Finding Ariel: Comparing Embedding Extractors for Zero-Shot Clustering and Classification of Fish in Danish Waters

Altea Fogh
IT University of Copenhagen
alfo@itu.dk

Supervisors

Yucheng Lu
yucl@itu.dk
IT University of Copenhagen

Malte Pedersen
mape@create.aau.dk
Visual Analysis and Perception Lab, Aalborg University

Stefan Hein Bengtson
shbe@create.aau.dk

ABSTRACT

The efforts to improve the state of the marine environments require the collaboration of research, industry and governments. One example is the MOSAR project, in which underwater cameras are employed to assess the impact of artificial reefs on the fish presence in the harbour of Hundested in Denmark. Images extracted from the video recording, after careful manual labelling and cropping around their bounding boxes, are used to test a zero-shot clustering of embedding pipeline, where four different models extract embeddings that are then passed to each of four clustering algorithms to determine the best model–algorithm combination. The results show that the self-supervised embedding extractor DINOv2, in combination with Agglomerative Clustering, provide the best outcome in terms of adjusted mutual information score, with a score of 31% on the training data. The clusters are then used for a K-nearest neighbours classifier, aimed at categorising embeddings extracted from unseen image. Additional experiments involving pre-processing images by gray-scaling and enhancing, and decreasing the number of classes, show that gray-scaling has a positive impact on the task, achieving an AMI score of 36% on the test set, while enhancing the contrast and decreasing the number of classes bring only marginal improvements, if any. The code and the results are available on github at this link.

Keywords: Unsupervised Clustering, Zero-Shot Learning, Underwater Computer Vision

1 INTRODUCTION

The rapid decline of the oceans due to human actions has spurred to the development of initiatives aimed at defining feasible actions for the improvement of the marine environment [1].

The Ocean Decade initiative, a framework defined by the United Nations, focuses on defining such actions by bringing together researchers, governments, NGOs, businesses and industries, and communities of all sorts, to tackle the issues from multiple sides, and support one another in the efforts. They defined challenges to be tackled and solved before the end of the decade. Among the challenges, the 2nd [2] focuses on designing ecosystem approaches to improve biodiversity by fostering cooperation between private and public organisations, and the 7th [3] focuses on expanding the oceans observing systems, with automatic tools powered by Machine Learning (ML) and Artificial Intelligence (AI).

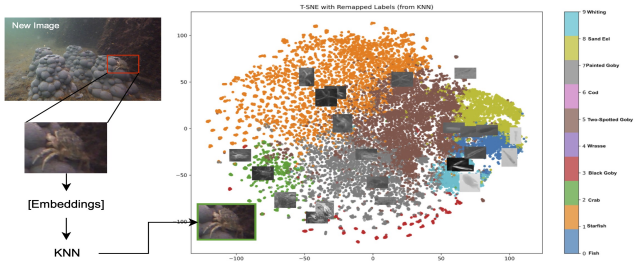


Figure 1: Illustration of expected results of the pipeline. Given a new subject, it should be classified correctly based on similarity of embeddings with training space

As seen with the design of these challenges, it is necessary to develop automatic systems for the monitoring of the seas. The areas affected by negative human behaviour are too big to manually surveil, so numerous automatic systems have been developed [4]. Among them, computer vision (CV) has been crucial in automatically monitoring underwater environments, for assessing improvements following human intervention. Using these CV models it is possible to analyse increasing amounts of data in mere hours, which would otherwise take years to go through. Underwater object detection and species classification is a particularly difficult task [4], due to the varying environmental conditions, based on light refraction and amount, turbidity, general density of the fluid and presence of debris.

Denmark in particular is suffering from the negative impacts of fishing with bottom trawling gear and oxygen depletion of the seas [5]. It has been estimated that around 44% of the Danish sea area is negatively affected by various anthropogenic factors, and that there has been a decrease of more than 20% in the amount of benthic animals, mainly due to fishing.

As part of the efforts to bring back marine species in the coastal areas around Denmark, the MOSAR (Monitoring of Small-Scale Artificial Reefs) [6] project was defined as a collaboration between Anemo Robotics, ReefCircular, and DTU Aqua. The project uses underwater cameras to monitor artificial reefs deployed in the Hundested Harbour in north Zealand, DK, over the span of 12 months. The goal is to assess the impact of the artificial reefs on the number and diversity of species, by using the recorded footage to train ML models to automatically analyse the marine biodiversity.

To do that, they employed experts to manually annotate a dataset of images, which required hundreds of hours of human work [6]. The labelled dataset was used for this project.

While supervised ML models can be very effective in recognising species they have seen during training, they consistently fail at detecting new species. Additionally, if the model has not been trained on sufficient data with enough examples, it is likely to mis-classify even familiar species when they appear in different positions or under varying lighting conditions. The issue could be mitigated by re-training the model regularly and adding new image samples.

However, increasing model complexity and re-training supervised models is not sustainable in the long run. The computational power and associated emissions would counteract any positive effort to improve the marine and terrestrial environment. Besides, from a practical perspective, retraining models implies additional manual annotation, which would undermine the time-saving of using computer vision. Therefore, inspired by the experiments and finding of Lowe et al. [7], who demonstrate that self-supervised models can extract high-quality embeddings for clustering without the need for extensive task-specific fine-tuning, we investigated whether a similar approach could be applied to underwater species monitoring.

The idea is to compare a model designed as a feature extractor, with feature extraction backbones from pre-trained classifiers, for the task of zero-shot clustering of embeddings. Each of the models is pre-trained on different data, that vary in size and in their level of generalisation compared to the task of fish classification. By identifying which model generates the most useful representation for clustering of fish species, we can possibly avoid both extensive re-training and the use of very complex, and computationally expensive models. After that, we conducted additional experiments to see if there are ways to improve performance by manipulating the images through pre-processing and augmentation.

