Using Computer Vision to Annotate and Count Dense 0-2mm Oyster Seeds

Emma Kate Conner

Maxwell Shroyer

Caleb Garza

MATH 4020:

Capstone Course

Dr. Nadejda Drenska

Dr. Peter Wolenski

Gowri Sunkara

Maganizo Kapita

December 1, 2025

Background

The Math Consultation Clinic (MC²) connects LSU mathematics students with real-world challenges by collaborating with businesses, government agencies, and research organizations. Through an integrated capstone research course, students leverage advanced mathematical modeling and computational tools to develop practical solutions for client problems. This hands-on experience not only strengthens students' technical and professional skills but also fosters innovation and collaboration. MC² supports the Department of Mathematics' educational mission, builds stronger partnerships between LSU and the Louisiana economy, and offers clients cost-effective access to high-level technical expertise.

This semester, and previous semesters, the DeVision Oyster team, a group specifically catered to fulfilling the needs of oyster seed farmers, partnered with the Louisiana Sea Grant. The Oysters team's main goal was to create an accurate counting system for oyster seeds utilizing StarDist and You Only Look Once (YOLO).

History and Goals

Over the past two years, the DeVision Oyster Team has partnered with the Louisiana Sea Grant Research Lab to develop an automated system for counting oyster seeds. Using a StarDist-based approach, we successfully built models for the 2–4 mm and 4–6 mm size ranges, achieving accuracy levels of 90–95%. However, our initial attempt at a 0–2 mm model reached only 75% accuracy, falling short of expectations. This semester, our primary goal was to refine and optimize the 0–2 mm model to deliver significantly improved performance and close the gap with our larger-size models.

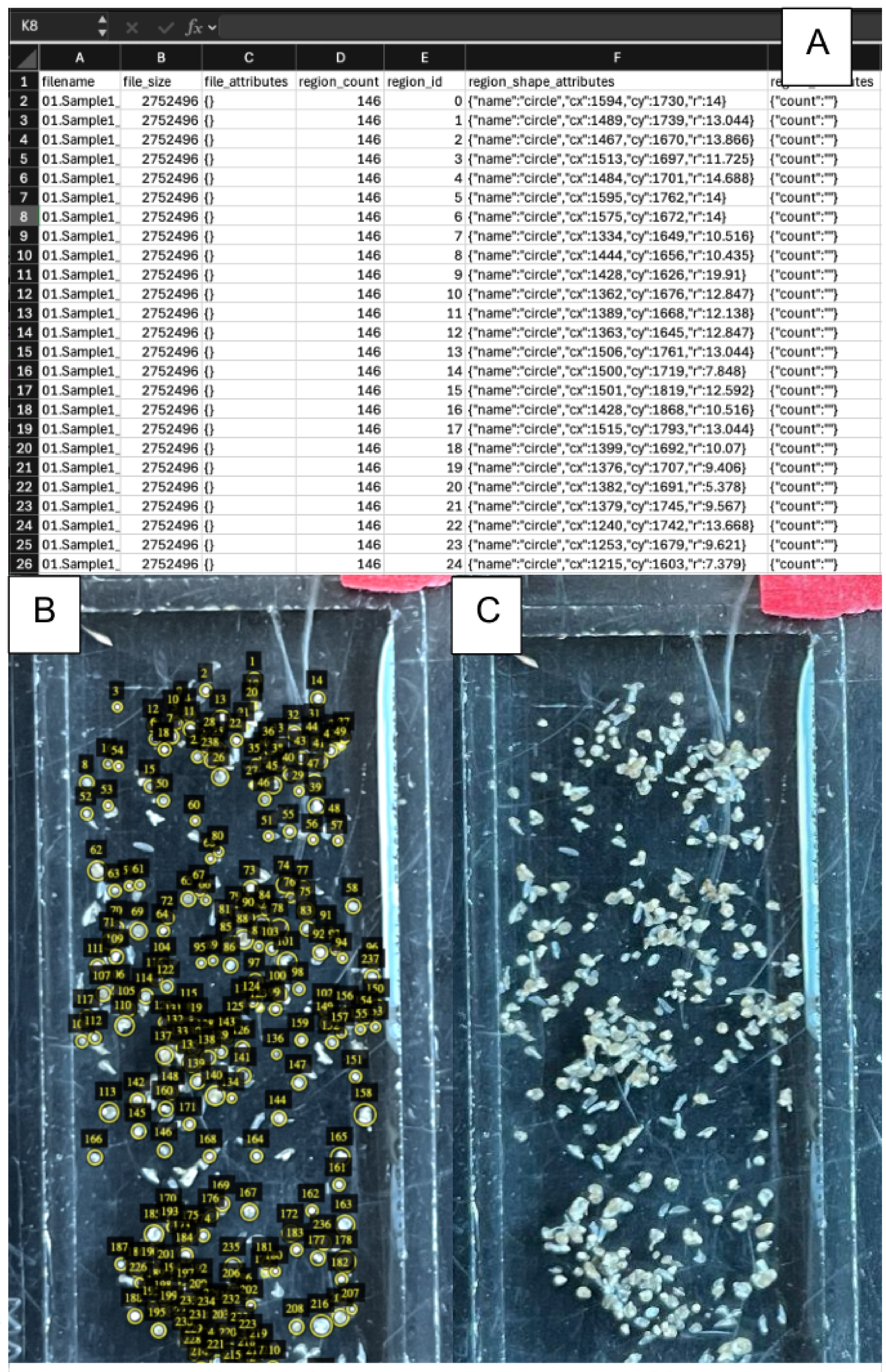
This semester, the primary goal of the project was to improve the accuracy of counting oyster seeds smaller than 2 mm from Petri dish images, a task that has proven challenging due to the seeds' small size and tendency to cluster. To accomplish this, we focused on refining our detection approach by comparing the performance of YOLO v10 (You Only Look Once) with StarDist-based models, aiming to identify the most effective solution for this size range. Additionally, we worked toward deploying the optimized model on a portable device, ensuring that oyster farmers, researchers, and other stakeholders can easily access and utilize this technology in real-world settings.

