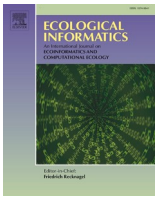
Ecological Informatics 81 (2024) 102619

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf



Machine learning for non-experts: A more accessible and simpler approach to automatic benthic habitat classification

Chloe A. Game [a,b,*], Michael B. Thompson [c], Graham D. Finlayson [b]

[a] Gardline Ltd., Great Yarmouth NR31 0NN, UK
[b] School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, UK
[c] Mott Macdonald Ltd, Norwich, NR1 1UA, UK

ARTICLE INFO

Keywords:
Machine learning
Computer vision
Convolutional neural network
Support vector machine
User-friendly
Benthic habitats

ABSTRACT

Automating identification of benthic habitats from imagery, with Machine Learning (ML), is necessary to contribute efficiently and effectively to marine spatial planning. A promising method is to adapt pre-trained general convolutional neural networks (CNNs) to a new classification task (transfer learning). However, this is often inaccessible to a non-specialist, requiring large investments in computational resources and time (for user comprehension and model training). In this paper, we demonstrate a simpler transfer learning framework for classifying broad deep-sea benthic habitats. Specifically, we take an 'off-the-shelf' CNN (VGG16) and use it to extract features (pixel patterns) from benthic images (without further training). The default outputs of VGG16 are then fed in to a Support Vector Machine (SVM), a classical and simpler method than deep networks. For comparison, we also train the remaining classification layers of VGG16 using stochastic gradient descent. The discriminative power of these approaches is demonstrated on three benthic datasets (574–8353 images) from Norwegian waters; each using a unique imaging platform. Benthic habitats are broadly classified as Soft Substrate (sands, muds), Hard Substrate (gravels, cobbles and boulders) and Reef (*Desmophyllum pertusum*). We found that the relatively simplicity of the SVM classifier did not compromise performance. Results were competitive with the CNN classifier and consistently high, with test accuracy ranging from 0.87 to 0.95 (average = 0.9 (±0.04)) across datasets, somewhat increasing with dataset size. Impressively, these results were achieved 2.4–5× faster than CNN training and had significantly less dependency on high-specification hardware. Our suggested approach maximises conceptual and practical simplicity, representing a realistic baseline for novice users when approaching benthic habitat classification. This method has wide potential. It allows automated image grouping to aid annotation or further model selection, as well as screening of old-datasets. It is especially suited to offshore scenarios as it can provide quick, albeit crude, insights into habitat presence, allowing adaptation of sampling protocols in near real-time.

1. Introduction

Benthic habitats may consist of multiple components: substrate, species and/or communities, their environmental tolerances and preferences (Davies et al., 2004; Diaz et al., 2004). They often act as simplified but powerful proxies of biodiversity, by allowing inference of occurring organisms through known ecological associations. Thus creation of extensive and accurate benthic habitat maps (Baker and Harris, 2020; Cogan et al., 2009; Harris and Baker, 2020) is a crucial component of marine spatial planning and conservation goals (European Parliament, 2008; Howell et al., 2020; Sala et al., 2018; United Nations, 2018; United Nations General Assembly, 2015) to mitigate anthropogenic impacts on the marine environment. Such maps establish baselines and support monitoring of impacts and recovery, which can allow proactive decision making. Data to support such endeavours requires processing of optical imagery which has a number of issues (1) annotation of resulting imagery is often inconsistent and error prone due to observer bias, fatigue, distraction and short-term memory limitations (Culverhouse et al., 2014; Durden et al., 2016), (2) it is costly (in the absence of volunteers) and (3) labour-intensive. This reality is particularly realised with Autonomous Underwater Vehicle (AUV) usage, in which one survey (~50 h) can produce over 170,000 images (Wynn et al., 2014) for

* Corresponding author at: Gardline Ltd., Great Yarmouth NR31 0NN, UK.
E-mail address: chloe.game@gardline.com (C.A. Game).

https://doi.org/10.1016/j.ecoinf.2024.102619
Received 12 January 2024; Received in revised form 26 April 2024; Accepted 26 April 2024
Available online 10 May 2024

example. Machine learning (ML) solutions, to automate the identification and enumeration of image features (objects, pixel patterns), are therefore generally believed to be essential to alleviating this problem. Their use has become increasing popular in ecology over the last few decades (Christin and Hervet, 2019; Rubbens et al., 2023; Weinstein, 2018), with Convolutional Neural Networks (CNNs), a deep ML algorithm, showing particular promise (Alzubaidi et al., 2021; Goodfellow et al., 2016; He et al., 2016; Huang et al., 2017; Krizhevsky et al., 2012; Lecun et al., 1998; LeCun et al., 2015; Simonyan and Zisserman, 2015). There have been a number of studies that have successfully used deep learning to localise (detection) and determine the spatial extent (segmentation) of objects in benthic imagery (Katija et al., 2022; Liu et al., 2023; Liu and Wang, 2021; Piechaud and Howell, 2022; Song et al., 2019; Zhang et al., 2022). However, in this work we focus on the foundation of each of these ML tasks, classification.

