VARS-ML AutoLabel: A VARS-Integrated Vector Embedding Classification Pipeline with Hierarchical Taxonomic Evaluation

Abstract

The Monterey Bay Aquarium Research Institute (MBARI) has amassed a 40-year, 30,000 hour archive of annotated deep-sea video, containing approximately 13 million annotations and 1.5 million localizations. We present a classification pipeline that uses vector embeddings from three open-source image encoders (DINOv3, BioCLIP2, and SigLIP2) to automatically label regions of interest (ROIs) drawn from this archive. Each localization within VARS is embedded and stored in a vector database optimized for similarity search, and new points are classified via a distance-weighted k-nearest neighbors (k-NN, k=5) search against this database, with predictions from all three models combined through an agreement-based ensemble scheme. Because taxonomic labels are hierarchical, we additionally introduce a rank-aware scoring system that credits predictions partially when they are close to, but not an exact match for, the true label, rather than scoring every non-exact prediction as a complete miss. On a representative evaluation run of 47,126 marine organism points, the pipeline achieved a pipeline accuracy of 63.89%, a precision of 73.61% on points it was confident enough to auto-label, and a coverage rate of 86.8%. These results indicate that the primary value of this system is not full automation, but targeted triage: the pipeline reliably identifies which ROIs can be trusted without review and which genuinely require a human annotator's attention, allowing MBARI's Video Lab team to prioritize their limited review time. We further discuss additional capabilities of the underlying vector database — including outlier detection, natural-language querying, and visual similarity search — that extend beyond classification alone.

Introduction

The Monterey Bay Aquarium Research Institute (MBARI) maintains a 40-year, 30,000 hour archive of annotated deep-sea video. This archive is managed through the Video Annotation Reference System (VARS), a searchable relational database of video observations containing approximately 13 million annotations and 1.5 million bounding box localizations. In the recent years, MBARI has been investigating the potential for vector embeddings which lead to each of these 1.5 million localizations to have precomputed vector embeddings using three open-source image embedding models — BioCLIP 2, DINOv3, and SigLIP 2 — and store them in a vector database optimized for similarity search. Vector databases store each point's embedding alongside a payload of associated metadata (linked back to the original VARS entry) and are

highly optimized for rapid approximate nearest-neighbor search across the embedding space (Malkov et al., 2018). This project sets out to answer three questions that follow directly from that infrastructure: whether embeddings can be used to accurately classify regions of interest (ROIs), how much annotator time this could potentially save, and what other capabilities the underlying vector database might offer beyond classification.

Prior work at MBARI has focused on saving annotator time through object detection models that locate organisms within frames. However, the models currently used for detection and classification support far fewer concepts than could realistically be classified — the object detector in use, a YOLO-based model (Jocher et al., 2026), recognizes only 414 concepts, since a more complex classifier requires proportionally more training data to learn each additional concept reliably. Vector embeddings offer a complementary approach; rather than relying solely on a fixed-vocabulary detector, embeddings can be used to label new localizations against a ground-truth reference set directly, or to check and verify ML-generated classification proposals, via nearest-neighbor search within the embedding space. The goal of this project is to provide an additional mechanism for labeling and verifying ML-generated classification labels that goes beyond what traditional object detection alone can provide.

The underlying question for this project, then, is whether embeddings can be used to accurately classify ROIs collected across MBARI's dives and expeditions. These expeditions generate an enormous volume of video for the Video Lab team to review manually: to date, MBARI has logged approximately 7,300 successful dives and nearly 30,000 hours of underwater video, with both figures continuing to grow. This volume of footage now exceeds what human annotators can reasonably annotate manually.

Classification difficulty is further compounded by deep-sea imagery artifacts — variable lighting and brightness, camera angle, motion blur, and camera quality — all of which affect image quality independently of the organism itself. Data sparsity is a related but distinct problem: some concepts have only a single labeled example, which is close to impossible for an embedding-based classifier to learn, since there is no cluster of similar points to draw on. By contrast, a well-represented concept such as *Pyrosoma*, with roughly 500,000 localized examples, is predicted with 90%+ accuracy on a given run — illustrating how strongly classification performance in this pipeline is tied to how well a concept is represented in the underlying vector database.

Beyond raw classification accuracy, a central motivation for this project is time savings for MBARI's annotators. While the full VARS archive spans over 30,000 hours of historical video, more than 1,000 hours of new footage moves through the Video Lab for analysis each year. The goal is not to replace the Video Lab team's domain expertise, but to aid their workflow — ideally by surfacing which ML-generated ROIs can be trusted at a glance and which genuinely need human review, rather than requiring every frame to be reviewed with equal scrutiny.

Our **contributions** are a pipeline that (1) embeds new ROI data and adds it to the vector database when not already present, and (2) classifies incoming data by taking a list of UUIDs, fetching each point's stored vector, performing a similarity search against the database and (3) a

harness to compare runs and parameters with. Requests are batched — multiple UUIDs are resolved in a single HTTP request — to avoid timing out the server or overwhelming it with thousands of individual similarity searches. This vector-first approach is substantially faster than a naive UUID-lookup-then-search pattern, since Qdrant is given the exact query vectors up front and does not need to resolve a UUID to a vector before the similarity search can begin.