Annotation

The process of training the machine learning model begins with manual annotations. First, the oyster seed image in .jpg format is uploaded into the VGG Image Annotator (VIA), an open-source browser-based tool designed for manual image annotation. Each seed is then manually outlined using circular regions, with partially obscured seeds intentionally excluded from the annotation process to maintain consistency. For each annotated region, the attribute used is labeled as “count.” Consistency in annotation is critical, as inaccurate labeling can lead to errors during model training.

The annotation process presented several challenges. First, it was extremely time-consuming, as each seed had to be carefully outlined. Second, distinguishing individual seeds was sometimes difficult, especially when they were clustered or the image quality was poor. Third, maintaining uniform annotation practices across different team members required clear guidelines and frequent communication to minimize discrepancies.

Once all annotations were complete, the data was exported in a .csv file format. This file includes columns such as the region count, region ID, and shape. The “.csv” file generated for a labeled image (Figure 1A), along with the original “.jpeg” images (Figure 1C), serve as the input for training the StarDist machine learning model as well as the YOLO model. The annotations are drawn onto an empty single-channel image the same size as the original color image, with values corresponding to the field



	A	B	C	D	E	F	
1	filename	file_size	file_attributes	region_count	region_id	region_shape_attributes	re [illegible] tes
2	01.Sample1_	2752496	{}	146	0	{"name":"circle","cx":1594,"cy":1730,"r":14}	{"count":""}
3	01.Sample1_	2752496	{}	146	1	{"name":"circle","cx":1489,"cy":1739,"r":13.044}	{"count":""}
4	01.Sample1_	2752496	{}	146	2	{"name":"circle","cx":1467,"cy":1670,"r":13.866}	{"count":""}
5	01.Sample1_	2752496	{}	146	3	{"name":"circle","cx":1513,"cy":1697,"r":11.725}	{"count":""}
6	01.Sample1_	2752496	{}	146	4	{"name":"circle","cx":1484,"cy":1701,"r":14.688}	{"count":""}
7	01.Sample1_	2752496	{}	146	5	{"name":"circle","cx":1595,"cy":1762,"r":14}	{"count":""}
8	01.Sample1_	2752496	{}	146	6	{"name":"circle","cx":1575,"cy":1672,"r":14}	{"count":""}
9	01.Sample1_	2752496	{}	146	7	{"name":"circle","cx":1334,"cy":1649,"r":10.516}	{"count":""}
10	01.Sample1_	2752496	{}	146	8	{"name":"circle","cx":1444,"cy":1656,"r":10.435}	{"count":""}
11	01.Sample1_	2752496	{}	146	9	{"name":"circle","cx":1428,"cy":1626,"r":19.91}	{"count":""}
12	01.Sample1_	2752496	{}	146	10	{"name":"circle","cx":1362,"cy":1676,"r":12.847}	{"count":""}
13	01.Sample1_	2752496	{}	146	11	{"name":"circle","cx":1389,"cy":1668,"r":12.138}	{"count":""}
14	01.Sample1_	2752496	{}	146	12	{"name":"circle","cx":1363,"cy":1645,"r":12.847}	{"count":""}
15	01.Sample1_	2752496	{}	146	13	{"name":"circle","cx":1506,"cy":1761,"r":13.044}	{"count":""}
16	01.Sample1_	2752496	{}	146	14	{"name":"circle","cx":1500,"cy":1719,"r":7.848}	{"count":""}
17	01.Sample1_	2752496	{}	146	15	{"name":"circle","cx":1501,"cy":1819,"r":12.592}	{"count":""}
18	01.Sample1_	2752496	{}	146	16	{"name":"circle","cx":1428,"cy":1868,"r":10.516}	{"count":""}
19	01.Sample1_	2752496	{}	146	17	{"name":"circle","cx":1515,"cy":1793,"r":13.044}	{"count":""}
20	01.Sample1_	2752496	{}	146	18	{"name":"circle","cx":1399,"cy":1692,"r":10.07}	{"count":""}
21	01.Sample1_	2752496	{}	146	19	{"name":"circle","cx":1376,"cy":1707,"r":9.406}	{"count":""}
22	01.Sample1_	2752496	{}	146	20	{"name":"circle","cx":1382,"cy":1691,"r":5.378}	{"count":""}
23	01.Sample1_	2752496	{}	146	21	{"name":"circle","cx":1379,"cy":1745,"r":9.567}	{"count":""}
24	01.Sample1_	2752496	{}	146	22	{"name":"circle","cx":1240,"cy":1742,"r":13.668}	{"count":""}
25	01.Sample1_	2752496	{}	146	23	{"name":"circle","cx":1253,"cy":1679,"r":9.621}	{"count":""}
26	01.Sample1_	2752496	{}	146	24	{"name":"circle","cx":1215,"cy":1603,"r":7.379}	{"count":""}

Figure 1: Result of annotation A. ".csv" file. B. original ".jpeg" image. C. manually annotated image mask

“region_id” +1. This value is then mapped to the corresponding “count” attribute under the column “region_attributes.”

Deep Learning

Deep learning, a subset of machine learning, leverages neural networks with multiple layers to model complex patterns in data. These networks consist of an input layer, one or more hidden layers, and an output layer, enabling the system to learn hierarchical representations of image features. During training, the network iteratively adjusts the weights of its connections based on the error between predicted and actual outputs, improving accuracy over time.

StarDist for Segmentation and Counting

StarDist is a deep learning model specifically designed for object detection and segmentation, particularly for images containing star-convex shapes such as cells or small biological structures. It predicts object boundaries by estimating distances along 32 fixed radial directions, making it highly effective for clustered or irregularly shaped objects. In this project, we utilized the StarDist graphical user interface (GUI) to apply a pre-trained model to our manually annotated oyster seed images. The GUI streamlined the process of uploading data, adjusting prediction parameters, and visualizing results, while ensuring consistency in annotations to improve overall performance. The model was then evaluated based on its ability to accurately detect and count seeds in test images.