Image classification is simply the assignment of a label, text or numerical category, to an image based on visual patterns. Whether undertaken by a human or a computer, the task is *broadly* the same. In marine applications, many studies have demonstrated the possibility of using CNNs to classify benthic taxa or substratum in optical imagery (Abad-Uribarren et al., 2022; Downie et al., 2022; Durden et al., 2021; Jackett et al., 2023; Kandimalla et al., 2022; Langenkämper et al., 2019a; Langenkämper et al., 2020; Marburg and Bigham, 2016; Piechaud et al., 2019; Piechaud and Howell, 2022; Vega et al., 2024).

However, there is currently less literature employing them at the habitat level (Jackett et al., 2023; Mahmood et al., 2020a; Mahmood et al., 2020b; Mohamed et al., 2020; Rao et al., 2017; Vega et al., 2024; Yamada et al., 2022; Yamada et al., 2023) with many focusing on the same benchmark datasets, Benthoz15 (Bewley et al., 2015) and Tasmania (Williams et al., 2012), in shallow waters. Shallow-water corals are a particularly common benthic target for CNN classification (Beijbom et al., 2016; Gómez-Ríos et al., 2019a; Gómez-Ríos et al., 2019b; Mahmood et al., 2016; Mahmood et al., 2019), largely thanks to the availability of open-source datasets such as MLC (Beijbom et al., 2012), EILAT and RSMAS. In some cases, reef-forming corals can be considered to occur as both a species and a habitat (Howell et al., 2011). However, the examples in these datasets are typically highly-zoomed images of coral texture and thus do not represent well the broader structure and contextual appearance of the habitat at the captured image resolution.

Without close collaboration with experts, training a CNN is highly intimidating, and often inaccessible, to a non-specialist such as marine ecologists (Crosby et al., 2023; Tuia et al., 2022). To use them requires a large investment in time both to understand the underlying theory and to learn how they are implemented and deployed in the field. If nothing else they require a knowledge of programming in languages such as Python and Matlab. Additionally, to deploy CNNs efficiently there is a need to use significant computing resources (often very high specification machines) and a need for'Big Data' (large numbers of annotated images). High compute power and expensive data collection may individually or together present barriers to using CNNs. Effort has been made to incorporate automatic classification tools into annotation softwares such as BIIGLE (Langenkämper et al., 2017), VIAME (Dawkins et al., 2017) and CoralNet (Chen et al., 2021). However these may be unsuitable for those requiring more flexibility in ML approaches or those wishing to integrate ML functionality into their custom annotation programs. They are also tailored for certain automation tasks.

For these reasons, in this paper, we propose a simpler approach that will be more easily understood and used by novice end-users. Specifically, we use an open-source 'off-the-shelf' pre-trained CNN to extract features (pixel patterns) from contextually-representative benthic images. We then train a shallow (non deep-net) ML classifier to classify these deep features. There are a range of classifiers that could be used for this task such as Random Forests (Breiman, 2001), K-Nearest Neighbour (KNN), Logistic Regression and Naive Bayes, however in this work we focus on Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Cristianini and Shawe-Taylor, 2000). SVMs have been shown to pair well with 'off-the-shelf' CNN features in benthic image applications (Mahmood et al., 2016; Mahmood et al., 2019; Mahmood et al., 2020a; Mahmood et al., 2020b; Mohamed et al., 2020), as well as more generally (Azizpour et al., 2015; Razavian et al., 2014; Salman et al., 2016). They are a classical method, well documented and offer a good trade-off in terms of complexity, performance, computational demand and time. SVMs are boundary classifiers; separating data points, either linearly or non-linearly, into groups for classification. As a result, they can be used with some flexibility to choose the best classification for your data. Many studies using an SVM present only one method (e.g. linear), thus in the interest of comprehensiveness we compare multiple.

The hybrid CNN & SVM approach is only really of interest if it provides good and competitive results to deep-learning (CNN). Thus, for comparison we also retrain the CNN on our habitat classification task (transfer learning). For our CNN, we use VGG16 (Simonyan and Zisserman, 2015), pre-trained on a large and unrelated dataset, ImageNet (Deng et al., 2009). This allows us to extract more general features for classification such as edges, lines, corners and simple textures in our own images (Mahmood et al., 2016; Mahmood et al., 2019; Razavian et al., 2014; Yosinski et al., 2014). It may also extract more complex geometric and textural qualities of seabed features, due to the presence of underwater classes in the ImageNet dataset that share similarities to our benthic habitat classes i.e. corals and sandbar. There are several reasons for choosing VGG16 as a basis for transfer learning. VGG is one of the most implemented algorithms for image classification and although several years old remains highly popular, with high performance across diverse image applications applications (Abosaq et al., 2023; Althubiti et al., 2022; Kaur and Gandhi, 2019; Krishnaswamy Rangarajan and Purushothaman, 2020; Yang et al., 2021a; Yang et al., 2021b). More importantly, it also has a record of good performance across marine classification tasks (González-Rivero et al., 2020; Kloster et al., 2020; Lumini and Nanni, 2019; Mahmood et al., 2019; Mahmood et al., 2020a; Zhang et al., 2019). Preliminary comparisons of VGG16 to other models (AlexNet (Krizhevsky et al., 2012), ResNet18 & ResNet50 (He et al., 2016) and VGG19 (Simonyan and Zisserman, 2015)) found VGG16 to produce deep features that were more accurately classified by an SVM. For inexperienced users, our target audience in this application, a model architecture such as VGG16 may be preferential over newer state-of-the-art model architectures. This is due to its inclusion in more accessible platforms/frameworks such as PyTorch (PyTorch, 2023) and TensorFlow (Abadi et al., 2016), its extensive guidance materials for implementation and well-documented high performance.