Materials and Methods

Dataset

Our dataset is drawn from the VARS annotation collection, comprising approximately 13 million annotations and 1.5 million annotated regions of interest (ROIs). These ROIs span 1,692 unique concepts across seven taxonomic ranks (kingdom, phylum, class, order, family, genus, species). The distribution of examples per concept is heavily long-tailed: well-represented concepts such as *Pyrosoma* have on the order of 500,000 examples and are predicted with 90%+ accuracy on a given run, while rare concepts with only a single example are effectively impossible for the model to learn, since there is no cluster structure for the embedding space to draw on.

Each point in the vector database includes the image ROI (bounding box) along with associated metadata: taxonomy (kingdom through species), depth, latitude/longitude, oxygen, pressure, and temperature. This ancillary data is not currently used in classification but is preserved for future work which may use depth, position, or other logged data to determine animal classification.

Embedding Models

Rather than training a classifier directly on pixel data, we represent each ROI as a fixed-length embedding vector produced by a pretrained image encoder. Each ROI is mapped to a fixed-length vector (768 length) such that visually and semantically similar images land near one another in the embedding space. The individual dimensions have no human-interpretable meaning; only relative distances between vectors are meaningful.

We evaluated three embedding models:

1. DINOv3 (Siméoni et al., 2025) — a self-supervised Vision Transformer that consistently outperformed the other two models across our experiments. This self-supervised training objective appears to produce embeddings that better separate visually and taxonomically distinct concepts.
2. BioClip2-image (Gu et al., 2025) — trained on biological imagery, giving it a useful prior for distinguishing animals and organisms, though it underperformed DINOv3 in our tests.

3. SigLIP2-image (Tschannen et al., 2025) — a multilingual vision-language encoder trained on paired image/text data. SigLIP2 consistently underperformed the other two models on our dataset and was ultimately dropped from the ensemble (see Results).

k-NN Classification

We selected k-nearest neighbors (k-NN) as our classification approach (Hastie et al., 2009), since it is non-parametric—it requires no training step and directly compares a query point's embedding against stored, labeled embeddings to make a prediction. This property makes it well-suited to a pipeline that needs to classify large volumes of incoming data continuously as new expeditions are processed.

We experimented with several classification approaches before settling on k-NN, including centroid-based classification and a Support Vector Machine (SVM). The SVM approach was set aside primarily due to the retraining overhead it would require as new data and concepts are added k-NN's lack of a training step avoids this compute cost entirely.

For a given query point, KNN identifies the K nearest neighbors in embedding space — using cosine similarity as the distance metric — and predicts a label based on those neighbors. The choice of **k** materially affects results: a small **k** (e.g., 3) is sensitive to individual noisy points, while a larger **k** smooths predictions but risks pulling in less-related classes. We additionally weight each neighbor's vote by its distance from the query point, so that closer neighbors contribute more strongly to the final prediction than more distant ones. This distance weighting can shift the outcome even when a majority of the unweighted **k** neighbors belong to a different class than the weighted result would suggest, since a very close match can outweigh several more-distant votes. After sweeping multiple values of **k** and evaluating both weighted and unweighted voting, we settled on **k = 5 with distance weighting as our final configuration.** (see Results)

Ensembling

Because we have embeddings from three independent embedding models, each can be queried independently via k-NN to produce its own label prediction for a given point. We combine these three predictions using the following decision logic:

1. **All three models agree** → the unanimous label is used with high confidence.
2. **Two of three models agree** → the majority label is used.
3. **No agreement** → the pipeline falls back to the prediction from whichever single model reported the highest confidence for that point.

Agreement across independently-trained models is a meaningful signal. Since each model arrives at its prediction independently, unanimous agreement across all three is strong evidence that the prediction is correct.