The Python environment supporting this workflow included over 150 packages optimized for deep learning and image analysis. Core libraries such as TensorFlow 2.18.0, Keras 3.8.0, and PyTorch 2.6.0+cu126 provided the foundation for model training, while OpenCV, scikit-image, and StarDist 0.9.1 handled image processing and segmentation. GPU acceleration was enabled through NVIDIA CUDA 12, ensuring efficient computation for large datasets. Development tools such as JupyterLab and visualization libraries like Matplotlib facilitated interactive experimentation and result analysis.

YOLOv10s for Real-Time Object Detection

To complement StarDist, we explored YOLOv10s (You Only Look Once), a state-of-the-art object detection algorithm known for its speed and efficiency. Unlike segmentation-based approaches, YOLO treats detection as a single regression problem, predicting bounding boxes and confidence scores directly from full images in one pass, as displayed in Figure 2. This architecture enables real-time performance, making YOLO particularly suitable for deployment on portable devices for oyster farmers and researchers.

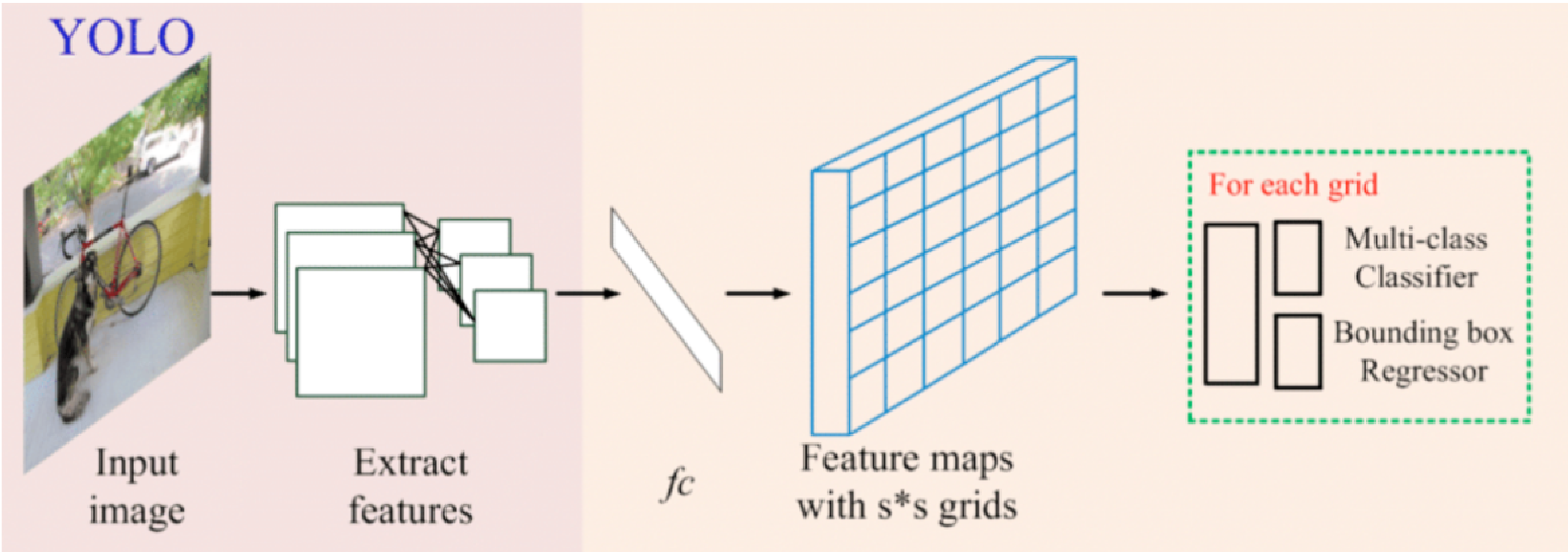


Figure 2: YOLOv10s Model Workflow

For this project, YOLO was fine-tuned on our annotated oyster seed dataset to detect seeds smaller than 2 mm. The model outputs bounding boxes for each detected seed, and the total count is derived by summing these detections per image. YOLO's ability to handle overlapping objects and maintain high accuracy under varying lighting and image conditions makes it a strong candidate for practical applications. Our evaluation compared YOLO's detection accuracy and speed against StarDist, aiming to identify the most effective solution for real-world deployment.

