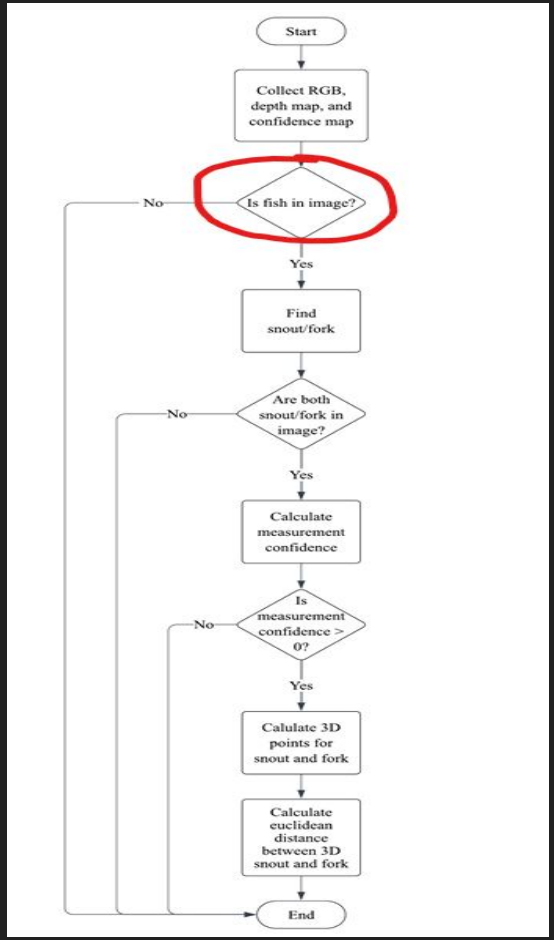
ENGINEERS FOR EXPLORATION

Active Learning for Underwater Fish Detection: Evaluating Uncertainty Sampling Strategies in Data-Scarce Environments