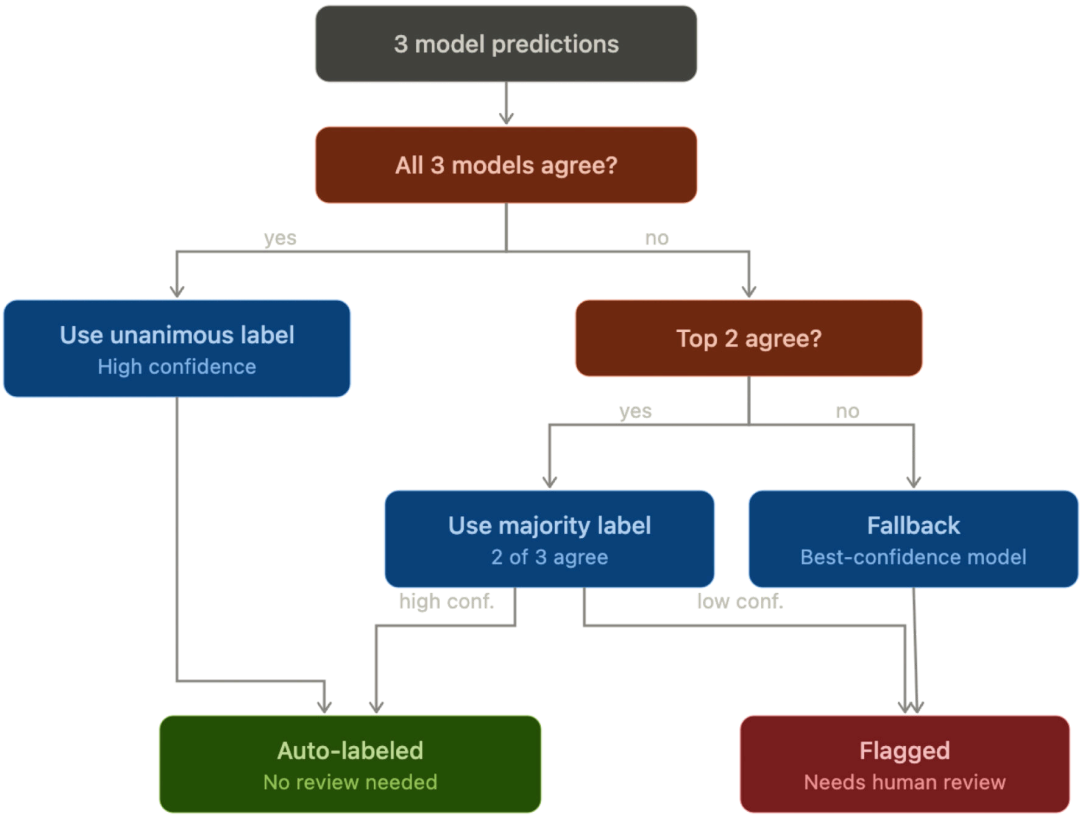


Figure 1

Figure 1. Shows the flow of being auto-labeled and being flagged.

Taxonomic Distance Scoring

A central finding of this project is that a flat, binary right/wrong accuracy metric is a poor fit for hierarchical taxonomic labels. A prediction at the genus level when the true label is a species within that genus is not necessarily *wrong* — it may simply be less specific. Likewise, a prediction that is one rank too specific (e.g., a species prediction when the true label is only known to the genus level) is not a full miss either. We also want to make sure we aren't going all the way to the very top of this tree and say "Animalia" each time just for some partial credit no matter what we predicted. So we stop giving partial credit at 3 or more steps.

To account for this, we implemented a rank-aware scoring system that “climbs the taxonomic tree” to check for a match at each rank (kingdom, phylum, class, order, family, genus, species) rather than only checking for an exact match. Each prediction is scored based on the number of steps between the predicted and actual rank:

Distance (steps)	Score
0 steps off	100 % - full credit
1 step off (near miss)	85 %
2 steps off	60 %
3+ steps off	0 % - no credit

Table 1
Predictions that are three or more steps away from the actual rank are considered too imprecise to be a meaningful match and therefore receive no credit. This general approach follows established work on hierarchical classification evaluation, which argues that flat measures like accuracy and precision are inadequate for problems where classes are hierarchically related and that errors closer to the true class in the hierarchy should be penalized less severely (Costa et al., 2007; Riehl et al., 2023).

Non-Organism Data

A subset of VARS annotations are not marine organisms — trash, plastic, rope, wood, and equipment are observed and labeled in our deep-sea footage. Because these categories exist outside our taxonomic tree entirely, they cannot be meaningfully scored using rank-distance logic (there is no shared ancestor to “climb toward,” and a distance threshold calibrated for the biological taxonomy would not be appropriate for a completely different classification scheme). We therefore label these points as “non-marine organisms” and exclude them from pipeline accuracy and precision metrics, while still retaining them as valid, usable data.

Vector Database Infrastructure (Qdrant)

All embeddings are stored in Qdrant (Qdrant Team, 2025), a vector database optimized for high-speed similarity search over large collections of vectors. Each vector is stored alongside its metadata (concept label, taxonomy, ancillary environmental data), enabling the k-NN step to retrieve nearest neighbors directly without a separate lookup step.

New data is embedded and inserted into the collection as needed. Classification requests are batched — rather than issuing one HTTP request per point, we submit a list of UUIDs in a single batched request that retrieves each point’s stored vector and performs a similarity search

against it. This avoids overwhelming the server with thousands of individual single-vector queries and is substantially faster than a naive point-by-point lookup-then-search pattern, since Qdrant is given the exact query vectors up front rather than needing to resolve a UUID to a vector before it can begin the similarity search.

Results

Evaluation Set Construction

To evaluate the pipeline, we built a stratified evaluation set restricted to concepts with sufficient representation for reliable evaluation. Specifically, we used a subset of concepts with at least 100 labeled examples in the *vars_annotations* collection, and randomly sampled exactly 100 points from each using a fixed random seed (42), so that the same evaluation set is reproduced across runs and different model configurations can be compared fairly on identical data. Concepts with fewer than 100 available examples were excluded from this evaluation set rather than sampled with replacement or padded artificially.

Model Comparison

Across our experiments, DINOv3 and BioCLIP 2 consistently outperformed SigLIP 2 (Figure 2). SigLIP 2 is retained in the full pipeline and ensemble (Figures 1–4), since the agreement-based voting scheme naturally down-weights its contribution when it disagrees with the other two models. For visual clarity in the remaining figures, Figures 5 and 6 are shown using only DINOv3 and BioCLIP 2.