To summarise, this project aims at implementing a pipeline for zero-shot clustering of embeddings extracted from images of marine life, in order to compare the performance of different models, designed for different tasks and pre-trained with different datasets, at varying degrees of generalisation. The clusters obtained from the best combination will then be used for a k-nearest neighbour classifier, which will categorise embeddings of a previously unseen image, hopefully recognising the correct species. Figure 1 presents the graphical overview of the process.

The analysis should provide a starting point for future improvement of the task, having identified which algorithms do and do not work for underwater classification.

The contributions of this paper are:

(1) Comparing embedding extractors trained on different types of datasets, both in size and degree of generalisation, for the task of zero shot clustering of embedding of underwater fish images.
(2) Assessing the impact of different pre-processing methods applied to the dataset to improve the classification task.

The paper is structured as follows: In section 2 we going to present the current research on the topics of zero-shot learning, clustering and underwater classification. The dataset we are working with is presented in section 3, and the methods used in section 4. Section 5 shows the results of the initial experiments, plus considerations of additional steps we took to improve performance. Final considerations and future work are discussed in section 6.

2 RELATED WORK

2.1 Zero-Shot Learning

Zero-shot learning is the task of learning to recognize objects whose instances have not been seen during the training of the model. The key difference between zero-shot and generalised zero-shot lies in the presence of training classes at test time. The generalised one allows for the presence of training classes during evaluation, which is usually more realistic in current tasks, but not applicable to the one for this paper.

Xian et al. [8] conducted a comprehensive analysis of the current status quo of the field of zero-shot learning. They first defined a benchmark for evaluation protocols, while also proposing a new dataset, and comparing different state of the art methods, for both generalised and classic zero-shot learning.

Lowe et al. [7] conducted a series of experiments aimed at analysing whether general self-supervised pre-trained models can be used in a zero-shot setting to generate useful representations of images, in order to then group unseen data into meaningful clusters. They conclude that it is possible to create well-defined clusters using encodings from self-supervised feature encoders, by testing different methods on a suite of well known, diverse datasets. They identify agglomerative clustering as the most successful clustering algorithm for this task.

Shojaee and Baghshah [9] propose a semi-supervised zero-shot learning approach that works on an embedding space corresponding to abstract visual features. They use both labelled samples of seen classes and unlabelled samples of unseen classes to find a representation of the labels on a space, and then use a clustering algorithm to assign labels to instances of unseen classes.

2.2 Unsupervised clustering for image classification

Clustering algorithms fall under the task of unsupervised learning, in which the models learn to identify patterns in unlabelled data. Specifically for images, the task aims at dividing them into meaningful groups, based on characteristics extracted from the images.

Caron et al. [10] present a clustering method that jointly learns parameters of a neural network and the cluster assignments of the resulting features, to define an end to end training of visual features on large datasets. Yang et al. [11] propose a method to simultaneously learn feature representations and perform clustering, optimizing the embedding space specifically for k-means. Ronen et al. [12] introduced a clustering method using deep-learning, DeepDPM, which does not require the number of clusters to be specified beforehand, but that is inferred during learning, by leveraging a split/merge framework and a dynamic architecture that adapts to the changing number of clusters. Van Gansbeke et al. [13] propose a two-step approach to automatically group images into semantically meaningful clusters, unlike previous single-pipeline methods.

First, semantically rich features are obtained via a self-supervised representation learning task; these features then serve as priors for a learnable clustering method, enabling a focus on higher-level features than in earlier approaches.

The solutions presented above are some of the more complex approaches to solve the image classification task. More traditional approaches, such as the process followed by Lowe et al. [7], and Rodrigues et al. [14], involve a multi step approach of extracting image characteristics, and then clustering with algorithms such as K-means [15], Agglomerative Clustering [16], etc.

2.3 Underwater fish image classification and detection

This paper focuses specifically on feature extraction and clustering of fish images, in unconstrained underwater environments of Denmark.

In their paper, Jian et al. [4] give an overview of different underwater object detection methods, both relying on hand made features, and on deep learning and automatic features definition. They also present seven representative datasets commonly used for underwater object detection. Subsequently, they analyse the main current challenges of the task, and offer some future suggestions.

On top of that, Ayyagari et al. [17] analyse the current available marine datasets annotated for detection and classification, and investigate the accuracy of models either pre-trained or not pre-trained on the fish domain. They conclude that pre-training on a specific fish dataset, OzFish [18], appears to yield the best performance for the fine tuning task, compared to those trained without pre-training and those pre-trained with other marine datasets. They also highlight that pre-training with other marine datasets leads to a worse performance compared to a random initialisation or pre-training with the COCO dataset, when tested on unrelated marine datasets. Additionally, they test the performance on the Brackish dataset [19], which is collected in Danish waters, specifically of the Limfjorden in northern Denmark. The paper concludes that none of the trained models showed good performance on this dataset. This is interesting due to the similarity of environment in the dataset used for this project, as both of them are collected in Danish waters.

Rodrigues et al. [14] propose five schemes for automatic classification of fish species, combining different techniques for feature extraction, clustering and input classifiers. They conclude that the combination of principal component analysis (PCA), adaptive radius immune algorithm (ARIA) and k nearest neighbours (k-NN) yields the best results for their live fish dataset.

Chuang, Hwang and Williams [20] conducted a comparison between two fish feature extraction methods, supervised and unsupervised, and conclude that the unsupervised approach achieves better recognition performance on live fish images.

2.4 Feature extractors

Oquab et al. [21] developed DINOv2 as a series of self-supervised image encoders pre-trained on large data. This work was done to close the performance gap with weakly supervised alternatives across benchmarks and without the need for fine-tuning. The model is larger in terms of training data, scale, and with improved parameters tuning.