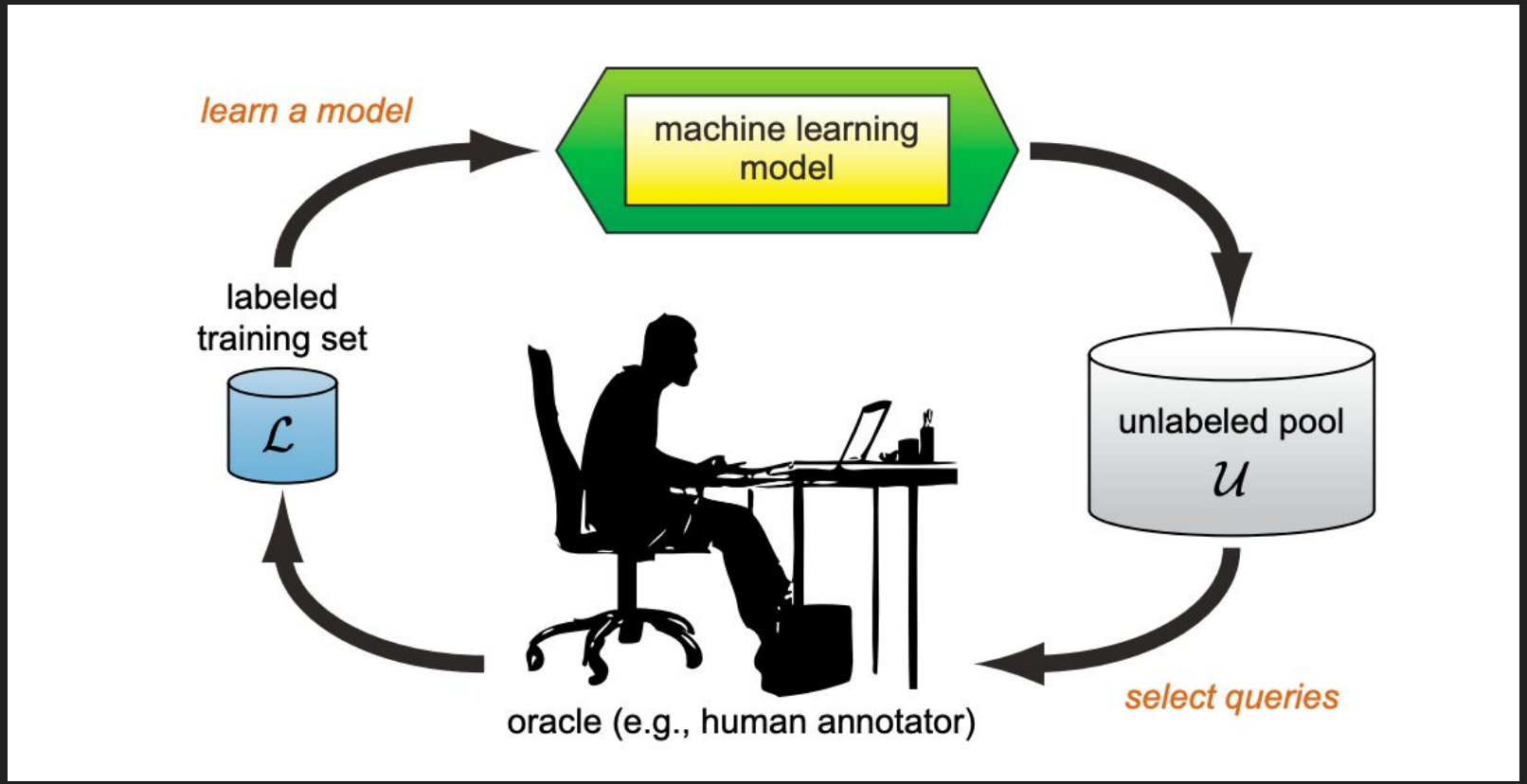
Akhil Nallacheruvu

Problem Statement

- FishSense app outputs head-to-tail measurements of fish, which requires an underwater vision model that can detect fish (current process is human labeling → can be automated by object detection model)
- Difficulty - Each domain (habitat/camera/image quality) requires the model to be trained on new data from that specific domain
- Need a way to train a model to infer on new domains with minimal amounts of data from that domain so that less time is spent on training

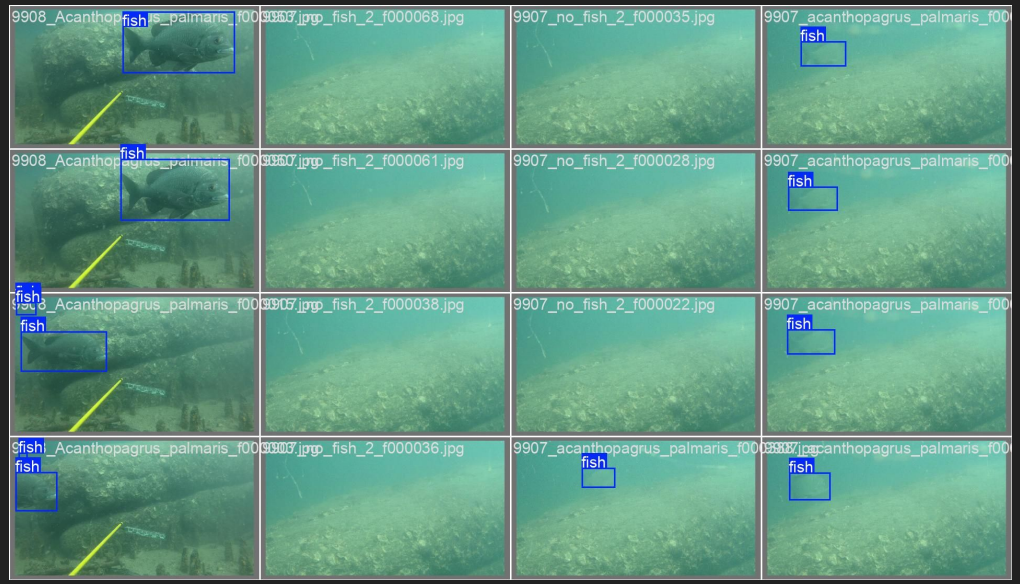

Start
Collect RGB, depth map, and confidence map
Is fish in image?
No
Yes
Find snout/fork
Are both snout/fork in image?
No
Yes
Calculate measurement confidence
Is measurement confidence > 0?
No
Yes
Calulate 3D points for snout and fork
Calculate euclidean distance between 3D snout and fork
End

Solution - Active Learning



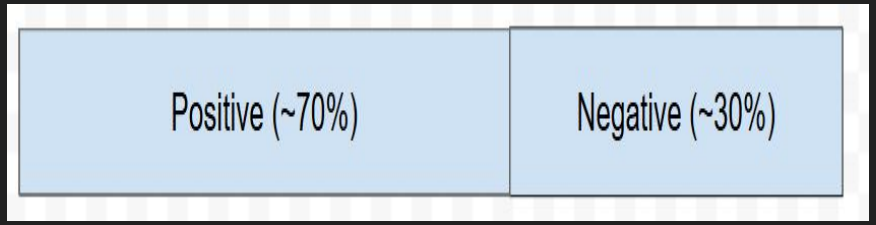
Implementation

- In this case, we must run supervised ML on a labeled dataset (increasing in size) → may require multiple iterations of training to reach desired metrics on evaluation set
- Research - Best way to select data for labeling



Dataset

- DeepFish dataset
- Positive-Negative Split:
  - Negative - 2012 images (~30%)
  - Positive - 4505 images (~70%)
- Train-Val-Test Split:
  - Train - 4546 images (~70%)
  - Val - 1288 images (~20%)
  - Test - 683 images (~10%)