This paper provides both a quick primer for novice users, facilitating comprehension and implementation of these approaches, and serves as proof of concept. We make the following contributions:

- We show that both deep and shallow learning can lessen the image analysis bottleneck of a highly important classification task, benthic (deep-sea) habitats, which is poorly represented in machine learning studies.
- We demonstrate an automation pipeline that leverages the power of deep learning (VGG16) and transfer learning but is made simpler and more accessible for inexperienced users with the use of a shallow SVM classifier.
- We compare both linear and non-linear SVM classifiers and show the benefit of hyperparameter tuning on performance.
- Our selected hybrid CNN & SVM approach is shown to be competitive with deep learning in terms of performance, time and ease of implementation.
- We validate the generality of the method across multiple datasets that vary in size, imaging platform and geographic region.
- A visual analysis demonstrated that inconsistent appearance of habitats (including novel features) and overlapping class characteristics can present challenges for automated classification.
- Lastly we provide recommendations for improving performance, albeit at the expense of complexity.

Table 1
Benthic image datasets used in this study.

ID	Platform	Components	Image Specifications	Location
1	ROV	Imenco Tiger Shark 14mpx with external flash (Lantern Shark), 10 × ROS (MV-4000) and 4 × Innova Gas lights.	8353 RGB images (4320 × 3240, *.jpeg*)	Norwegian Sea
2	ROV	Konsberg/Simrad (OE14–208) 5.0mpx, 1 × forward-facing strobe, 2 × fixed & 2 × mobile LED lamps.	1240 RGB images (2592 × 1944, *.jpeg*)	Norwegian Sea
3	Drop camera	Konsberg/Simrad (OE14–208) 5.0mpx, 1 × forward-facing strobe, 2 × fixed & 2 × mobile LED lamps.	574 RGB images (2592 × 1944, *.jpeg*)	Norwegian Sea

Table 2
A comparison of the habitat classification system used to EUNIS (Level 2) habitats (European Environment Agency, 2022; Evans et al., 2016).

Broad habitat	Abbreviation	EUNIS (Level 2)	Included Sub-habitats
Soft Substrate	Soft Sub.	ME5: Upper bathyal sand, ME6: Upper bathyal mud	Heavily bioturbated Soft Sub., Single sea pen, Sea pen community, Soft Sub. sponge community
Hard substrate	Hard Sub.	ME1: Upper bathyal rock	Gravel area, Scattered cobbles, Cobble and boulder area, Boulder area, Hard Sub. sponge community
Reef	Reef	ME2: Upper bathyal biogenic habitat	Coral rubble zone, Dead *Desmophyllum pertusum* reef framework, Live *Desmophyllum pertusum* reef

2. Materials and methods

2.1. Image datasets

Three datasets were used to explore the generality of the pipeline presented. Datasets span two types of imaging platform; ROVs and Drop Cameras, the specifications of which are listed in Table 1. Note that each platform has a unique camera and lighting configuration. Additionally, the datasets cover multiple geographic regions; 3 unique locations within the Norwegian Sea. Further geographic information cannot be reported due to commercial sensitivity.

Benthic images were collected and manually annotated by Gardline Ltd. The primary habitat of each image was categorized according to an in-house seabed classification guide, simplified in Table 2. For better contextualization of seabed classes, we also present the equivalent European Nature Information System (EUNIS) 2022 classification (European Environment Agency, 2022; Evans et al., 2016). For the purpose of this study, habitats have been grouped at a broad-scale (Table 2). We focus on three that are typically recorded in deep-water surveys: 1) Soft Substrate, 2) Hard Substrate and 3) Reef. For brevity, we refer to these as Soft Sub., Hard Sub. and Reef. Sub-habitats included in each of these broad categories can also be found in Table 2. Example images of the benthic habitats encountered are presented in Fig. 1.

The variety of habitats, multitude of imaging platforms and geographic separation of these datasets ensure that associated analyses and findings will have a great analytical relevance for both the wider marine ecology and computer science communities.