The most general supervised model used in this paper is ResNet50 [22], which is often used as a benchmark for image classification problems. It is structured as a residual learning framework to improve the difficulties of deep neural network training, and is trained on ImageNet.

MegaDescriptor [23] is another model used for embedding extraction, which is provided as a foundation model by Čermák et al. The paper presents first a comprehensive toolkit for accessing wildlife datasets, and then develop MegaDescriptor to be used as a foundation model for individual re-identification of a range of land species.

Finally, the dataset for the domain specific model was presented by Khan et al. [24]. FishNet is a dataset comprising of underwater species organised into different taxonomical levels, from class down to specific species, which are used to construct three benchmarks, namely fish classification, fish detection and functional trait prediction. They published the weights for the models, and the ResNet50-based classification model benchmark for fish classification was used for this project.

Overall, the literature presented covers diverse topics expansively, but shows the lack of a comparative analysis of different types of feature encoders for the task of fish species classification. It also highlights the need of testing unsupervised solutions, and the importance of comparing models pre-trained on datasets with varying levels of generalisation.

3 DATA PREPARATION

The images of the MOSAR dataset are collected from video footage recorded over three months, in the harbor of Hundested, Denmark. Three different cameras were deployed underwater, of which two directed towards artificial reefs, in order to monitor their impact on the fish presence, and one as a control.

From the video footage, recorded for 1 minute every 30 minutes in the time span of July 2024 to October 2024, a total of 5992 frames were sampled. The data was then manually annotated by marine biologists with bounding boxes and relative species classification.

The images were then cropped around the bounding boxes, resulting in 52.121 images, which were used as training and test data. This process allowed us to only focus on the fish subjects, which is helpful for fish classification instead of detection.

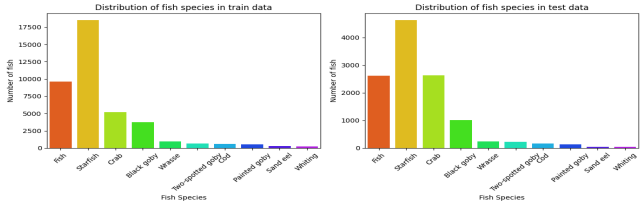


Figure 2: Distribution of fish images by label

Figure 2 shows the distribution of each label in the train and test data, showing the clear class imbalance, that will have to be taken into account. The fish category is peculiar, as there are both fish that were not categorised in their own classes, and some mislabelled images of starfish or crab, among others.

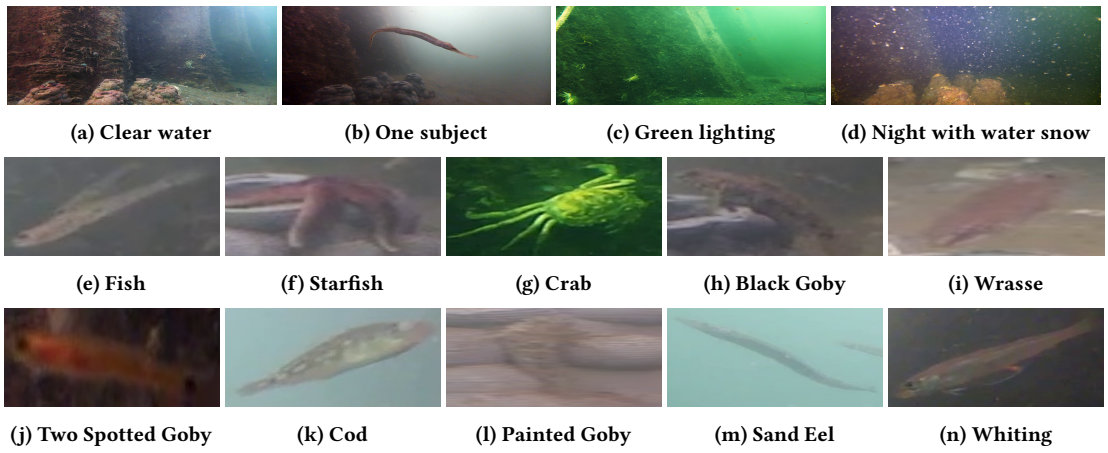


Figure 3: Example of images from the dataset. At the top, the original images with varying degrees of quality and clarity, and below, instances from each class.

Figure 3 gives an overview of the type of images in the dataset. Images a-d show examples of the photos extracted from the videos, with varying degrees of clarity (3a and 3d), colour (3c) and size of the subjects (3b). Additionally, figures e-n show the crops derived from the original photos. It is clear to see that the quality of the images is very varied, due to the water turbidity, the cropping and the nature of the image itself. These examples are chosen specifically as they were more discernable by the human eye than others in the dataset. However, there are also variations in water colour, amount of light, degree of turbidity and granularity of the picture itself, which account for the difficulty of solving a problem such as the one presented in the paper.

4 METHOD

The project focuses on comparing the performance of a feature extractor designed for the task, with the feature extraction backbone from different classification models for zero-shot clustering of the extracted embeddings and subsequent classification of fish images. The decisions made in terms of which embedding extractor, clustering algorithm, and classifiers to use has been decided based on use case, personal experience, and the results of different papers presented in section 2.

Given an image cropped around the bounding box, we employ four models to extract its embeddings, which are then passed to each of four clustering algorithms, that should be able to recognise the difference between the species enough to classify them in different clusters. Having the clusters, we are then able to define classifiers that, given a new image, and having extracted its embedding, is able to categorise the fish in the right cluster. The graphical overview of the process is presented in figure 4

4.1 Embedding Extractors

As embedding extraction models, we used ResNet50 pre-trained on ImageNet and FishNet (which will be referred to as ResNet-ImageNet and ResNet-FishNet, for concision and clarity), Dinov2, and MegaDescriptor. We additionally fine-tuned a ResNet-50 on the data in a fully supervised manner, to be used as a baseline. We only used the feature extraction backbone from the classification models. Refer to the code for more details.