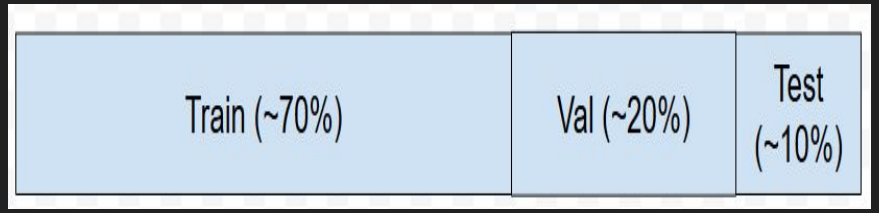
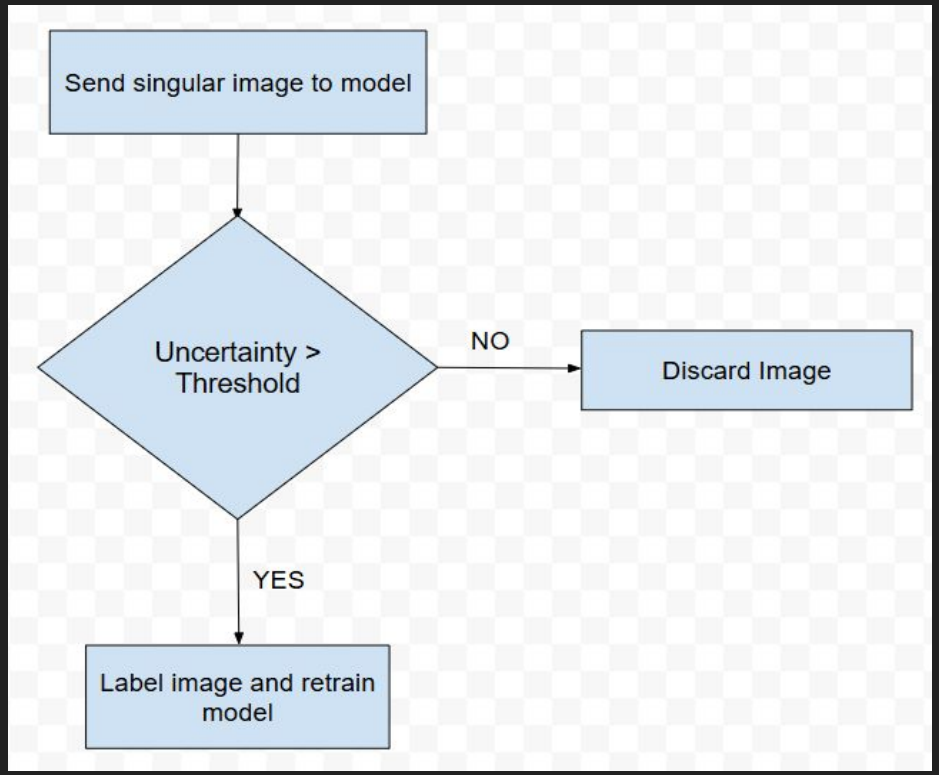


Image from DeepFish dataset

Strategies - Overview

- **Sampling Strategies:**
  - Stream-based
  - Pool-based
- Query Selection Strategies:
  - Uncertainty Sampling
  - Diversity Sampling