2.2. Machine learning pipeline

In this section we present two approaches to automate classification of benthic habitats from optical imagery. Each approach consists of two main phases: feature extraction and classification. Simply put, feature extraction learns to find and highlight patterns in pixel information (features). The classification phase then learns to link these features to one of the three habitat classes. The most complex of these methods uses the full CNN (VGG16) model (architecture) for both feature extraction and classification. The alternative, and our recommended process, uses only the feature extraction components of VGG16 paired with an SVM classifier. Thus the two approaches in this work differ only with respect to classification. For brevity, these methods are henceforth referred to as **CNN** and **CNN + SVM**, respectively.

How these model architectures explicitly work is beyond the scope of this paper. However, please refer to Section 2.2.1 for a brief background on the two models. Specific methodological details and the full pipeline are provided in Sections 2.2.2 to 2.2.6 and in Table 3, to guide their implementation. We also present a graphical representation of our work flow in Fig. 2.

All analysis was conducted in Python, largely using the libraries *scikit-learn* for machine learning (i.e. SVMs) and *torch*, the basis of the

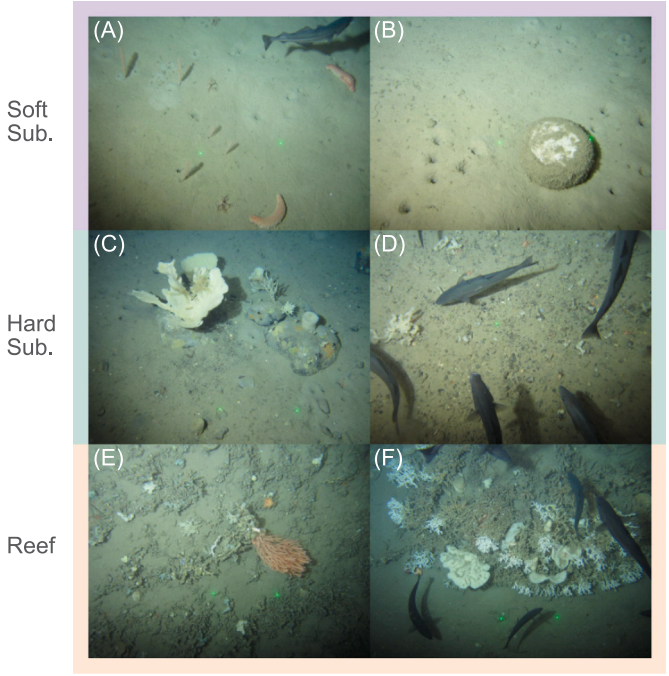


Fig. 1. Examples of the benthic habitats encountered within this study: (A) & (B) depict Soft Sub., (C) & (D) Hard Sub. and (E) & (F) Reef.

Table 3
Required steps in our ML workflows.

Data Preparation
• Organise images into class folder structure
• Separate image dataset into training and testing
• Resize imagery for CNN *i.e. 224×224*
• Normalize images to train dataset (ImageNet)
Model Preparation
• Load pre-trained VGG16 (with ImageNet weights)
• Freeze all layers
• Duplicate the network
For **CNN+SVM**:
• Remove all CNN layers beyond FC1
• Initialize SVM with default parameters
For **CNN**:
• Unfreeze (replace) FC2 & FC3 in classifer
• Set FC3 outputs to the number of classes *i.e. 3*
• Set HP *i.e. learning rate, optimizer, loss function*
Feature Extraction
• Set batch size for feature extractor
• Extract features from training and testing set
Classification
For **CNN+SVM**:
• Optimize SVM hyperparameters with grid-search
• Train SVM with optimal choice
For **CNN**:
• Optimize learning rate during training
Performance Evaluation
• Assess training performance
• Predict classes in test data
• Evaluate test performance

[a]Note that the order of steps may vary with ML frameworks (i.e. *pytorch, tensorflow*) or may be achievable simultaneously.
[b]Grey shading denotes model-specific steps.

Pytorch (Paszke et al., 2019) ML framework for deep learning (CNNs). These libraries also include tools for data pre-processing, model selection and evaluation. To keep our model training and analysis pipelines comparable, we use *skorch*, a *scikit-learn* compatible neural network library that wraps PyTorch. This allows the same *scikit-learn* training and evaluation procedure to be used for both models. *Skorch* is also helpful for end-users in CNN training, as it has a clear and simple interface. It only requires end-users to add the prepared datasets, model and specify the associated hyperparameters (Table 4). Documentation for the entire machine learning pipeline can be found at (PyTorch, 2023) for Pytorch (Paszke et al., 2019; scikit-learn, 2023) for *scikit-learn* and (skorch, 2022) for *skorch*. Commercial restrictions apply to the availability of data used in this work. However, links to public code examples of workflow stages (Table 3) can be found in Supplementary Table 1. This work was supported by an NVIDIA GeForce RTX 2080 SUPER Graphical Processing Unit (GPU) with 8GB VRAM and an Intel Core i7–9700 CPU.