4.1.1 ResNet-50 - Baseline. ResNet50 [22] has been used in the literature (Section 2) as a benchmark for computer vision classification. It uses residual connections to allow the network to learn a set of residual functions which map input to output, which enables the network to be much deeper without having problems with vanishing gradients. We used ResNet-50 as a baseline to compare the other models to. We fine-tuned the available model pre-trained on ImageNet1K_V2 (default weights) from Torchvision, on the training dataset, and subsequently removed the classification head when extracting the feature embeddings. While the baseline cannot be used for the actual task, as it does not fall under the zero-shot learning framework, it allows us to have a meaningful comparison for the other models. Searching the literature, we could not find any similar experiments for underwater fish images, thus making it necessary to have a benchmark for comparison.

4.1.2 ResNet50 - ImageNet. We used ResNet50 pre-trained on ImageNet as the first model of the feature extractors from the classifiers. This version has 50 layers, from which it takes the name, and it is trained on ImageNet 2012, with 1.28M training images divided in 1000 classes. Some of the classes are of different fish species, but it is unclear how the images look. Independently, there are a few fish classes, one of which containing Starfish [25]. This model has been chosen due to its popularity, large training data, and its simplicity compared to other models. We used the default model from torchvision, which is the one pre-trained on ImageNet1k_v2, with 25M parameters.

4.1.3 ResNet50 - FishNet. The other ResNet50 model used was pre-trained on FishNet [24]. FishNet is a dataset containing 94.532 images from 17,357 aquatic species, organised according to taxonomical orders, from order, to family, to genus, to species. It was developed to aid in the task of automatic aquatic species recognition and monitoring, and was used to develop three benchmark models, for fish detection, classification and trait prediction. The fish

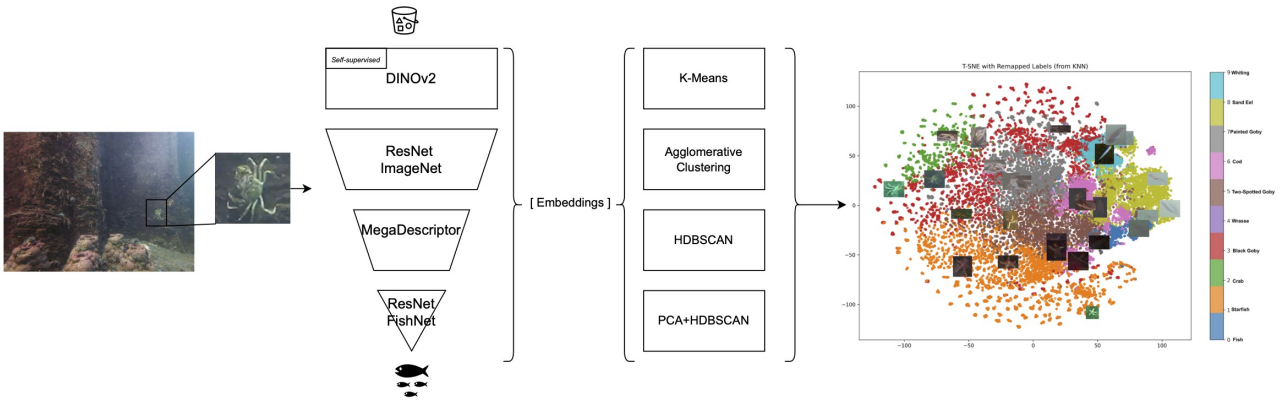


Figure 4: Visual overview of the process. Notably, each embedding extractor is pre-trained on more task-specific datasets.

classification task used ResNet based architectures, among which ResNet50. They used the backbone pre-trained on ImageNet, and then fine-tuned it on FishNet. We used the model pre-trained for Family classification, the lowest taxonomical level we could find. As this model was pre-trained on ImageNet and fine-tuned on FishNet, with images that more closely resemble the ones from our dataset, we can expect to have some interesting results.

4.1.4 DINOv2. DINOv2 [21] is a suite of self-supervised vision transformer models that learn robust features from unlabelled images. It is a foundation model that produces image features that can be used across many tasks without fine-tuning. It has been trained on 40 diverse datasets, combining around 1.2B images from which they extracted a curated set of 142M images for training. We used the small model with 22M parameters.

4.1.5 MegaDescriptor-WildLifeDataset. MegaDescriptor [23] is a foundation model developed for individual re-identification of a wide range of animal species. It is trained on the 29 publicly available datasets presented in the WildlifeDataset toolkit, also presented in the paper, which amount to 253,811 images [26]. The model has been developed for animal re-identification, meaning accurately recognizing individual animals within one species based on unique characteristics. This task is expected to help the model to extract meaningful characteristics of the fish. On the other hand, having been trained on fewer images than other models, it is uncertain how well it can perform. We used the Base model with 109M parameters.

ResNet-50 (both for baseline, ImageNet, and FishNet) generates embeddings of size 2048, DINOv2 of 384, MegaDescriptor of 1024. The difference in sizes is notable enough that it spurred the need to test a dimensionality reduction method.

4.2 Clustering Algorithms

The clustering algorithms used are: HDBSCAN, PCA+HDBSCAN, K-means clustering, and Agglomerative clustering.

4.2.1 K-means Clustering. K-means clustering is one of the simplest clustering algorithms to understand, while being fast and easily implemented [15]. We are using it with the pre-defined number of clusters being the same as the number of classes in the data.