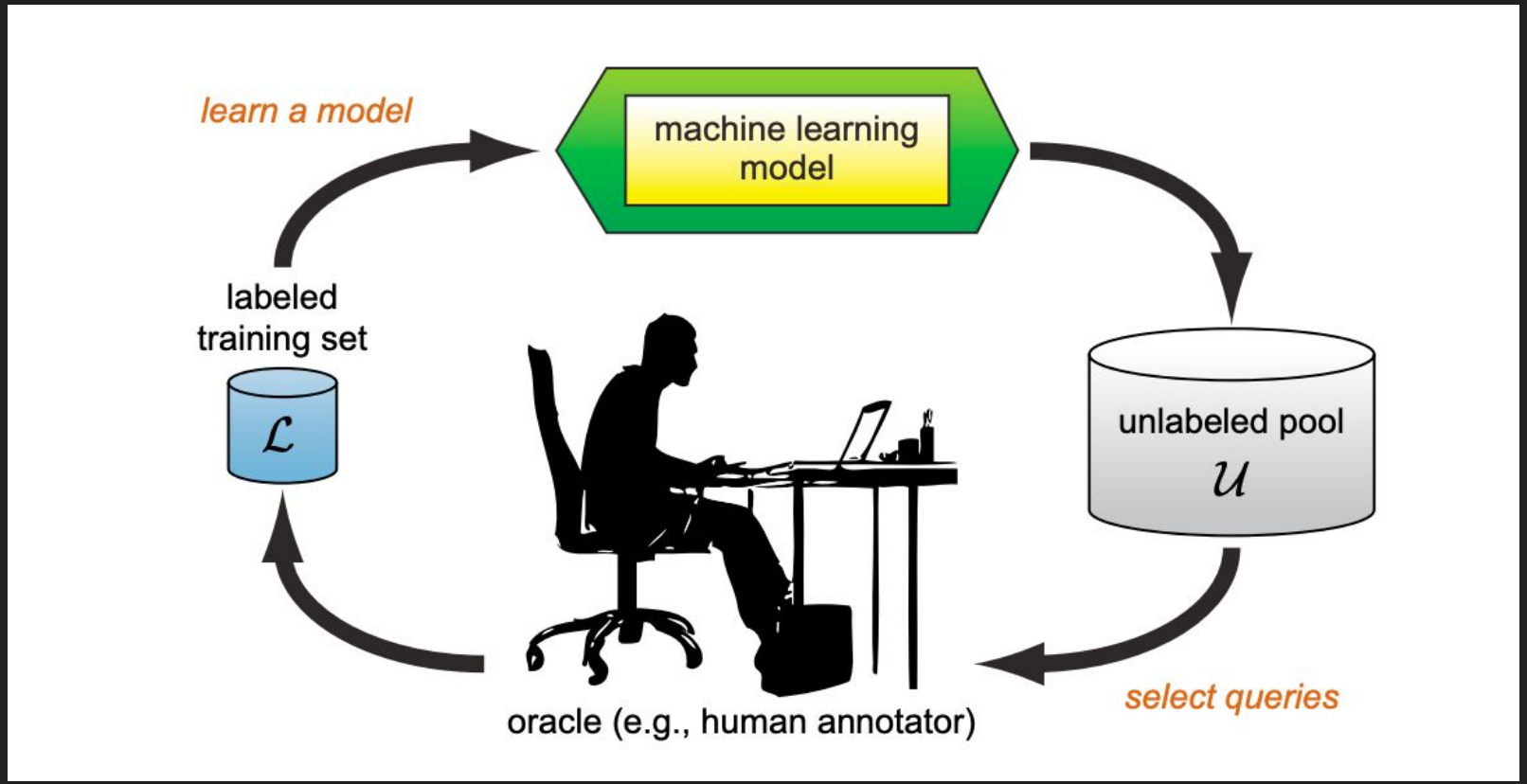
Stream-based Sampling - Process



Stream-based Sampling - Advantages & Disadvantages

- Advantages:
  - Fast inference when data is received by oracle for labeling in real-time
  - Doesn’t require a huge dataset
- Disadvantages:
  - Once an image is skipped by oracle, it won’t come back in inference
  - FishSense requires detection to work in real-time - not labeling

Pool-based Sampling - Process

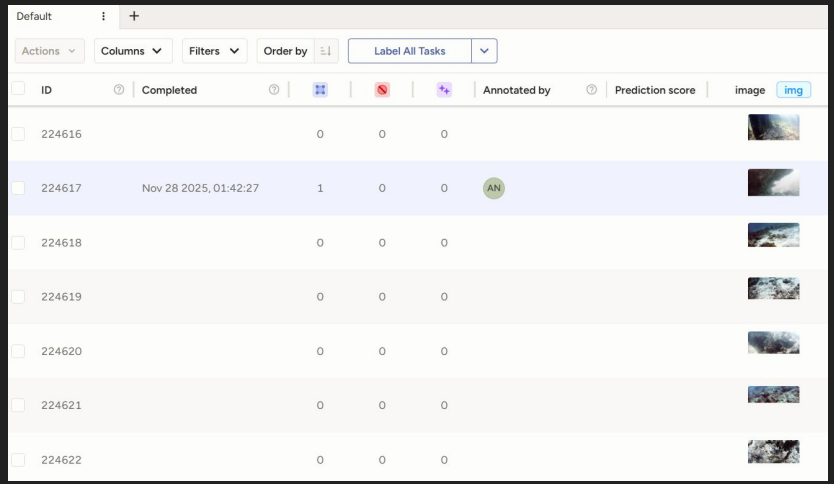


Pool-based Sampling - Advantages & Disadvantages

- Advantages:
  - Able to pick the most informative samples from a large set of unlabeled data
  - Can still access data in the next iteration of training that wasn't considered informative in the previous iteration of training (no discarding)
- Disadvantages:
  - Inference performed on ALL unlabeled instances (more unlabeled data → more expensive to label → more time spent on training/evaluation process)

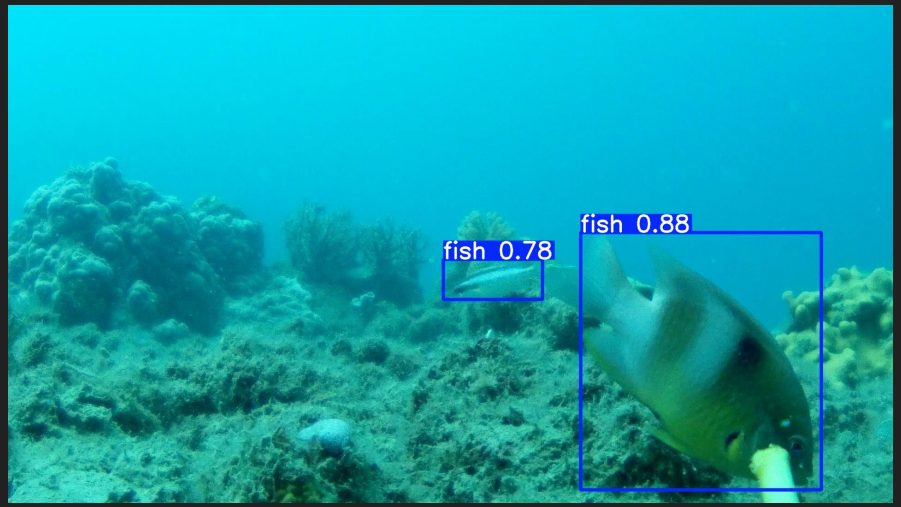
Decision: Pool-based Sampling

- FishSense requires detection to work in real-time not labeling
- Easier to integrate onto LabelStudio