2.2.1. Background

CNNs extract features from imagery primarily using a process known as convolution (Alzubaidi et al., 2021; Goodfellow et al., 2016; LeCun et al., 2015); a sort of sliding filter (matrix) that transforms pixel values, see Fig. 2. These image features may correspond to low-level objects such as edges, circles and lines up to high-level features such as sponge branches for example. Features extracted by the convolutional layers are then 'connected' (mapped) to a specified number of outputs. We refer to these layers as fully connected (FC). Similar to regression, FC layers are merely approximating functions, that best map every input value (feature) to each output. These functions can be thought of as complex fitted curves or hypersurfaces that are created by (1) linear transformations, multiplying features by weights and adding biases and (2) non-linear transformations, adjusting features with an activation function.

CNNs typically house multiple (>2) FC layers in succession. Although each of them perform the same function in a technical sense, it is helpful to think of the first FC layers as downsampling the convolved features to fewer outputs, or further clarifying feature patterns. Whereas the last FC layer can be thought of as the true classification or output layer, in which the number of outputs corresponds to the number of classes, in our case the 3 habitats. Here the output values (logits) are interpreted *like* a probability (their sum may be >1), showing which class the image features best correspond to. This is decided by taking the maximum output value, known as the *argmax*. Training the CNN will help to better match these mapped (predicted) outputs to the true classes of imagery passed through the network, ready for prediction on a novel test dataset.

SVMs do not possess any feature extraction capabilities. Instead feature data, associated with each class, can be provided as inputs to an SVM. It will then find the best boundary, or hyperplane, between these data points that enables class distinction. This separation occurs in a feature space of n-dimensions, where *n*=number of image features, and can be both linear and non-linear, depending on the kernel (function) used. SVMs classify data points simply by observing where they lie with respect to the hyperplane, see Fig. 3. Unlike a CNN, the output is therefore a predicted class rather than a *probability* that it is either of the classes. The SVM uses the data points closest to the hyperplane, known as support vectors, to guide hyperplane placement. The support vectors are the hardest points to classify, given the potentially close proximity of support vectors of each class. Optimal placement is therefore found by maximizing the distance between the support vectors of each class and the hyperplane (known as the margin) such that the misclassification rate is minimized. This is why SVMs are referred to as maximum-margin classifier; they find the hyperplane that is equidistant between the two classes. Using only the support vectors, and thus a subset of the data, to learn where to place the hyperplane is very memory efficient, compared to a CNN which uses all data in training.

As SVMs were created for binary classification problems, strategies exist that enable use of an SVM with multi-class problems (>2 classes), as is the case with our data. We use a typical approach called One-Vs-One. This splits the dataset into multiple binary classification problems that are assessed per each pair of classes. Compiling the classifications of all binary SVMs allows a final classification to be made for each data point, based on the class that received the most votes, see Figs. 2 & 4.

2.2.2. Data preparation

Pre-trained CNNs are designed to expect images in a certain format before feature extraction (or classification). For VGG16 (and other networks trained on ImageNet (He et al., 2016; Howard et al., 2017; Russakovsky et al., 2015)), RGB images must be 224 × 224 pixels. We therefore resize images to 224 pixels along the x-axis, preserving their aspect ratio. We then crop the center of images such that they are square. Following standard practice, RGB values were also normalized (centered and scaled) to the training dataset (ImageNet), see Table 3. The images in each dataset were split into 80% training (including 5-fold cross-validation) and 20% testing subsets. Splits were stratified to preserve the class-ratio.

2.2.3. Model preparation

The VGG16 network was sourced from the *torch* library and all layers frozen, preventing any further training (updates to model parameters), see Table 3. We then duplicated this network to provide a foundation for each modelling approach. For our **CNN + SVM** modelling approach we kept the architecture up to the first FC layer (FC1), creating a feature extractor (Fig. 2). We then paired it with an SVM, sourced from the *scikit-learn* library. We evaluate two types of SVM: a linear SVM and a non-linear SVM, known as a Radial Basis Function (RBF) (Fig. 3). An RBF SVM is a good default choice, as it can find both a linear and non-linear hyperplanes at high dimensions.

For the **CNN** approach however, we use the full VGG16 network, leaving the feature extractor and classifier intact. In its frozen state the