4.2.2 HDBSCAN. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a hierarchical clustering algorithm which extracts a flat clustering based on the stability of the clusters [27]. It transforms the space according to the density/sparsity of the data, and builds the minimum spanning tree of the distance weighted graph. It then constructs a cluster hierarchy of connected components and condenses it based on a minimum cluster size. Finally, it extracts stable clusters from the condensed tree. It does not require a pre-defined set of clusters, but intrinsically finds them from the data. We chose the algorithm due to the ability of define the number of clusters itself, its architecture and its ability to distinguish between noise and meaningful clusters. Additionally, as taxonomy is hierarchical in nature, the task could benefit from a hierarchical model.

4.2.3 PCA+HDBSCAN. We also tested the performance of using Principal Component Analysis (PCA) before HDBSCAN, to assess if using the 50 most descriptive components of the data can improve on the performance of the clustering algorithm. The reasoning being that not all the data is relevant for the embeddings of images, so it can be relatively interesting to assess whether it improves the performance of the model, on top of the fact that each extractor produces different embedding sizes.

4.2.4 Agglomerative Clustering. Agglomerative Clustering is a suite of algorithms that are based on the same idea of starting with each point being its own cluster and then merging them based on a set reason [16]. It thus creates a hierarchy tree of clusters, which is then used to choose the final clusters based on a criterion, such as number needed. We use the default values of the algorithm, which links clusters based on minimising the variance of the clusters being merged, and the number of clusters to be the same as the classes of the data. It is more computationally expensive than K-means, but is the algorithm that has achieved the best results in the Lowe et al. paper, which contributed to our decision to include it in the experiments.

4.3 Classification Method

4.3.1 K-Nearest Neighbours. K-nearest neighbours (K-NN) [28] is a supervised method for classification and regression widely used for its simplicity. It assigns a label to a new data-point based on the majority vote of the k-nearest points, where k is a pre-defined hyper parameter. Normally, clustering and classification are two different tasks that are not combined. However, this project is aimed at classifying images in a novel way, so we modified the output from the clusters such that it would work for classification. In practice, the input data for the K-NN space was modified such that each data-point assigned to a cluster would have the same label. The specific labelling strategy is detailed below. Having the data re-labelled according to the cluster definition, the K-NN space is then generated, and the new data-points classified as usual. Had we not done this, we would have ignored the first part of the pipeline, and we would have had a normal supervised classifier, which was not the goal.

4.4 Performance Metric

In order to assess the performance of the pipeline, and compare the results both between each combination and with the current literature, we selected two performance metric indicators. One to assess the clustering and one for the classification performance.

4.4.1 Adjusted Mutual Information score (AMI). We use AMI as the metric to analyse the clustering algorithms' performance. It is widely used for the task, as it is an adjustment of mutual information that accounts for chance. Mutual information is biased towards two clusterings with higher numbers of clusters, regardless of the amount of information shared. AMI fixes the issue. It is also independent of the absolute values of the labels, which is crucial when the labels of the classes do not necessarily correspond to the labels of the clusters, as in our case. It is more effective than other scores when the ground truth clustering is unbalanced [29], as it is with the data used for this project. The value of the AMI score reaches a maximum of 1 indicating that the partitions are identical, while random labelling is expected to be around 0, with possible negative values.

Having searched the literature on the topic, we could not find any baseline comparison for the AMI score on this task adapted to fish images. Lowe et al. [7] conducted experiments on two animal based datasets, NABirds and iNaturalist-2021. The first being a 24K images dataset of fine-grained bird species, and the second comprising of 100K images of plant and animal species. The best results they achieve on the birds dataset is between 42% and 44% , while for the iNaturalist one, the range is from 4% to 10%. We will keep this in mind, but will mostly focus on the baseline's results, when assessing the performance of the pipeline.

4.4.2 Macro F1 score. Macro F1 score is the unweighted average of the F1 scores per class [30]. We chose it, compared to the weighted average of per-class F1, as it provides a more faithful metric on the behaviour of the classifier on each class, which should all be treated equally, and not discounted based on the number of instances.

4.5 Implementation Details

4.5.1 Image data pre-proccessing. Each image was processed such that the subjects were isolated by cropping around the bounding boxes. Knowing the label of the species, it was possible to thus have a single-subject labelled image of the different fish. The data was split in 80% training, 10% validation, and 10% testing. Before extracting the embeddings, the data went through the same transformation of resizing and normalising to the necessary size for the model used. Each model required a different size, so the details are available in the code.

4.5.2 Cluster labelling strategy. In order to calculate the performance metrics for the classification of new data-points in the generated clusters, we needed to devise a labelling strategy for the clusters. For each cluster we thus got the number of images of each class, and noticed that most clusters contained a varied combination of classes. Labelling the cluster based on the majority class would have brought incorrect results due to the class imbalance favouring Starfish. Therefore, we decided to calculate the percentage of images in each class that are assigned in the cluster, in respect with the total number of images of that class. This means that if cluster X contains 2000 instances of class 0, and 300 of class 5, but class 0 has 9000 images, and class 5 only 600, then the percentage of images clustered in cluster X is much higher for class 5 (50% vs 22%). This calculation allowed us to assign labels to the cluster such that the label represents the class with the highest percentage, with minor changes if there are multiple clusters with the same majority class (approximate greedy approach for labelling when repetitions occur). The cluster-class distribution of labels is available in the GitHub repository.

4.5.3 Gray-scale. As additional experiments, we will test the performance difference between having images in colour and in grayscale. The reasoning will be explain in section 5. We used the "cvtColor" function from OpenCV [31] to convert the images from RGB to grayscale.