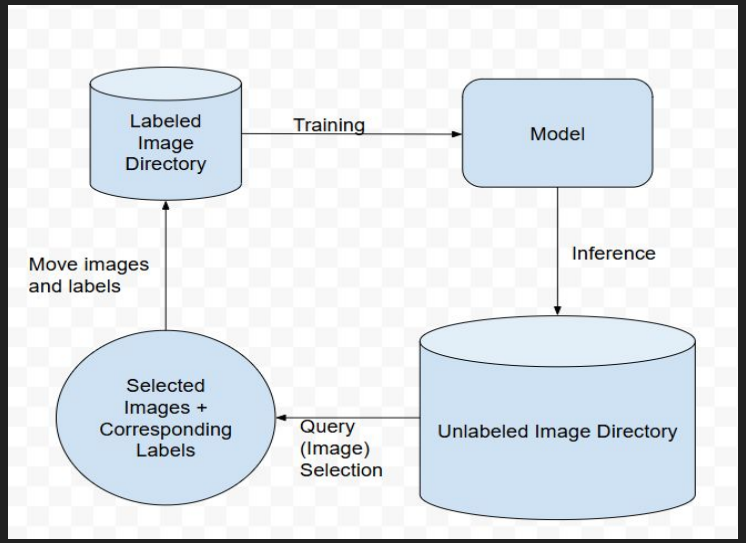
Pool-based Sampling - Implementation

- Initial training done on 5% of training data (deterministic shuffling using Pandas → initial training is done on the same set of samples in each experiment)
  - Done by data setup class in AL_toolkit
- Model performs inference on images in unlabeled pool (predicts bounding boxes around fish and outputs confidence scores of predictions) and all query selection strategies are based on these scores



Pool-based Sampling - Implementation

- Once queries are selected, then oracle labeling is simulated
- Experiment: Run the process 3 times (total of 4 training iterations) and collect data from inference on the test



Pool-based Sampling Process Flow

Strategies - Overview

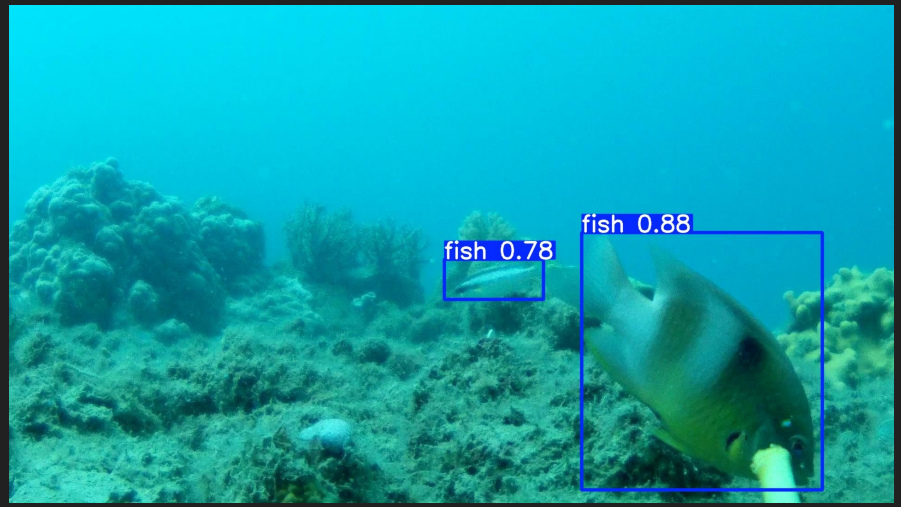
- Sampling Strategies:
  - Stream-based
  - Pool-based
- **Query Selection Strategies:**
  - Uncertainty Sampling
  - Diversity Sampling

Diversity Sampling

- Set of strategies for identifying unlabeled items that are unknown/underrepresented by model in current state
- Goal: target instances for labeling that maximize the coverage of the data that the model is trained on
- Pool-based Sampling Object Detection Example:
  - Embedding images using feature extractor
  - Running k-means clustering on embeddings
  - Select samples closest to the center of each cluster

Uncertainty Sampling

- Set of strategies for identifying unlabeled items that are near decision boundary of model (instances model is unsure of)
- Instances selected that model is most likely to misclassify
- Strongly based on confidence scores in object detection



Decision - Uncertainty Sampling

- AL typically uses a mix of uncertainty and diversity sampling and this project does the same
- Explicitly constructed uncertainty sampler and outsourced diversity sampling to iteration cycles due to the small amount of differences (same object + small number of domain shift differences that don’t justify additional computation)
- Lots to explore for performance of model trained, tested, and evaluated on this dataset on different uncertainty sampling methods
- Will need to explicitly build a diversity sampler as dataset becomes more diverse