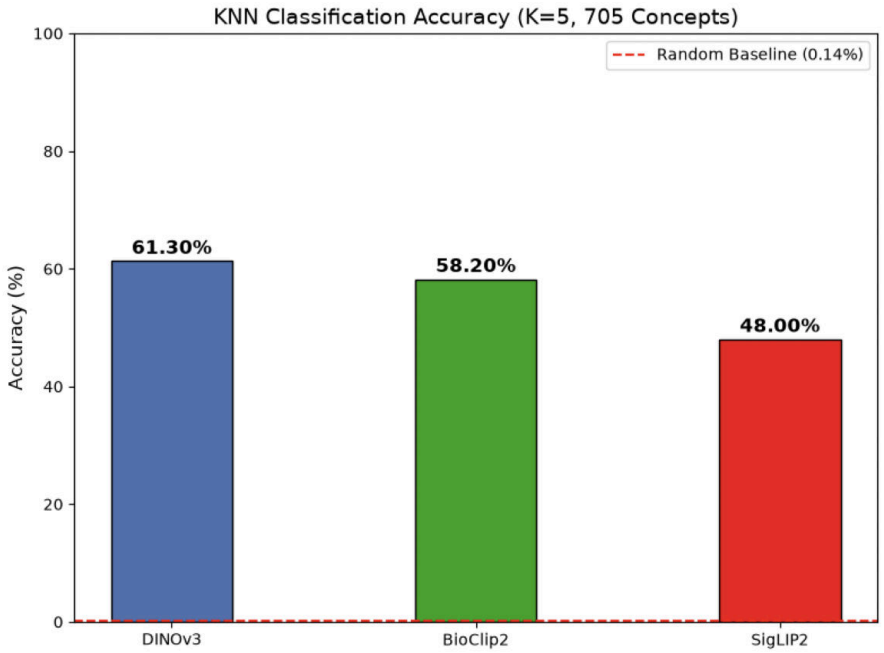


Figure 2

Figure 2. k-NN raw accuracy for all three models across 705 concepts with at least 50 labeled examples in the collection — a looser threshold than the ≥100-example filter used for the main pipeline evaluation reported elsewhere in this paper.

Per-model raw accuracy (single-model k-NN, before ensembling), simply if they voted exact matching, no taxonomical hierarchy:

Model	Accuracy
DINOv3	61.30 %
BioCLIP 2	58.20 %
SigLIP 2	48.00 %
Ensemble (DINOv3 & BioCLIP 2)	64.59 %

Table 2

Table 2. Per-model raw accuracy compared to ensemble accuracy for a representative evaluation run (k=5, weighted k-NN). The ensemble substantially outperforms either individual model, reflecting the value of requiring agreement between independently-trained models.

Pipeline Performance Overview

Exploring how K affects our accuracy we find that increasing K does not uniformly improve performance, it trades coverage for very little gain elsewhere. Coverage rate is highest at K = 3 and K = 5 (86.0% and 86.8%) and then drops sharply as K grows, falling to just 74.9% by K = 20. Pipeline accuracy follows the same pattern, peaking at K = 5 (63.9%) and declining steadily to 54.8% at K = 20. Precision, by contrast, stays essentially flat across all values of K (73.6–74.6%), meaning that larger K values are not making individual predictions meaningfully more correct. They are simply making the pipeline less willing to commit to a label in the first place, since larger neighborhoods are more likely to include mixed or conflicting classes and drive confidence below the auto-label threshold. (Figure 3) This is consistent with the earlier discussion of K in the KNN section: a larger K smooths predictions but risks pulling in less-related classes, and this sweep shows that cost outweighs any benefit beyond K = 5. K = 5 sits at the peak of both accuracy and coverage while matching the precision achieved at every other value tested, making it the best-performing choice among those evaluated rather than an arbitrary default.