4.5.4 CLAHE. Similarly, we used the library OpenCV to enhance the images by improving image contrast [32]. More specifically, we used CLAHE (Contrast Limited Adaptive Histogram Equalization) which divides the images into tiles, and equalises the histogram of each tile. This entails using a transformation function to map pixels from overrepresented regions (brighter or darker) to the pixels of the full region. To avoid amplifying noise, the contrast limiting makes sure that if any histogram bin is above a specified contrast limit, those pixels are clipped and distributed uniformly to the other bins. Finally bilinear interpolation is used to remove artifacts in the tile borders.

5 RESULTS AND ANALYSIS

5.1 Zero-shot clustering results

The adjusted mutual information scores for each combination of embedding extractor and clustering method is presented in table 1.

The first row of the table shows the baseline results of the ResNet-50 fine-tuned on the dataset. As the model was trained for the task of classifying these fish species, it can't be used in the zero-shot learning framework, but it offers a baseline for understanding

how well the other models can perform. As the AMI score is not near 1 (or 100%) we can surmise that the other experiments will not achieve perfect results. Both HDBSCAN algorithms performed quite poorly, while K-Means and Agglomerative Clustering achieve rather good results. This is probably due to the pre-set number of clusters, which provides more accurate results with the adjusted mutual information score. Given this results, it is expected that these two clustering algorithms will provide better scores for the other models.

	K-MEANS	HDBSCAN	PCA+HDBSCAN	AC
ResNet-50-Baseline	65.38	18.13	19.62	72.01
DINOv2	28.36	20.55	20.03	**30.93**
ResNet-50-ImageNet	24.43	0.052	0.034	25.96
ResNet-50-FishNet	7.71	14.94	14.68	7.47
MegaDescriptor	2.7	0.43	0.1	2.79

Table 1: We report the AMI score (%) of each combination of feature extractor and clustering algorithm, to assess how well the training data has been clustered compared to the ground truth labels. The first row shows the results of the baseline, pre-trained on the dataset. The best result is in bold. The best clustering algorithm per embedding extractor is underlined.

For DINOv2, the embedding extractor designed for the task, the best clustering algorithm was agglomerative clustering as well, closely followed by K-means. Interestingly, the difference of performance with HDBSCAN is not as big as for the baseline, and the results are actually better. This suggests that the embeddings extracted might be rather informative overall. Even if the best AMI score is less than half compared to the baseline, the results are rather promising.

ResNet50 - ImageNet performs well with agglomerative clustering and K-means, surpassing DINOv2+ HDBSCAN (with and without PCA). The results are much lower than the baseline, but close enough to DINOv2 to be the best model of the embedding extractor backbones from pre-trained classifiers. This result is probably due to the high level of generalisation of the model. ResNet50-FishNet is the only model whose performance is better with HDBSCAN, by double compared to AC and K-means. MegaDescriptor performs poorly with all clustering algorithms. We can hypothesise that it is due to the nature of the pre-training dataset. While still being in the animal realm, it is not trained on enough images like ResNet-ImageNet, nor on images closely resembling the current task, such as ResNet-FishNet. Thus it does not have the generalisation capabilities of a much larger model, nor the domain abilities of a more focused one.

As a final remark, adding PCA before HDBSCAN either didn't affect the results, or decreased them a bit (apart from for the baseline). However, it decreased the running time significantly, making it a viable addition to future work on the other clustering algorithms.

5.2 Additional experiments

In order to assess how well the clustering algorithms divide the data, we decided to analyse the cluster results in two ways. First, we visualised the clustered data with T-SNE and plotted a few images that should be representative of the clusters (Figure 5 shows the results after KNN for the first experiment we carried on AC after DINOv2, RGB 10 classes). The results showed that the images were clustered also, if not only, based on colour, which is not ideal. Additionally, the plot should show 3 representative images per cluster, but many of them are starfish.

The second analysis involved manually inspecting the clusters, such that we could assess which classes were grouped in which cluster. We could overall see that most clusters had most classes represented, if not all. Refer to the GitHub for these results.

Based on these two analyses, we decided to conduct further experiments that would counteract the colour problem and the cluster-class imbalance. The former by gray-scaling the images, and the latter by modifying the classes such that there are 4, instead of 10. Specifically, Starfish (18.543 instances) and Crab (5213 instances) are kept as such, as they are easily distinguishable, and have a considerable number of instances each. The three types of Goby were merged into one class of 4934 instances, and all the remaining species were added to the Fish class, which becomes the second biggest at 11.669 instances. There is still class imbalance, clearly, but it is much lower, and allows us to keep the results comparable with the 10 imbalanced classes.

Overall, the K-nearest neighbours classifier was ran 4 times. With RGB images divided in 10 classes and in 4, and with gray-scaled images divided in 10 classes and 4. To calculate performance metrics that are class specific, such as F1 score, we manually analysed the clustering results, and mapped each cluster to a label. The strategy is presented in the Methods section (4). Considering that this manual analysis can be trusted only relatively, we also calculate adjusted mutual information for the classification of the new images into the correct clusters, and use this score to choose the best value of K, to then calculate the final results on the test data. Table 2 shows the results of the four classifiers.

	AMI valid	Best K	AMI test	Macro F1 - test
RGB 10 classes	32.70	190	32.24	0.2135
RGB 4 Classes	22.46	200	22.63	0.4714
Gray 10 classes	36.48	190	**36.38**	0.2284
Gray 4 classes	33.36	150	34.11	**0.5221**
Gray 10 Classes enhanced	29.35	170	28.70	0.2173
Gray 4 classes enhanced	27.25	150	27.43	0.4220

Table 2: Results of the K-nearest neighbours classifier on 4 combination of colour and number of classes. The best value of K was chosen based on the highest AMI score on the validation data. The best result per metric is in bold.