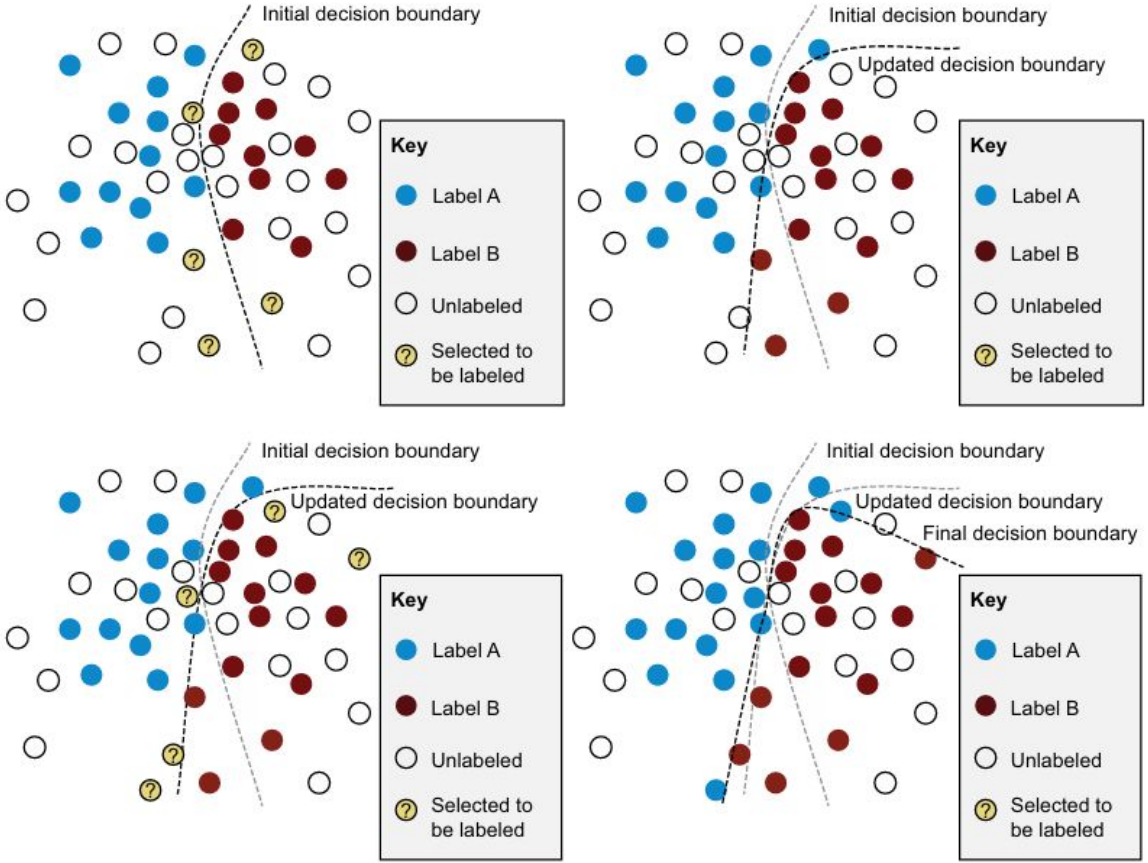


Figure 1.3 The iterative active learning process. *Top left to bottom right:* Two iterations of active learning. In each iteration, items are selected along a diverse selection of the boundary, which in turn causes the boundary to move after retraining, resulting in a more accurate machine learning model. Ideally, we requested human labels for the minimum number of items as part of our active learning strategy. This request speeds the time to get an accurate model and reduces the overall cost of human annotation.