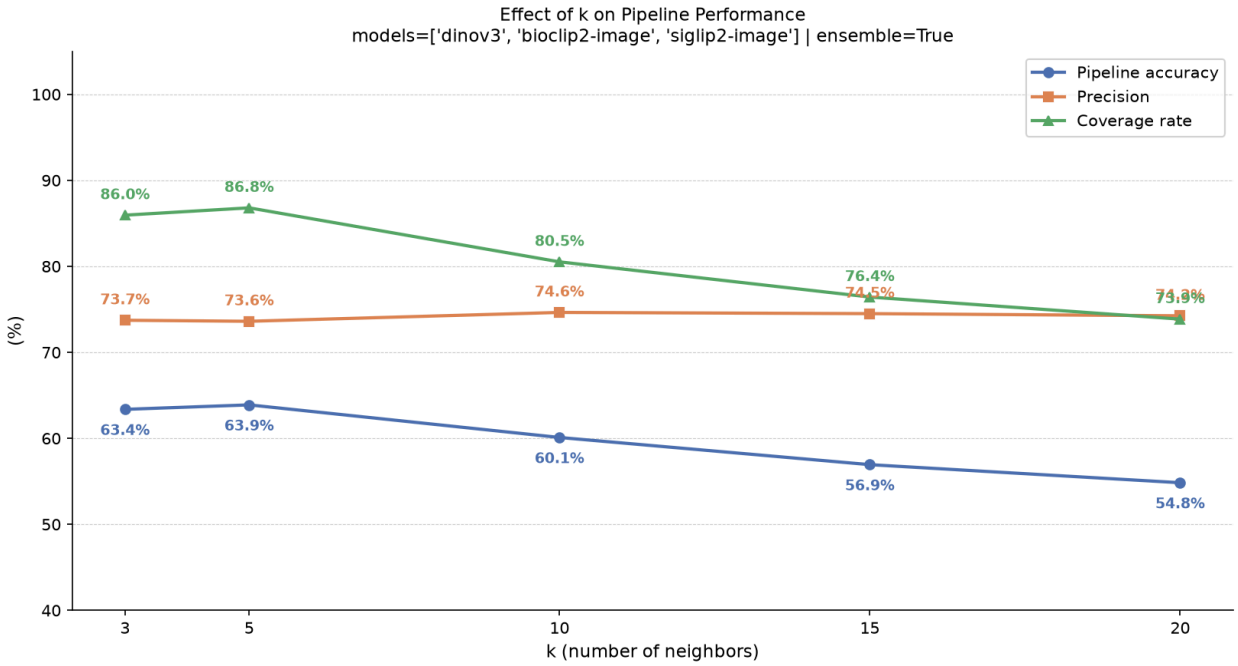


Figure 3

Figure 3. Effect of K on pipeline performance across five values (3, 5, 10, 15, 20), holding all other parameters fixed (models = DINOv3, BioCLIP2, SigLIP2; ensemble = True). Pipeline accuracy and coverage rate both peak around K = 5 and decline steadily as K increases, while precision stays roughly flat across all values tested.

Across a representative evaluation run of 53,239 total points with all 3 models (DINOv3, BioCLIP 2, SigLIP 2), with minimum examples per concept of 100, and k = 5, we get approximately 6,113 points that were excluded as non-marine-organism data (trash, debris, and similar), leaving 47,126 classifiable points. Within this classifiable set, the pipeline achieved a pipeline accuracy of 63.89%, meaning that this was the overall proportion of points correctly labeled out of all the marine organisms the pipeline attempted to classify. Precision — the proportion of points that were correct among those the pipeline was confident enough to commit to a label on — was 73.61%. Coverage rate, defined as the proportion of classifiable points the pipeline was confident enough to auto-label rather than flagging for human review, was 86.8%. These three metrics are related: coverage rate × precision ≈ pipeline accuracy (86.8% × 73.61% ≈ 64%), reflecting that overall accuracy is a function of both how much of the data the pipeline is willing to commit to a label on, and how often it is correct when it does.

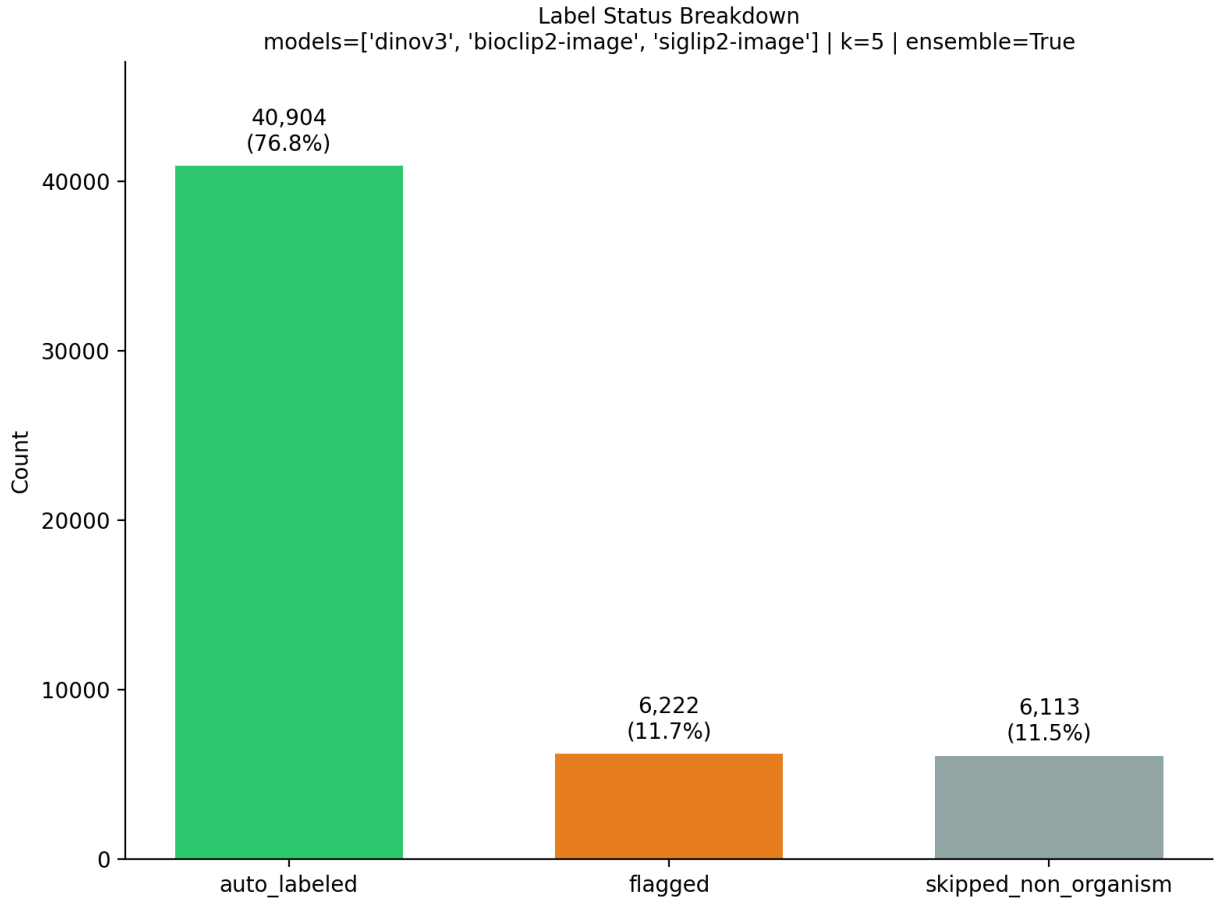


Figure 4

Figure 4. Breakdown of label status across the evaluation set: auto-labeled (high-confidence, no review needed), flagged (low confidence or model disagreement, needs human review), and skipped/non-organism (excluded from scoring).