Based on both metrics, gray-scaled images work better in this experimental setting. The AMI score is higher for 10 classes, while

the Macro F1 is higher for the 4 classes. The Macro F1 score reflects the performance of all classes, as an unweighted average. If the classifier performs very poorly on some classes, for example due to class imbalance, then having more classes would negatively influence the score, which could explain why it has a higher value for the 4 classes experiments. On the other hand, AMI should not be influenced by the number of clusters, so the value could be considered more reliable. On top of that, the cluster labels have been manually calculated, which is an additional reason why we would choose the best model based on the adjusted mutual information score.

Generally, the improved performance on the gray-scaled images suggest that the embedding extractors are focusing on features that are more representative than colour.

Additionally, we can see that the difference in AMI between the 10 classes and 4 classes for the gray-scaled images is much lower compared to the one for RGB images, which provides another reassurance that the embedding extractors are extracting more meaningful features which have a positive impact on the clustering algorithms performance.

Figure 5 and figure 6 show visual examples of the cluster results after KNN, both for RGB and for gray-scaled. The example images plotted in each cluster are actually more representative of each cluster in the gray-scale plot, compared to the prevalence of starfish in the RGB one. Additionally, in figure 5, there are more clusters overlapping with one another, and that are fragmented in different areas of the graph, compared to the ones in figure 6, in which the clusters are more compact and less divided by other data points.

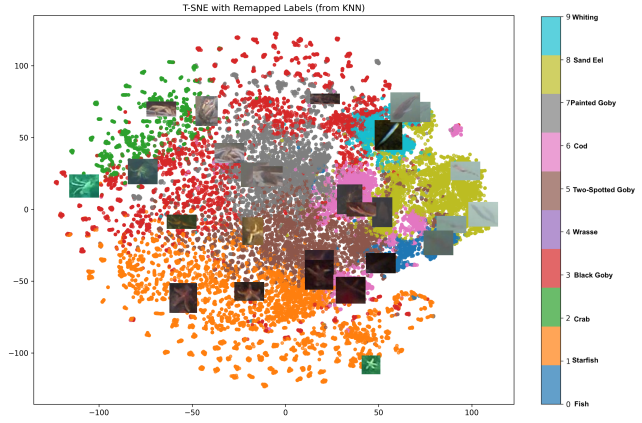


Figure 5: T-SNE visualisation of clusters after KNN, with relative example images (RGB 10 classes). The labels/colours are defined from the manual cluster labelling.

As an additional, and final, experiment, we decided to assess the possible improvement of the model's performance after enhancing the image contrast. Visually, distinguishing between some of the species is rather difficult in some situations, and the varying amounts of light, turbidity, and distance of the fish to the camera, make the discernment even more challenging. Enhancing the image, by adjusting for contrast, could thus improve the results. When analysing the results of the KNN classifier, however, we can see

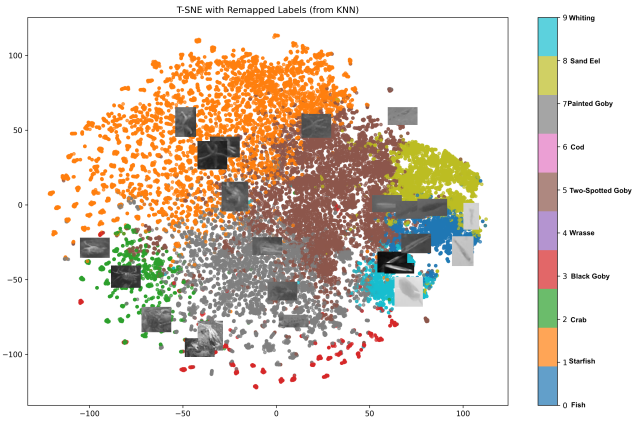
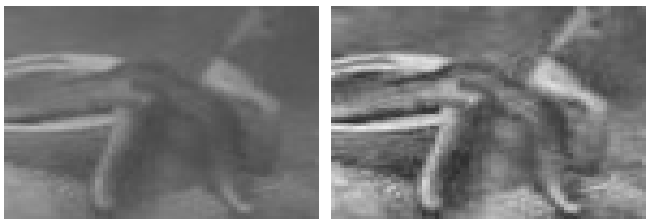


Figure 6: T-SNE visualisation of clusters after KNN, with relative example images (Gray-scaled 10 classes). The labels/colours are defined from the manual cluster labelling.

that the performance is lower than for the non-enhanced images (table 2, last two rows).



(a) example of image before contrast enhancement **(b) Example image after applying CLAHE**

Figure 7: Before and after applying the contrast enhancement algorithm

The AMI score is still higher than for RGB 4 classes, which was expected. Having inspected the enhanced images, of which an example is shown in Figure 7, we could see that while some looked clearer, others looked much worse. Generally though, this happened to images that were already difficult to distinguish. We attribute this results to the varying sizes of the images, which would require a much more adaptive algorithm than the one we used. We tried to see if applying CLAHE to the images before cropping would bring better results, but it was not the case. This shows that it is an issue with the size of the cropped images, which makes them overall difficult to analyse.

To conclude, given the considerations presented above, it appears that the best model combination is agglomerative clustering after DINOv2 embedding extractor, for gray-scale images and 10 classes.

6 DISCUSSION AND FUTURE WORK

The task undertaken was to compare the performance of different models used as embedding extractors for zero-shot clustering and classification of fish in unconstrained underwater environments. The models compared were either designed for feature extraction,

such as DINOv2 which was pre-trained on a large and highly generalised dataset, or were feature extraction backbones from classification models, each pre-trained on datasets varying in size and in their level of generalisation relative to fish images.

