test images, with Cellpose generally remaining closer to the manual reference counts.

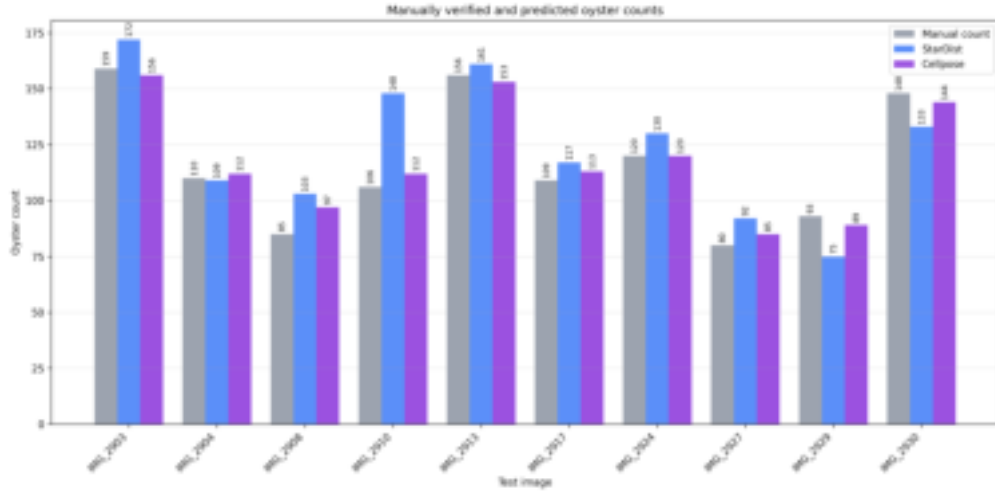


Figure 7: Manual reference counts compared with StarDist and Cellpose predicted counts across the held-out test images.

Absolute Count Error

Absolute count error measures the magnitude of the difference between a predicted count and the corresponding manually verified count:

$$\text{Absolute Count Error} = \left| C^{\text{predicted}} - C^{\text{manual}} \right|.$$

Lower absolute error indicates that the automated count is closer to the manual reference count. Across the evaluated test images, Cellpose produced lower absolute count errors on most images, while StarDist showed greater variation.

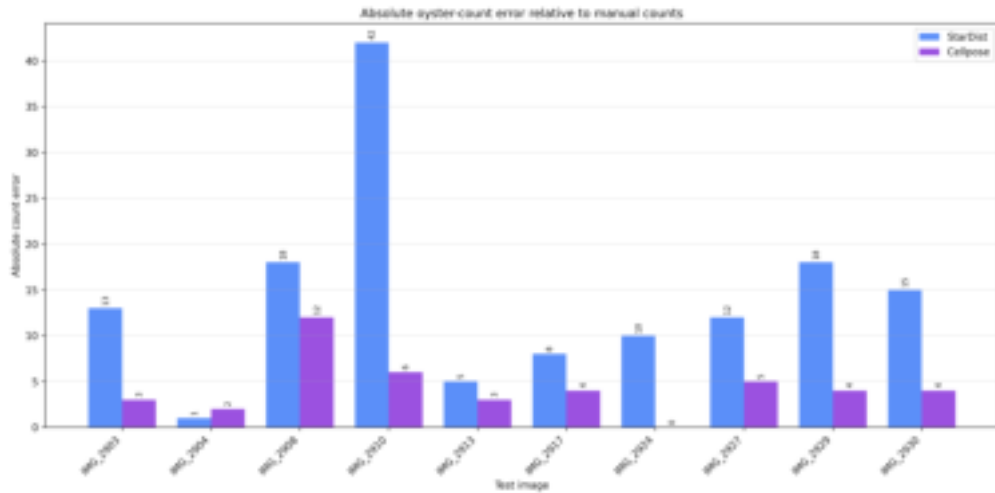


Figure 8: Absolute count error for StarDist and Cellpose across the held-out test images.

Overall Model Comparison

Mean Absolute Error (MAE) was used to summarize counting error across the full test set:

$$MAE = \frac{1}{N} \sum_{i=1}^N |C_i^{\text{predicted}} - C_i^{\text{manual}}|.$$

The overall performance of StarDist and Cellpose is summarized in Table 2.

Table 2: Overall performance of StarDist and Cellpose at IoU = 0.5.