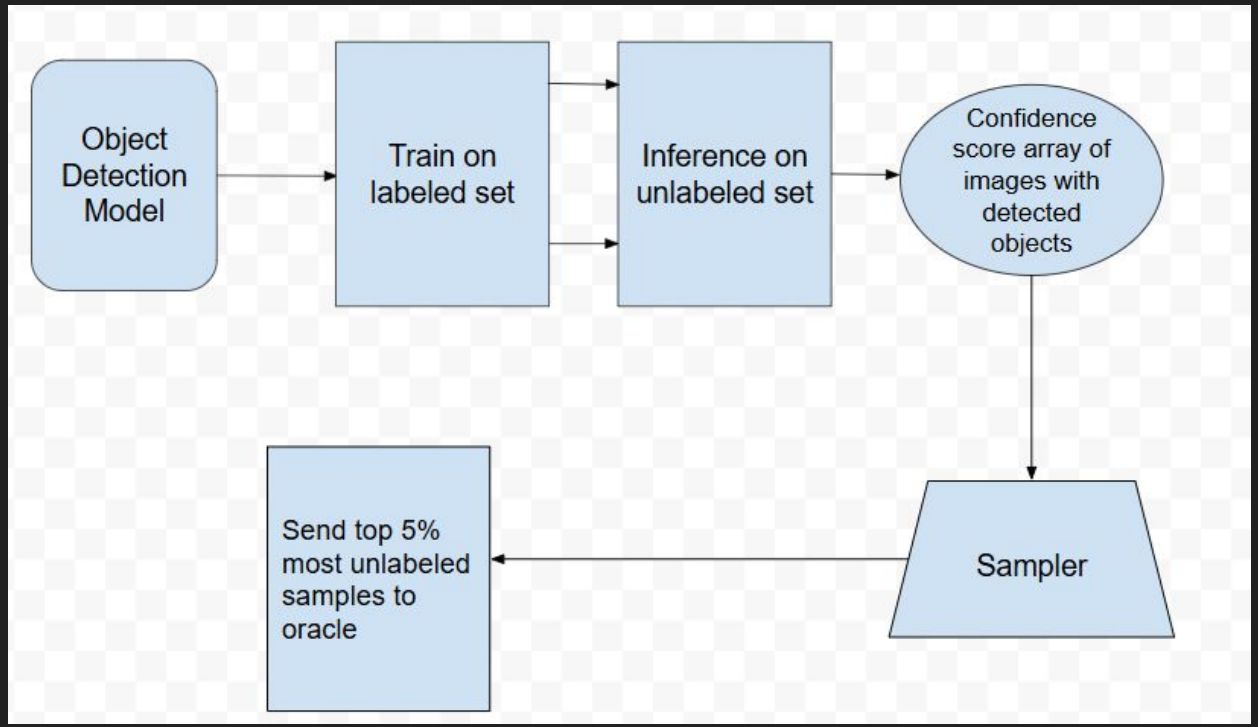
Uncertainty Sampling - Implementation

- Relies heavily on confidence scores generated by model during inference (overall image confidence score is selected by sampling the confidence scores of all detected objects in an image)
- When a model doesn't detect an object, it automatically sets the score to 0

$$\Pr(\text{Class}_i \mid \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Confidence Score Formula

Model Training/Query Selection Process Flow



Detection Model Selection - YOLOv8n

- Faster training
- Less space
- Easier to deploy to application / backend (extensive community support when compared to other YOLO models)
- Model Handler (YOLOModelHandler): AL_toolkit/src/models/model_handler.py at main · UCSD-E4E/AL_toolkit

Baseline - Random Sampling

- Sampler selects images in random order to send to oracle
- Baseline for all experiments
- Any sampling strategy that will be selected for active learning on FishSense must beat the performance of random sampling

Uncertainty Sampling Strategies - Overview

- Least-Confidence Sampling
- Entropy Sampling
- Code:
  - (UncertaintySampler): AL_toolkit/src/al_toolkit/uncertainty_sampling.py at main · UCSD-E4E/AL_toolkit
  - (Oracle Simulation): AL_toolkit/src/dataset_tools/move_data.py at main · UCSD-E4E/AL_toolkit

Least-Confidence Sampling

- Sampler selects images with lowest overall confidence scores
- Advantage: simpler to implement, scalable to large datasets
- Strategies for acquiring image confidence scores:
  - Mean Confidence Score Aggregation
  - Minimum Confidence Score Aggregation

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Mean Confidence Score Aggregation

Let N denote the number of objects detected in an image by a given model. For each detected object $i \in \{1, \ldots, N\}$, let $c_i \in [0, 1]$ be the model's confidence score associated with that object. The image-level confidence score, denoted C_{img}, is defined as the arithmetic mean of the individual object confidence scores:

$$C_{\text{img}} = \frac{1}{N} \sum_{i=1}^{N} c_i.$$

This quantity provides a single scalar measure summarizing the model's overall confidence across all detections in the image.

Minimum Confidence Score Aggregation

Let N denote the number of objects detected in an image by a given model. For each detected object $i \in \{1, \ldots, N\}$, let $c_i \in [0, 1]$ be the model's confidence score associated with that object. The image-level confidence score, denoted C_{img}, is defined as the minimum of the individual object confidence scores:

$$C_{\text{img}} = \min_{1 \leq i \leq N} c_i.$$

Entropy-based Sampling

- Selects images with the highest overall entropy scores
- Advantage: can give a better representation of uncertainty across the image than simply confidence scores
- Confidence score array turned into entropy array
- Strategies for acquiring image entropy:
  - Mean Entropy Aggregation
  - Maximum Entropy Aggregation

Let $c_i \in [0, 1]$ denote the confidence score assigned by the model to the i-th detected object, interpreted as the probability that the detection is correct. The entropy associated with this detection, denoted $H(c_i)$, is defined using the Bernoulli entropy:

$$H(c_i) = -\,c_i \log(c_i) \;-\; (1 - c_i) \log(1 - c_i).$$

Mean Entropy Aggregation

Let N denote the number of objects detected in an image by a given model. For each detected object $i \in \{1, \ldots, N\}$, let $c_i \in [0, 1]$ be the model's confidence score, and let the corresponding entropy be

$$H(c_i) = -c_i \log(c_i) - (1 - c_i) \log(1 - c_i).$$

The image-level entropy score obtained via mean entropy pooling, denoted H_{img}, is defined as

$$H_{\text{img}} = \frac{1}{N} \sum_{i=1}^{N} H(c_i).$$

This quantity represents the average uncertainty across all detected objects in the image.