Results

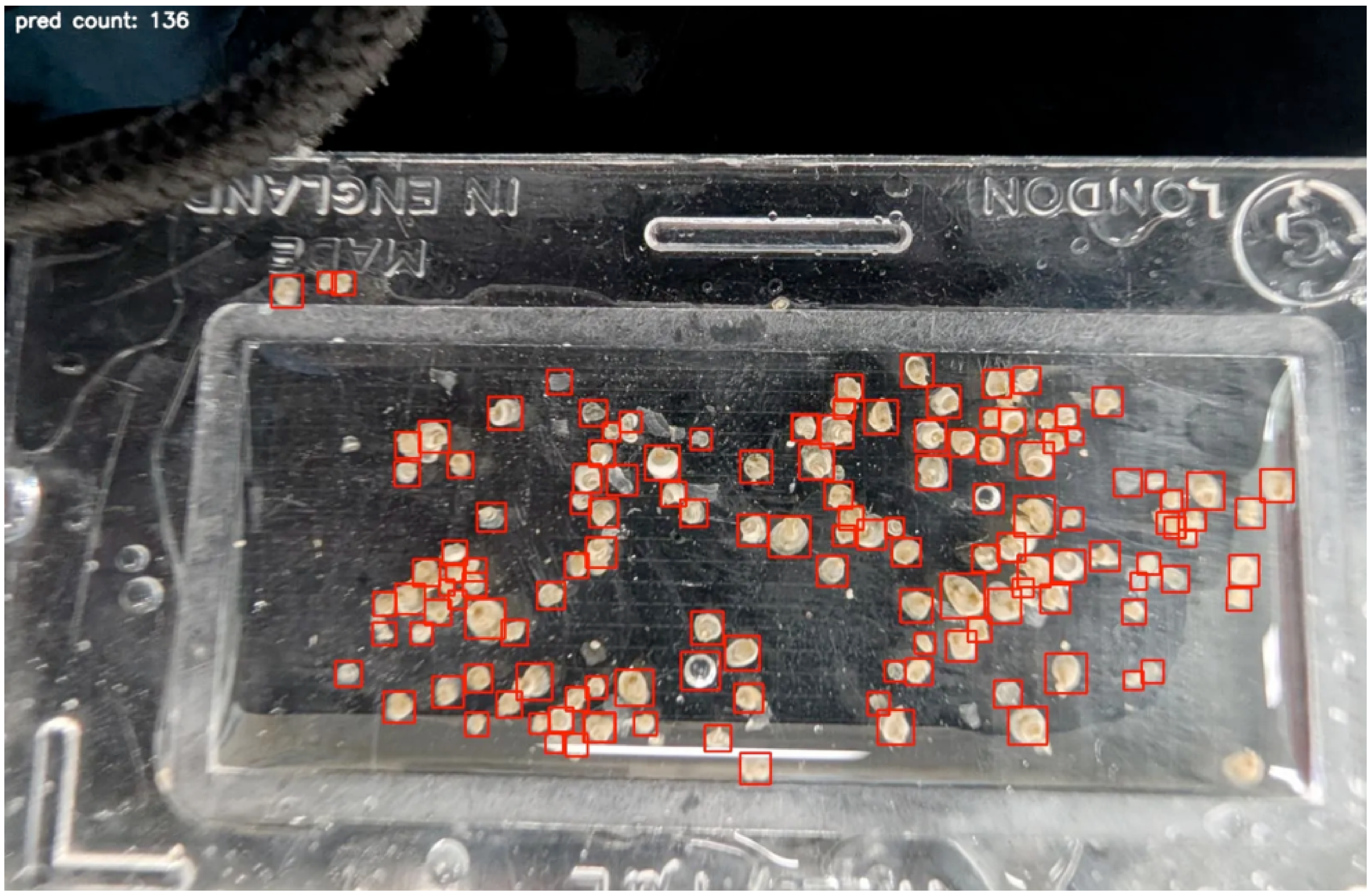


Figure 3: Fine-tuned YOLOv10s output (best performance)