Model	Precision	Recall	F1	Inst. Acc.	MAE
StarDist	0.895	0.772	0.825	0.715	14.2
Cellpose	0.975	0.798	0.876	0.782	4.3

Cellpose achieved higher overall precision, recall, F1 score, and instance accuracy than StarDist. It also produced a substantially lower mean absolute count error. Overall, Cellpose demonstrated stronger detection and counting performance on the evaluated 0–2 mm oyster-seed test set.

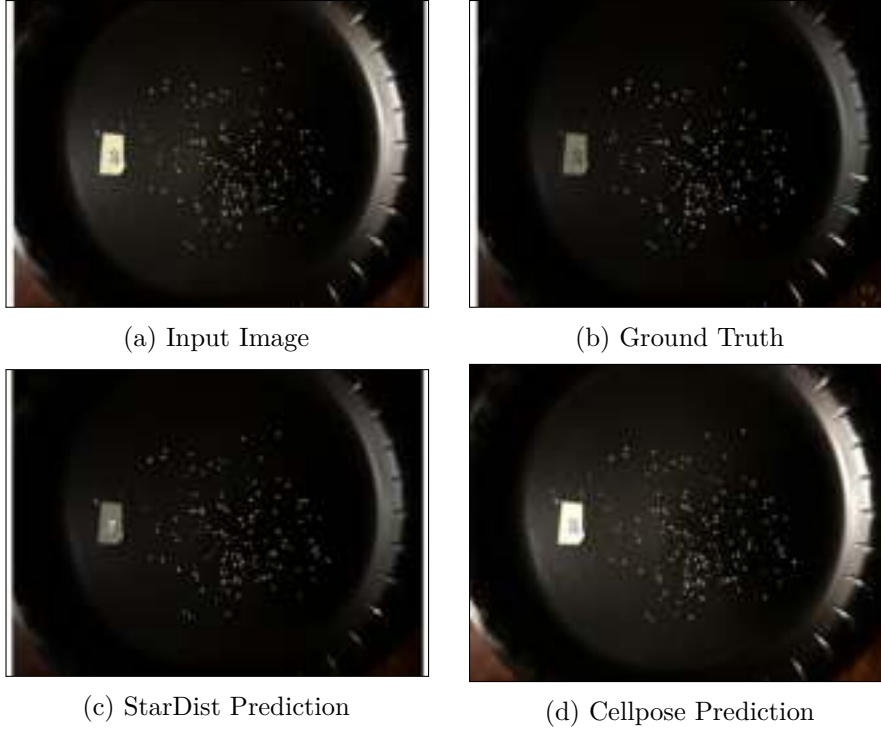


Figure 9: Qualitative comparison of the segmentation results: (a) input image, (b) ground-truth annotation, (c) StarDist prediction, and (d) Cellpose prediction.

Graphical User Interface

The existing oyster-counting graphical user interface was updated during Summer 2026 to support the StarDist and Cellpose 0–2 mm models.

The updated GUI automatically loads the available models and provides NVIDIA CUDA GPU support with CPU/GPU switching. A live progress bar was added during prediction, and prediction results can be exported directly to CSV.

A GPU-to-CPU cache-related issue encountered during development was also corrected.

At the time of the project presentation, the Cellpose model weights were not bundled directly with the GUI because the weight package was approximately 1.19 GB and exceeded the GitHub file-size limit. AMD GPU support had also not yet been implemented.