It is important to note that a query point is always excluded from voting for itself during nearest-neighbor search, preventing the trivial case of a point matching itself at zero distance. However, this does not exclude near-duplicate frames of the same individual organism captured moments before or after the query point within the same dive, which may still appear among its nearest neighbors. Because such frames can be nearly identical in embedding space, their presence could make classification easier than it would be for a genuinely independent observation, meaning the accuracy figures reported here are best read as an upper bound on performance against truly novel footage rather than a guaranteed floor.

Label Status Breakdown: Triage, Not Full Automation

We define three label statuses for each point: **auto-labeled** (high confidence, either all three models agreeing or two models agreeing with sufficiently high confidence), **flagged** (models could not reach consensus, or reached consensus with too low a confidence to auto-label), and **skipped/non-organism** (points excluded from taxonomic scoring, per the section above).

When we auto-label, we are stating that the pipeline has predicted a label with confidence higher than the cutoffs shown below (Table 3). This is because committing to a specific taxonomic label carries more risk of error than committing to a coarser one. As a result, we do not apply a single fixed confidence cutoff across all predictions. Instead, we use a rank-dependent set of confidence thresholds: a prediction must clear a higher bar to be auto-labeled high confidence at the species level than it does to be auto-labeled at the class level, reflecting the fact that species-level distinctions are finer and easier to get wrong. If a prediction's confidence does not meet the threshold for its predicted rank, the pipeline walks up the taxonomy tree and checks whether confidence meets the threshold at each successively coarser rank, auto-labeling at the first rank along that path that clears its bar. A point is only flagged for human review if no rank meets its threshold. Table 3 lists the threshold used at each rank.

Rank	Threshold
Kingdom	NA
Phylum	NA
Class	0.40
Order	0.50

Family	0.60
Genus	0.70
Species	0.80

Table 3

Table 3. Confidence threshold required to auto-label a prediction at each taxonomic rank. Thresholds decrease at coarser ranks, reflecting that less specific labels require less certainty to commit to. A prediction is auto-labeled at the coarsest rank along its predicted taxonomy path whose confidence meets the corresponding threshold; if no rank meets its threshold, the point is flagged for human review.

This breakdown reframes what “time savings” means for this pipeline. We are not aiming for full automation — that is not achievable with the current setup and error rates. Instead, the value of the pipeline is in **targeted triage**: rather than annotators reviewing every point in a video equally, the flagged status identifies specifically which points genuinely need a human’s judgment, versus which can be trusted without review. This lets the video lab team prioritize their limited review time on the ambiguous cases rather than treating all footage as equally uncertain.

Per-Concept Accuracy

Per-concept accuracy across all evaluated concepts forms a smooth, long-tailed distribution rather than a clean split between “concepts the model handles well” and “concepts it doesn’t.” High-performing concepts tend to be both well-sampled (many stored examples in Qdrant) and visually distinct from other concepts in the dataset. Sample count loosely tracks with accuracy, but not perfectly — some well-sampled concepts still underperform, suggesting that visual or taxonomic ambiguity, not just data scarcity, contributes to lower accuracy for certain concepts.