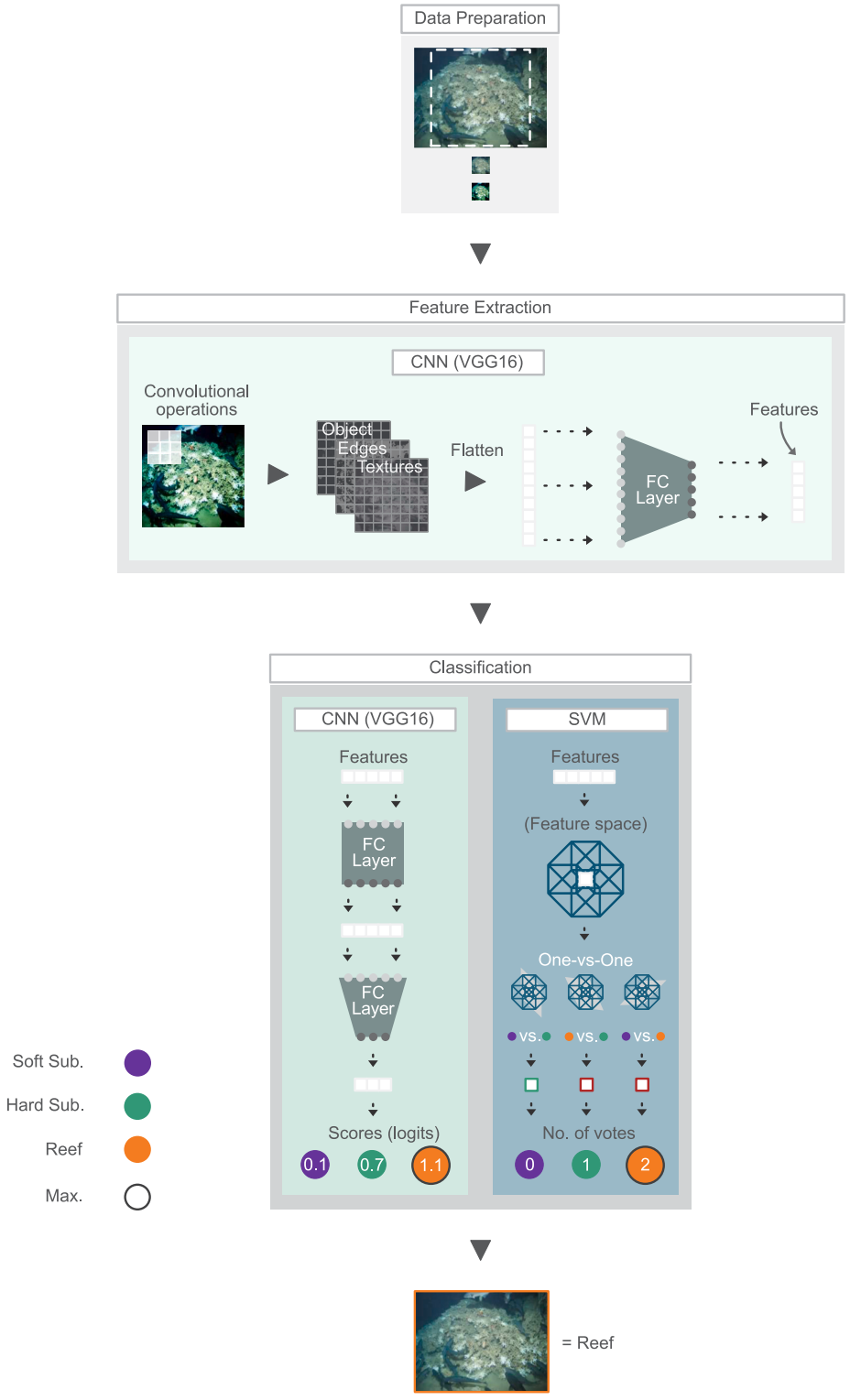


Fig. 2. Infographic of the ML workflows (**CNN** & **CNN** + **SVM**) used in this study.

VGG16 network cannot be trained, we therefore unfroze (replaced) the final FC layers (FC2 & FC3) to re-initialize the weights for training. Training only these FC layers enables comparison to the **CNN** + **SVM** approach in which they have been replaced with an SVM classifier. We also reduced the number of output nodes in FC3 from 1000 (number of ImageNet classes) to 3, to prepare the **CNN** to classify the 3 habitat classes.

2.2.4. *Feature extraction & visualisation*

In each of our ML pipelines, every image that passes through the VGG16 feature extractor (Fig. 2) results in a matrix of 1×4096 features. These are then passed as inputs to the classifier. This happens automatically in the **CNN** approach, as the classification (FC) layers are still present in the architecture.

These extracted 'deep' features are numerous and difficult to interpret. Therefore, before undertaking any classification of the extracted

Table 4
Model hyperparameter glossary for CNN & SVM training.

CNN:	
Batch size	The number of images you send to the model in each iteration. *Model parameters are updated after each batch during training.*
Epochs	How many times you pass the full image dataset through the model.
Loss function	The error metric that you wish to minimize. e.g. *Cross entropy loss for multi-class classification.*
Learning rate	A small number $(0, 1]$ that determines the amount to alter parameters during training with respect to the loss. *Also known as the step size.*
Optimizer	An algorithm that modifies CNN parameters according to a particular strategy to minimize the loss. e.g. *the Adam optimizer sets the learning rate adaptively for faster and more efficient training.*
SVM:	
C	A regularization parameter that offers a trade-off between the maximum-margin and misclassification rate. e.g. *A large C enforces a small margin hyper-plane maximizing classification accuracy.*
γ	A value to determine the distance over which support vectors influence the hyperplane. e.g. *A high γ considers only points that are close to each other and causes the decision-boundary to be highly curved.*