Figure 4: Fine-tuned StarDist output for Comparison

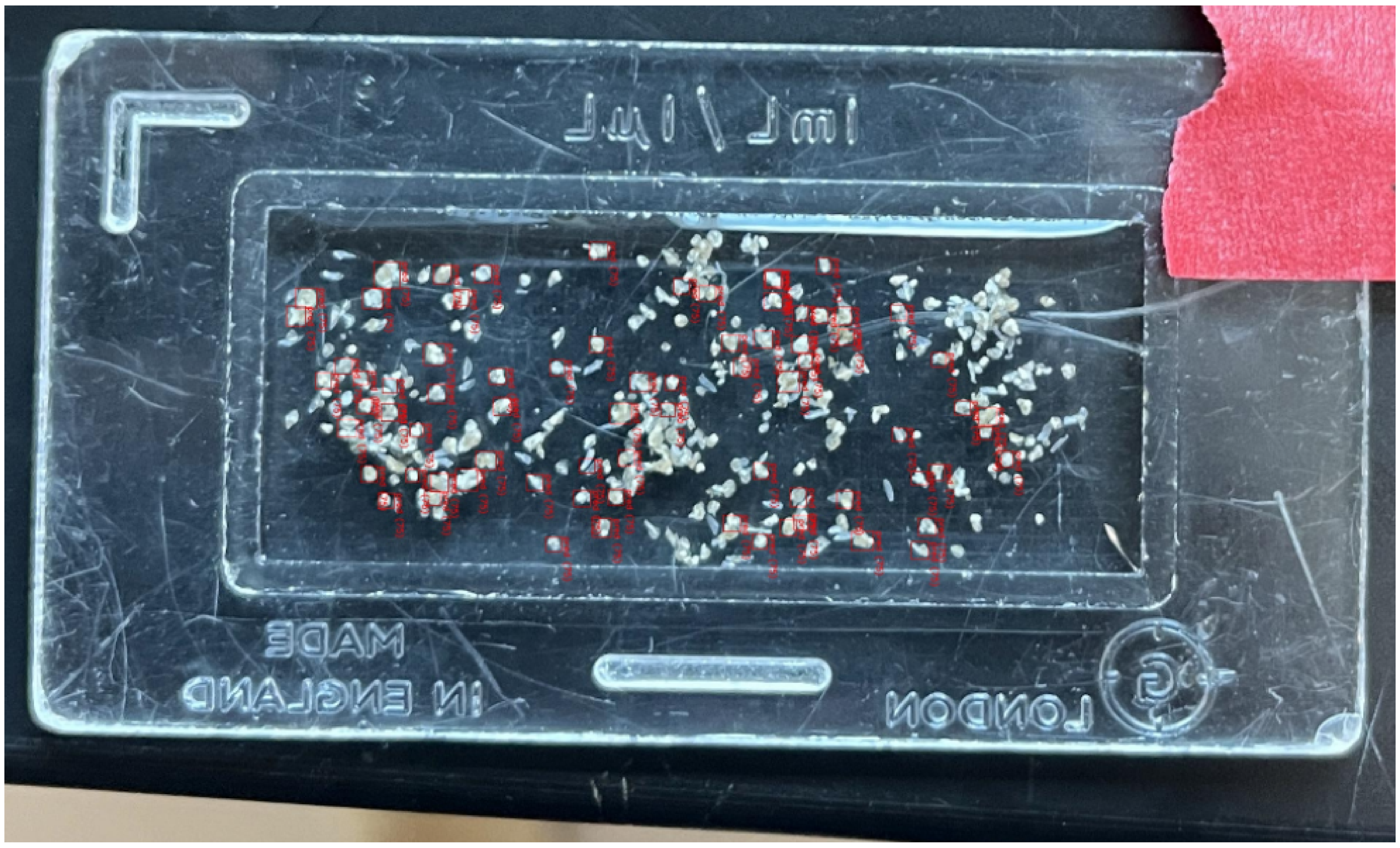


Figure 5: Fine-tuned YOLOv10s output (worst performance)