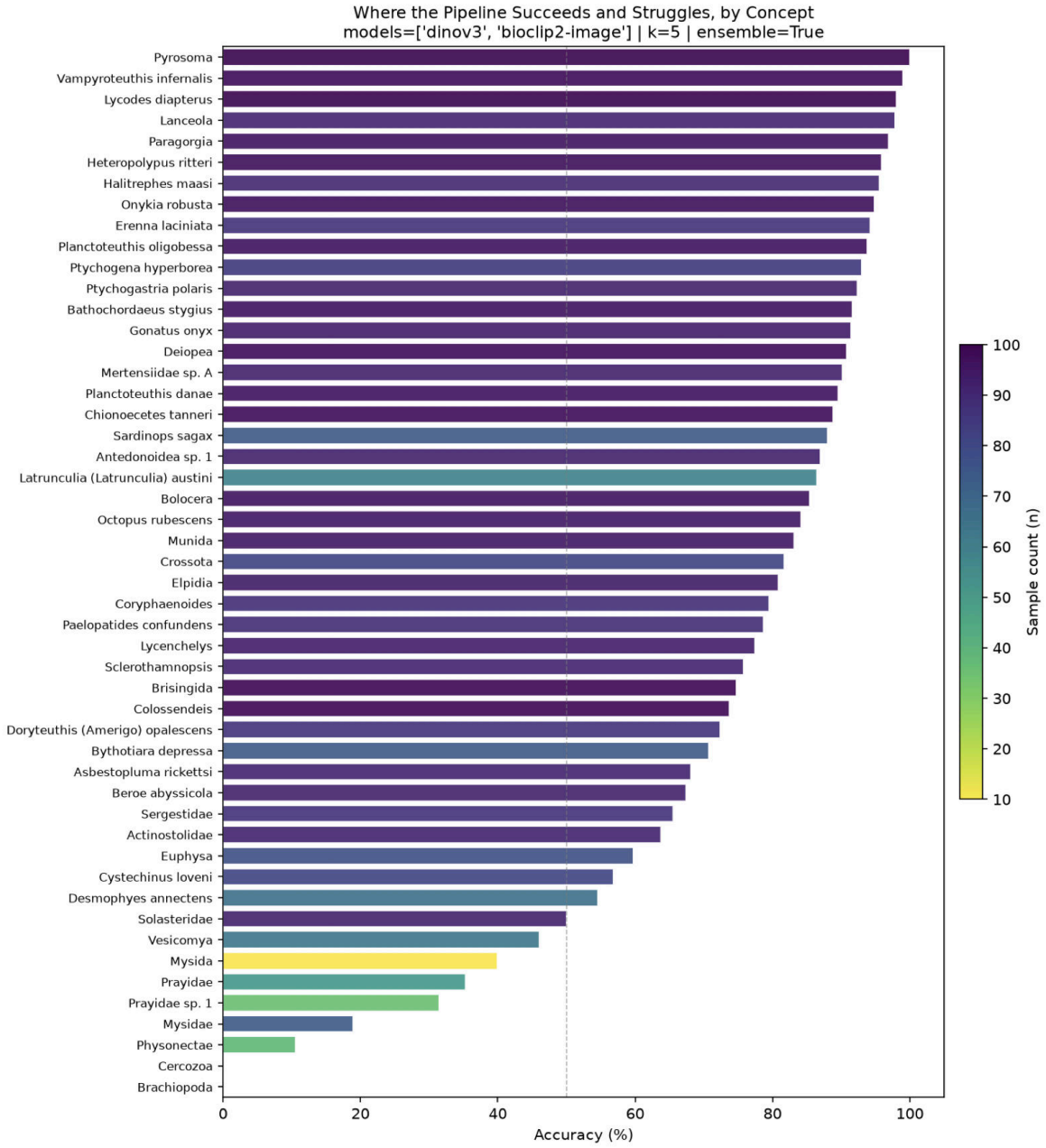


Figure 5

Figure 5. Per-concept accuracy for a representative subset of concepts, sorted from highest to lowest and colored by sample count. The smooth decline (rather than a sharp cliff) supports the use of rank-aware scoring, since many “incorrect” predictions are likely near-misses rather than complete failures.

Common Misclassifications

Examining the most frequent misclassification pairs reveals that confusion clusters around concepts that are either visually similar or taxonomically adjacent. The most common misclassification in our evaluation set was *Copepoda* being predicted as *Euphausiacea*. Other frequent confusions (e.g., *Sebastes* → *Sebastolobus*, *Benthopecten* → *Ophiuroidea*) likewise occur between closely related taxa, reinforcing that many pipeline errors are near-misses at a coarser taxonomic rank rather than arbitrary mistakes — precisely the pattern our rank-aware scoring system is designed to credit appropriately.