Maximum Entropy Aggregation

Let N denote the number of objects detected in an image by a given model. For each detected object $i \in \{1, \ldots, N\}$, let $c_i \in [0, 1]$ be the model's confidence score, and let the corresponding entropy be

$$H(c_i) = -c_i \log(c_i) - (1 - c_i) \log(1 - c_i).$$

The image-level entropy score obtained via maximum entropy pooling, denoted H_{img}, is defined as

$$H_{\text{img}} = \max_{1 \le i \le N} H(c_i).$$

This pooling strategy assigns the image-level score according to the most uncertain detected object in the image.

Model Performance - YOLOv8n

- Performance of model measured by mAP50-95 on test set after 4 iterations of training (50 epochs / iteration with 5% of total training instances added before each training iteration)

Method	Seed 1 mAP50-95	Seed 2 mAP 50-95	Seed 3 mAP50-95	Seed 4 mAP50-95	Seed 5 mAP50-95	Mean	Standard Deviation
Random Sampling	0.565	0.559	0.561	0.568	0.568	0.5642	0.004086563348
Least Confidence (Mean Pooling)	0.51	0.53	0.539	0.514	0.515	0.5216	0.01234098862
Least Confidence (Min Pooling)	0.575	0.568	0.586	0.573	0.591	0.5786	0.009555103348
Entropy (Mean Pooling)	0.566	0.563	0.549	0.566	0.571	0.563	0.008336666
Entropy (Max Pooling)	0.583	0.575	0.574	0.582	0.577	0.5782	0.004086563348

T-Test Results

- Least Confidence Sampling with Mean Confidence Score Pooling performs worse than Random Sampling with statistical significance ($p < 0.05$)
- Entropy Sampling with Mean Entropy Score Pooling performs about the same as Random Sampling ($p >= 0.05$)
- Entropy Sampling with Maximum Entropy Score Pooling and Least Confidence Sampling with Minimum Confidence Score Pooling performs better than random sampling with statistical significance ($p < 0.05$)
- No statistical significance between performance of Least Confidence Sampling with Minimum Confidence Score pooling and Entropy Sampling with Maximum Entropy Score pooling ($p >= 0.05$)

Next Steps

- Model deployed onto LabelStudio with predictions enabled; will expand to training (uncertainty sampling only available for Enterprise, will develop workaround in backend)
- Will deploy model with both LC-Min and Ent-Max enabled and design experiment for determining which sampler performs better during deployment (LabelStudio or Application)

References

[1] C. L. Crutchfield, N. Rekasius, A. Jajodia, T. Darci-Maher, A. Chen, A. Budhathoki, J. Tallman, C. Wilson, J. Elstner, B. Mwamba, S. Shrestha, Z. Merson, S. Doyle, G. Turner, B. X. Semmens, C. Schurgers, and R. Kastner, *FishSense Mobile: A Mobile Device App for On-Deck Fisheries Management Operations*. [Online]. Available: oceans2025-fishsenseMobile.pdf

[2] M. Hao, H. Yu, and D. Li, "The Measurement of Fish Size by Machine Vision – A Review," in *Computer and Computing Technologies in Agriculture IX (CCTA 2015)*, D. Li and Z. Li, Eds., vol. 479, Cham, Switzerland: Springer, 2016, pp. 17–28. doi: 10.1007/978-3-319-48354-2_2.

[3]A. K., V. P. S., and T. Kumaki, "A Survey on Fish Size Measurement in Aquatic Environments," *2024 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, Karkala, India, 2024, pp. 23–27, doi: 10.1109/B-HTC60740.2024.10562392.

[4]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

References

[5]T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” *arXiv preprint arXiv:1405.0312*, 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–308, Jun. 2010. [Online]. Available: The Pascal Visual Object Classes (VOC) Challenge - Microsoft Research

[7] B. Settles, *Active Learning Literature Survey*, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, Jan. 26, 2010. [Online]. Available: Active Learning Literature Survey

[8] R. Munro, *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI*. Shelter Island, NY: Manning, 2021.

[9] C. Yang, L. Huang, and E. J. Crowley, “Plug and Play Active Learning for Object Detection,” *arXiv preprint arXiv:2211.11612*, 2024. [Online]. Available: https://arxiv.org/abs/2211.11612

References

[10] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, “A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis,” *Sci. Rep.*, vol. 10, no. 1, Art. no. 14671, 2020, doi: 10.1038/s41598-020-71639-x.

[11] *Deep Fish Object Detection*, [Online]. Available: Deep Fish Object Detection

[12] L. Alashrafi, R. Badawood, H. Almagrabi, M. Alrige, F. Alharbi, and O. Almatrafi, “Benchmarking Lightweight YOLO Object Detectors for Real-Time Hygiene Compliance Monitoring,” *Sensors*, vol. 25, no. 6140, 2025, doi: 10.3390/s25196140.

[13] *YOLOv8 vs YOLOv9: A Technical Comparison for Object Detection*, [Online]. Available: YOLOv8 vs YOLOv9: A Technical Comparison for Object Detection

[14] *detect - Detect objects using YOLO v4 object detector - MATLAB*, [Online]. Available: detect - Detect objects using YOLO v4 object detector - MATLAB