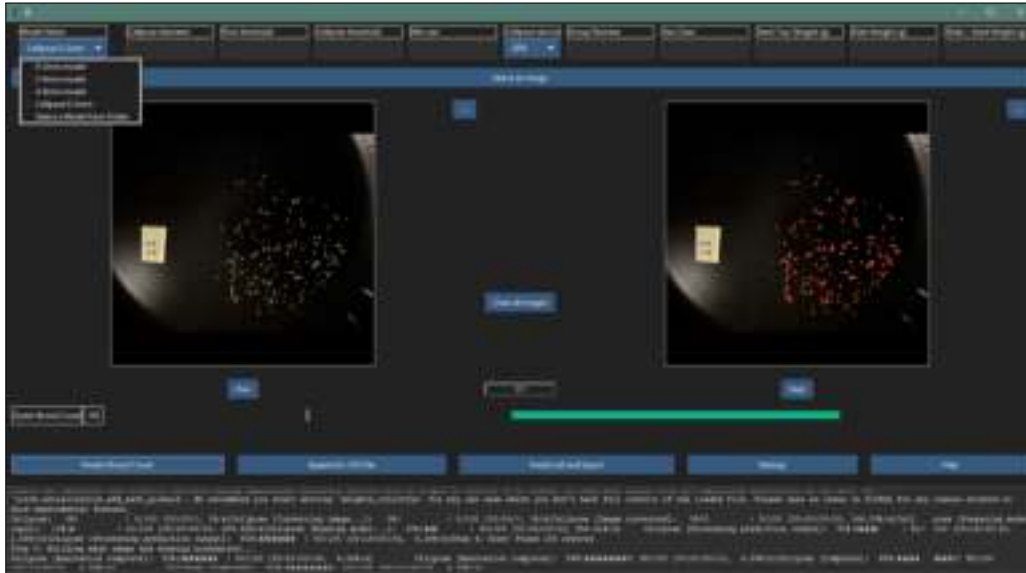


Figure 10: Updated graphical user interface for automated 0–2 mm oyster-seed detection and counting.

Practical Comparison

StarDist and Cellpose provide different practical advantages because of their different instance-segmentation representations.

StarDist uses star-convex polygons to represent individual objects [2]. It provides a relatively lightweight instance-segmentation approach and was practical for CPU-based prediction in this project.

Cellpose uses spatial flow fields rather than a fixed geometric object representation [3]. This representation provides greater flexibility for irregular or closely spaced objects.

For the evaluated 0–2 mm oyster-seed test set, Cellpose achieved stronger overall evaluation metrics and substantially lower counting error. StarDist, however, produced strong results on several individual test images.

From a deployment perspective, StarDist was more practical for CPU-only systems, whereas Cellpose benefited substantially from NVIDIA GPU acceleration. Therefore, model selection may depend on both the desired counting performance and the computational hardware available for deployment.

Future Work

- Increase the size and diversity of the 0–2 mm oyster-seed dataset.

- Evaluate model robustness under less-controlled imaging conditions.
- Test the updated GUI and models at the Grand Isle laboratory.
- Integrate the best-performing model with the Raspberry Pi setup.
- Continue improving CPU and AMD hardware support.

Acknowledgements

We would like to thank Dr. Peter Wolenski and Dr. Nadejda Drenska for their guidance and support throughout the project.

We also acknowledge Elizabeth M. Robinson, Dr. Sarah Bodenstein, the Louisiana Sea Grant Research Lab, and the Michael C. Voisin Oyster Hatchery for their support and domain guidance.

We thank Nikkos Svoboda for assistance with the “Chaos” high-performance computing system and the LSU Department of Mathematics for providing infrastructure and project support.

Oyster Team – Summer 2026

Faculty Advisors	Dr. Peter Wolenski Dr. Nadejda Drenska
Team Members	Gowri Priya Sunkara Sanjida Nupur Tasnim Tanha Mim Esrat Nur Asad Swati Tre’ Henson Briana Cox Emmanuel Asante

References

- [1] Louisiana Sea Grant, “Oyster Research Lab,” Louisiana State University.
- [2] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, “Cell Detection with Star-Convex Polygons,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pp. 265–273, 2018.
- [3] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, “Cellpose: a generalist algorithm for cellular segmentation,” *Nature Methods*, vol. 18, pp. 100–106, 2021.
- [4] M. Pachitariu, M. Rariden, and C. Stringer, “Cellpose-SAM: Superhuman Generalization for Cellular Segmentation,” 2025.
- [5] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, article 60, 2019.
- [6] L. Perez and J. Wang, “The Effectiveness of Data Augmentation in Image Classification using Deep Learning,” arXiv preprint arXiv:1712.04621, 2017.
- [7] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and Flexible Image Augmentations,” *Information*, vol. 11, no. 2, article 125, 2020.
- [8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.
- [9] D. M. W. Powers, “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
- [10] S. Bodenstein, D. Draper, M. C. Herpin, A. C. Louque, A. S. Wilson, S. Nupur, M. T. Shroyer, A. M. Johnson, G. P. Sunkara, S. Shalini, N. Drenska, P. R. Wolenski, and E. M. Robinson, “Applying Machine Learning Tools to Advance Quality Control in Oyster Nurseries,” *Aquacultural Engineering*, 2026, doi: 10.1016/j.aquaeng.2026.102772.