N.B. This list of hyperparameters is not exhaustive. It simply covers hyperparameters relevant to our approaches.

high-dimensional feature space, we reduced the number of features (or dimensions) to ≤3 for visualisation purposes only. Although the features themselves remained cryptic, this allowed us to visually assess any structure or clustering in the data that would enable good separation of classes and thus a high performing model. For this task we use a Pairwise Controlled Manifold Approximation (PaCMAP) (Wang et al., 2021) from the library *pacmap*. This technique uses euclidean distance to quickly, and simply, find a low-dimensional representation of the complex feature space (that is structurally most-similar). It is straightforward to use and has proven to accurately capture data distributions (Wang et al., 2021).

2.2.5. Classification

Each ML approach requires hyperparameters to classify imagery, which when optimized during training can increase model performance, see Table 4 for a hyperparameter glossary. Given the computational efficiency of the SVMs and the few hyperparameters required, each of these can be optimized simply and relatively quickly (subject to dataset size) during a k-fold $(k = 5)$ cross-validated fine grid-search on the training data. For our **CNN + SVM** method, we followed hyperparameter recommendations by (Hsu et al., 2016), authors of the *LIBSVM* library (Chang and Lin, 2011). For our non-linear RBF SVM we searched hyperparameters $C = 2^3, 2^{3.25}, \ldots, 2^7$ and $\gamma = 2^{-15}, 2^{-13}$ & 2^{-11}. For the linear SVM, we used the same hyperparameter search for its sole parameter C. We also looked at the RBF and linear SVM with

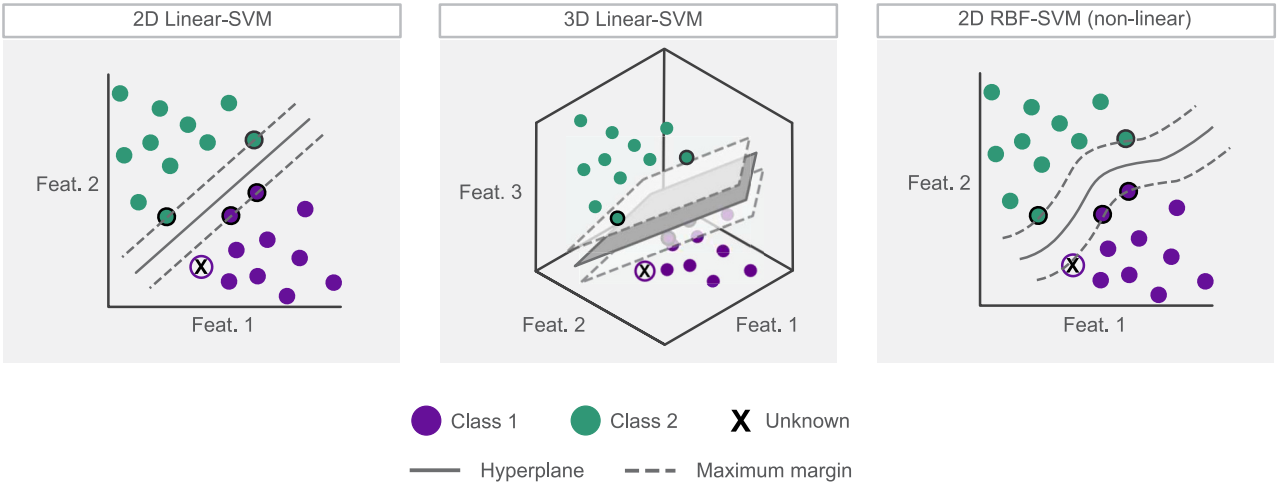


Fig. 3. A diagram of various support vector machines.