Table 1. Comparison of StarDist and YOLOv10s predictions on test images

Image	Actual Count	StarDist Pred.	YOLOv10s Pred.	StarDist Error	YOLOv10s Error
Sample19-16-25	118	62	117	-56	-1
Sample2_9-16-25	128	99	136	-29	+8
Sample3_9-16-25	161	128	184	-33	+23
Sample4_9-16-25	138	116	151	-22	+13
Sample5_9-16-25	186	153	214	-33	+28
Sample1_5-13-24	254	112	70	-142	-184
Sample2_5-13-24	275	252	179	-23	-96
Sample3_5-13-24	221	201	145	-20	-76
Sample4_5-13-24	249	235	187	-14	-62
MAE				**41.3**	**54.6**
Accuracy				**77.4%**	**76.4%**

Figure 6: A Table of the Comparison of StarDist and YOLOv10s Predictions on Test Images

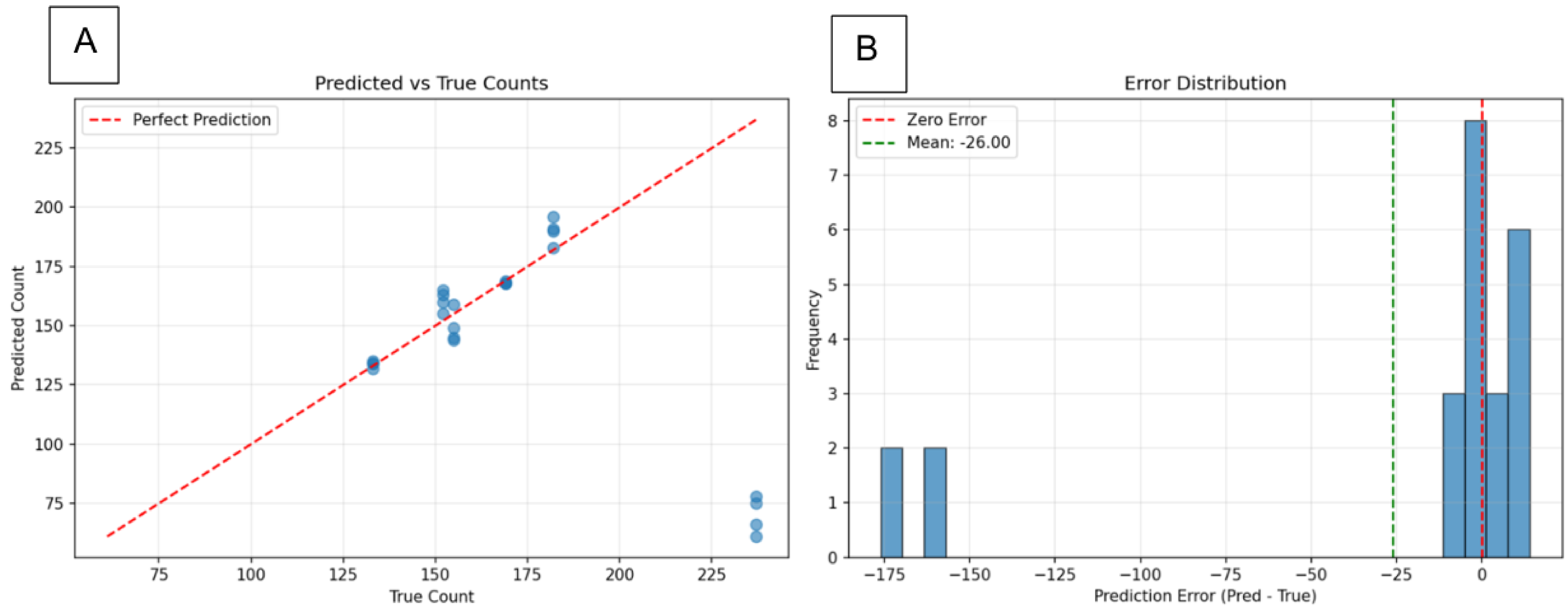


Figure 7: YOLOv10s Results. A. Predicted vs True Counts. B. Error Distribution

Discussion

The comparison between StarDist and YOLOv10s reveals distinct strengths and weaknesses for counting oyster seeds smaller than 2 mm. Figure 3 illustrates YOLO's best performance, where seeds in close-up, high-resolution images were detected accurately with minimal error. Conversely, Figure 5 shows YOLO's worst performance, which occurred in images where seeds were blurry or distant, leading to undercounting due to overlapping bounding boxes and reduced confidence scores.