The results of the experiments show that, while DINOv2 achieves the best performance, Resnet50 pre-trained on ImageNet achieves an AMI score that is close but lower than DINOv2+AC and DINOv2+ K-means, while outperforming of DINOv2+ HDBSCAN. Overall, it appears that feature extractors pre-trained with more data, and more generalised images perform better than domain specific ones. Additional data manipulation showed that ignoring the water colour by gray-scaling the image improves the results consistently, and that merging classes to decrease taxonomical variation was not a good strategy. Image enhancement did not produce improvements, possibly due to the small size of the cropped images. These results provide a good basis for additional experiments on comparing DINOv2 and ResNet-ImageNet, ideally focusing only on agglomerative clustering and K-means, after gray-scaling the images. Overall, none of the models performed comparably with the baseline, which is to be expected given the difficulty of the task.

The wide range of results highlight both the potential and the challenges of applying unsupervised learning to underwater fish species classification. While the performance is not yet comparable with supervised models, the computational costs of regularly retraining models, and the good initial results of these experiments, suggest that this process can be highly relevant, and it should be analysed further to improve performance.

Future work could include testing agglomerative clustering and K-means (after both DINOv2 and ResNet-ImageNet) with different number of clusters, with the goal of generating clusters with less intra-variations. The additional use of PCA, which in this paper proved to have limited effect on the results but decreased the computational time significantly, could be beneficial especially for AC. It would additionally dampen the difference of embeddings sizes, and possibly provide more comparative results. Attaching a classification head to DINOv2, to assess the performance of the task could also provide insights into the model. On top of that, testing additional self-supervised models could be interesting.

REFERENCES

[1] IOC-UNESCO. State of the ocean report. *IOC Technical Series, 190*, 2024.

[2] United Nations. Challenge 2: Protect and restore ecosystems and biodiversity, 2021. 22 July 2025.

[3] United Nations. Challenge 7: Sustainably expand the global ocean observing system, 2021. 22 July 2025.

[4] Muwei Jian, Nan Yang, Chen Tao, Huixiang Zhi, and Hanjiang Luo. Underwater object detection and datasets: a survey. *Intelligent Marine Technology and Systems*, 2(1):9, 2024.

[5] Anna Rindorf, Esther Beukhof, Josefine Egekvist, Jeppe Olsen, Jonathan Stounberg, and Grete E. Dinesen. Udvikling i havbundens tilstand i havene omkring danmark: Analyser til støtte for status for havstrategiens deskriptor 6 - development in the state of the seabed in the seas around denmark: Analyzes to support the status of the marine strategy's descriptor 6, 2024.

[6] Anemo Robotics. Anemo robotics - mosar dashboard, 2025.

[7] Scott C Lowe, Joakim Bruslund Haurum, Sageev Oore, Thomas B. Moeslund, and Graham W. Taylor. Zero-shot clustering of embeddings with pretrained and self-supervised learning encoders, 2024.

[8] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *CoRR*, abs/1707.00600, 2017.

[9] Seyed Mohsen Shojaee and Mahdieh Soleymani Baghshah. Semi-supervised zero-shot learning by a clustering-based approach, 2016.

[10] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[11] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering, 2017.

[12] Meitar Ronen, Shahaf E Finder, and Oren Freifeld. Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9861–9870, 2022.

[13] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[14] Marco TA Rodrigues, Mário HG Freitas, Flávio LC Pádua, Rogério M Gomes, and Eduardo G Carrano. Evaluating cluster detection algorithms and feature extraction techniques in automatic classification of fish species. *Pattern Analysis and Applications*, 18(4):783–797, 2015.

[15] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[16] Steve Astels Leland McInnes, John Healy. Comparing python clustering algorithms, 2016.

[17] Devi Ayyagari, Talukder Wasi Alavi, Navlika Singh, Joshua Barnes, Corey Morris, and Christopher Whidden. Dataset selection is critical for effective pre-training of fish detection models for underwater video. *ICES Journal of Marine Science*, 82(4):fsaf039, 2025.

[18] Australian Institute of Marine Science (AIMS). Ozfish dataset - machine learning dataset for baited remote underwater video stations, 2019.

[19] Malte Pedersen, Joakim Bruslund Haurum, Rikke Gade, and Thomas B Moeslund. Detection of marine animals in a new underwater dataset with varying visibility. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 18–26, 2019.

[20] Meng-Che Chuang, Jenq-Neng Hwang, and Kresimir Williams. Supervised and unsupervised feature extraction methods for underwater fish species recognition. In *2014 ICPR Workshop on Computer Vision for Analysis of Underwater Imagery*, pages 33–40. IEEE, 2014.

[21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[23] Vojtěch Čermák, Lukas Picek, Lukáš Adam, and Kostas Papafitsoros. Wildlifedatasets: An open-source toolkit for animal re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5953–5963, 2024.

[24] Faizan Farooq Khan, Xiang Li, Andrew J Temple, and Mohamed Elhoseiny. Fishnet: A large-scale dataset and benchmark for fish recognition, detection, and functional trait prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20496–20506, 2023.

[25] WekaDeeplearning4j. Imagenet 1000 class list, 2025.

[26] Using deep learning to recognize individual animals in images.

[27] Steve Astels Leland McInnes, John Healy. How hdbscan works, 2016.

[28] J. Laaksonen and E. Oja. Classification with learning k-nearest neighbors. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 3, pages 1480–1483 vol.3, 1996.

[29] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17(134):1–32, 2016.

[30] Kenneth Leung. Micro, macro weighted averages of f1 score, clearly explained, Jan 4, 2022.

[31] OpenCV. Color conversions - documentation, 2025.

[32] OpenCV. Histograms - 2: Histogram equalization - documentation, 2025.