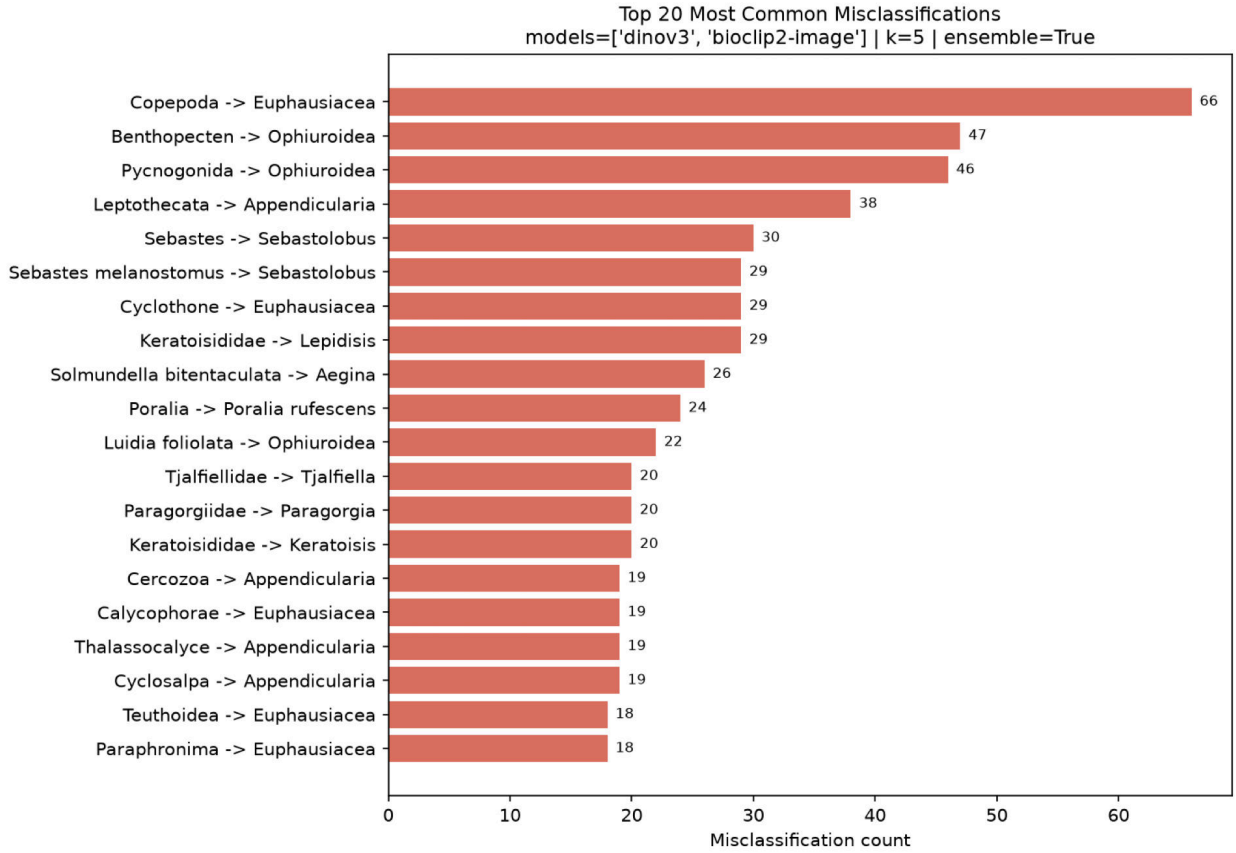


Figure 6

Figure 6. The twenty most common misclassification pairs (predicted → actual) in the evaluation set, by count. Shown using DINOv3 and BioCLIP 2 only, for visual clarity.

Discussion

Going back to our original research questions; can embeddings be used to accurately classify ROIs? Yes, with meaningful caveats. The pipeline achieves an overall pipeline accuracy of approximately 63.89% and a precision of approximately 73.61% on the predictions it is confident enough to commit to. Performance is strongly tied to how well a concept is represented in the

vector database — well-sampled, visually distinct concepts are classified reliably, while rare or visually ambiguous concepts remain difficult regardless of which embedding model is used.

Answering the questions of how much time could this potentially save? The time savings this pipeline offers do not come from full automation, which is not achievable given current error rates, but from targeted triage. Rather than requiring annotators to review every ROI with equal scrutiny, the pipeline's flagging logic identifies the subset of points — roughly 12% of the total — that the models could not confidently agree on, and surfaces specifically those for human review. The remaining points can be trusted without review. This reframes the annotator's task from reviewing all incoming footage uniformly to prioritizing the cases that genuinely require judgment, which is a more realistic and immediately useful form of time savings than attempting to replace the Video Lab team's role outright.

Additionally we found the vector database to be useful well beyond classification. It can flag outliers — points whose embeddings don't sit near any existing cluster, hinting at novel or unseen concepts. It supports natural-language queries, since embeddings relate images and text in the same space (e.g. "red fish under a certain depth") (Qdrant Team, 2025). And given a single query image, it can return visually similar ROIs directly, with no label required at all.

Together, these capabilities suggest that the value of this pipeline is not limited to the classification task it was originally built for. The same underlying vector database — populated once, at embedding time — can support outlier flagging, natural-language search, and reverse-image lookup with no additional model training, since all of these operations are just different ways of querying the same stored embedding space.

References

Hastie, T., R. Tibshirani, and J. Friedman. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer. https://doi.org/10.1007/978-1-4757-4645-4

Costa, E.P., A.C. Lorena, A.C.P.L.F. Carvalho, and A.A. Freitas. 2007. A review of performance evaluation measures for hierarchical classifiers. *AAAI Workshop Technical Report* WS-07-05: 1–6. https://cdn.aaai.org/Workshops/2007/WS-07-05/WS07-05-001.pdf

Gu, J., S. Stevens, E.G. Campolongo, M.J. Thompson, N. Zhang, J. Wu, A. Kopanev, Z. Mai, A.E. White, J. Balhoff, W.M. Dahdul, D. Rubenstein, H. Lapp, T. Berger-Wolf, W.-L. Chao, and Y. Su. 2025. BioCLIP 2: Emergent properties from scaling hierarchical contrastive learning. *arXiv preprint* arXiv:2505.23883.

Jocher, G., J. Qiu, M. Liu, S. Lyu, F.C. Akyon, and M.E. Kalfaoglu. 2026. Ultralytics YOLO26: Unified real-time end-to-end vision models. *arXiv preprint* arXiv:2606.03748.

Malkov, Y.A., and D.A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *arXiv preprint* arXiv:1603.09320.

Riehl, K., M. Neunteufel, and M. Hemberg. 2023. Hierarchical confusion matrix for classification performance evaluation. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 72(5): 1394–1412. https://doi.org/10.1093/jrsssc/qlad057

Siméoni, O., H.V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. 2025. DINOv3. *arXiv preprint* arXiv:2508.10104.

Tschannen, M., A. Gritsenko, X. Wang, M.F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai. 2025. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint* arXiv:2502.14786.

Qdrant. 2025. Qdrant: High-performance, massive-scale vector database and vector search engine. https://github.com/qdrant/qdrant. Accessed: 2025-10-05.