StarDist, shown in Figure 4, performed better in scenarios where seeds were less distinct or partially blurred. Its segmentation-based approach, which predicts object boundaries along radial directions, allowed it to separate clustered seeds more effectively than YOLO in these cases. This advantage is further supported by the data in Figure 6, which compares predictions across multiple test images. As shown in Figure 6, StarDist achieved an overall accuracy of 77.4%, slightly outperforming YOLOv10s at 76.4%, despite YOLO's superior performance on close-up images. The Mean Absolute Error (MAE) values further highlight this trend, with StarDist at 41.3 compared to YOLO's 54.6, indicating that StarDist produced fewer large deviations from actual counts. StarDist consistently outperformed YOLO in images taken from farther distances or under poor lighting conditions, while YOLO excelled in close-up, well-focused images

Figure 7 provides additional insight into YOLO's performance, showing predicted versus true counts and error distribution. While YOLO demonstrated strong accuracy in ideal conditions, its variability across different image qualities suggests that further optimization and dataset expansion are necessary. Overall, these results indicate that StarDist is more robust for challenging image conditions, whereas YOLO offers superior speed and practicality for real-time applications when image quality is high.

Future Work

In the future, the DeVision Oyster team plans to continue efforts to deploy the YOLO model on a Raspberry Pi, enabling a portable solution for oyster farmers and researchers. This implementation would allow end users to perform real-time seed counting directly in the field without requiring high-end computing resources. Achieving this goal will involve optimizing the model for edge devices by reducing its computational footprint through techniques such as model pruning, quantization, and leveraging lightweight YOLO variants.

Additionally, performance improvements will be pursued through expanding and diversifying the training dataset. Collecting more images under varying conditions, such as different lighting, seed densities, and backgrounds, will help the model generalize better and maintain accuracy in real-world scenarios. Data augmentation strategies, including rotation, scaling, and contrast adjustments, will also be applied to enhance robustness.

Conclusion

This semester's work focused on improving automated counting of oyster seeds smaller than 2 mm, a critical challenge for Louisiana's oyster industry. By comparing StarDist and YOLOv10s, demonstrated that each model offers unique advantages: StarDist excels in handling blurry or distant images due to its segmentation-based approach, while YOLO performs best on close-up, high-resolution images and provides real-time detection capabilities. These findings suggest that YOLO is the most practical solution for deployment on portable devices, whereas StarDist remains valuable for high-precision tasks in controlled environments. Future improvements, including dataset expansion, model optimization, and edge-device integration, will further enhance accuracy and usability. Ultimately, this project represents a significant step toward delivering a scalable, efficient, and accessible tool for oyster farmers and researchers, supporting sustainable aquaculture practices in Louisiana and beyond.

References

A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," in Advances in Neural Information Processing Systems (NeurIPS), 2024. arXiv:2405.14458.

G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015. NIPS 2014 Deep Learning Workshop.

"Louisiana Sea Grant." (n.d.). Retrieved October 21, 2025, from https://www.laseagrant.org/

O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, vol. 9351 of Lecture Notes in Computer Science, pp. 234–241, 2015.

Weigert, M., Schmidt, U., Haase, R., Sugawara, k., Myers, G. (2020). Star-convex polyhedra for 3D object detection and segmentation in microscopy- IEEE (WACV) 2020.

Weigert, M., Schmidt, U. (2022). Nuclei Instance Segmentation and Classification in Histopathology Images with Stardist - IEEE (ISBIC) 2022.