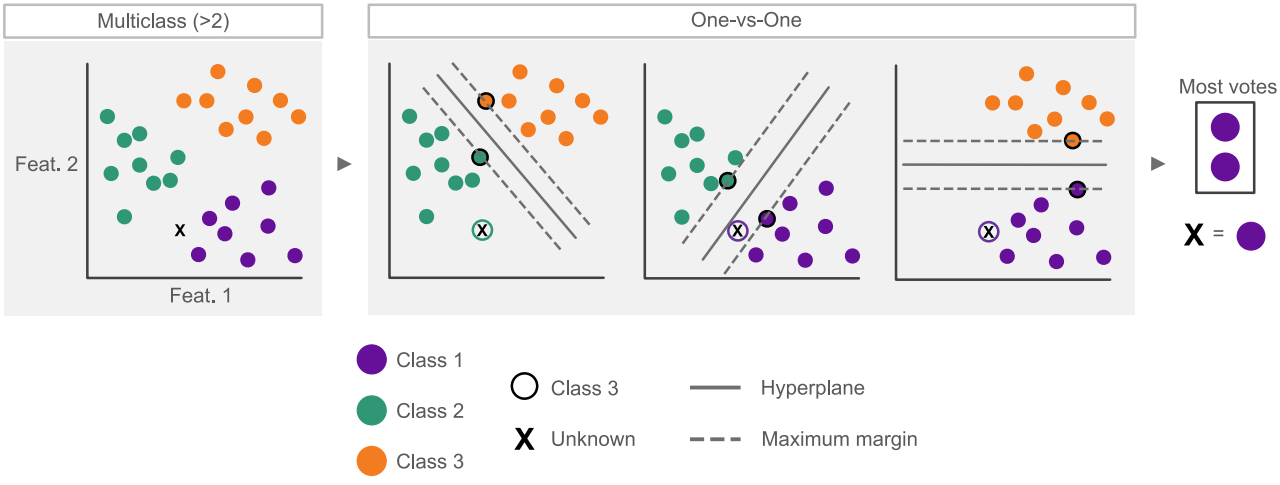


Fig. 4. Multi-class support vector machines: a diagram of the one vs. one strategy.

default *scikit-learn* values, $C = 1$ and $\gamma = 1/(f \times Var(F))$, where f is the number of features, F the features and Var the variance. Each fold in the training dataset is used once as a validation set, while the $k-1$ remaining folds form the new training set. Hyperparameter combinations that received the highest average accuracy (proportion of correct classifications), across validation folds were selected.

For the CNN, The large number of hyperparameters (Table 4) and high computational demand, mean that an exhaustive grid-search is inappropriate. Instead our preliminary work showed that common default parameters, were suitable for our data. These include a batch-size of 32 images and a cross-entropy loss function. We also used the Adam learning rate optimizer (Kingma and Ba, 2015), which automatically adjusted our initial learning rate of 1e-03 during training in a way that improved performance. Adam is computationally efficient and straight-forward to use. In preliminary work, each model was set to train for 100 epochs maximum. However for later time-saving and better automation, we enabled early stopping if the validation error (loss) did not reduce for 10 epochs. This identified a suitable number of epochs for each dataset: 14, 14 and 23 epochs for Datasets 1, 2 and 3, respectively. As with SVM training, we conducted the CNN training protocol using 5-fold cross-validation of the training set; the same k-folds as the SVM training.

Following cross-validation of each modelling approach, mean validation accuracy across k-folds was determined. We then undertook final training on the entire training dataset. In the case of the SVM, final training was undertaken with the best performing hyperparameter combinations.

2.2.6. Performance evaluation

We used a number of metrics to evaluate model performance. Each are scored between 0 & 1, with optimal performance reached at 1. Perhaps the simplest of these measures is accuracy, which indicates the proportion of images that the model classifies correctly. However, to account for performance within classes we also calculated the recall, precision and F_1 score for each class. Recall, or sensitivity, refers to the proportion of images of each class that were correctly classified. Precision however determines the accuracy of the predictions themselves; it measures the proportion of images that were assigned a correct class when classified. For each habitat class c, recall and precision are calculated as:

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \tag{1}$$

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \tag{2}$$

where TP refers to the number of true positives (those correctly classified as class c), FP the false positives (those that are incorrectly classified as class c) and FN the false negatives (those of class c that are incorrectly classified). For a balanced view of model performance we also calculated the F_1 score, a harmonic mean of the precision and recall. This gives an idea of how well the model recognises images of each class and distinguishes between images of other classes. For each class, F_1 is calculated as:

$$F_{1_c} = \frac{2^*(Recall_c{}^*Precision_c)}{(Recall_c + Precision_c)} \tag{3}$$

For each of the class metrics used (recall, precision and F_1), we present an average across classes, also known as the macro-average, alongside their 95% confidence intervals (CI). The macro-average for each metric m is calculated simply by:

$$Macro - average_m = \frac{\sum_{c=1}^{C} m_c}{C} \tag{4}$$

where C is the total number of classes. The macro-average weights class importance equally, irrespective of the number of images associated (instances of each class), and therefore represents model performance more reliably. This is particularly useful for our application, given that the deep seafloor is largely a soft-sediment habitat with intermittent hard-substrate and reef and thus a class imbalance is typically present in image surveys.

3. Results

3.1. Interpretability of feature space

The VGG16 network activations resulted in a feature space with conspicuous clustering of each habitat, see Fig. 5. This suggests the general features provided by the VGG16 network are suitable for classifying benthic habitats, though the extent to which may be variable across the Datasets used and classes encountered. Across all datasets, the PaCMAP dimensionality reduction showed the strongest partition of Soft Sub. & Hard Sub. Features associated with Reef habitat were also clearly separable from Soft Sub. habitats. However, they do exhibit a degree of overlap with Hard Sub., in 3 dimensions. This may indicate higher similarity with this habitat, with respect to the general features extracted.

Of the datasets, habitat clusters were particularly notable within Dataset 1, at both a higher (3D) in Fig. 5 and lower (2D) dimension in 6a. In Fig. 6b, image thumbnails demonstrate the visual transition between image characteristics of each class for better comprehension of the

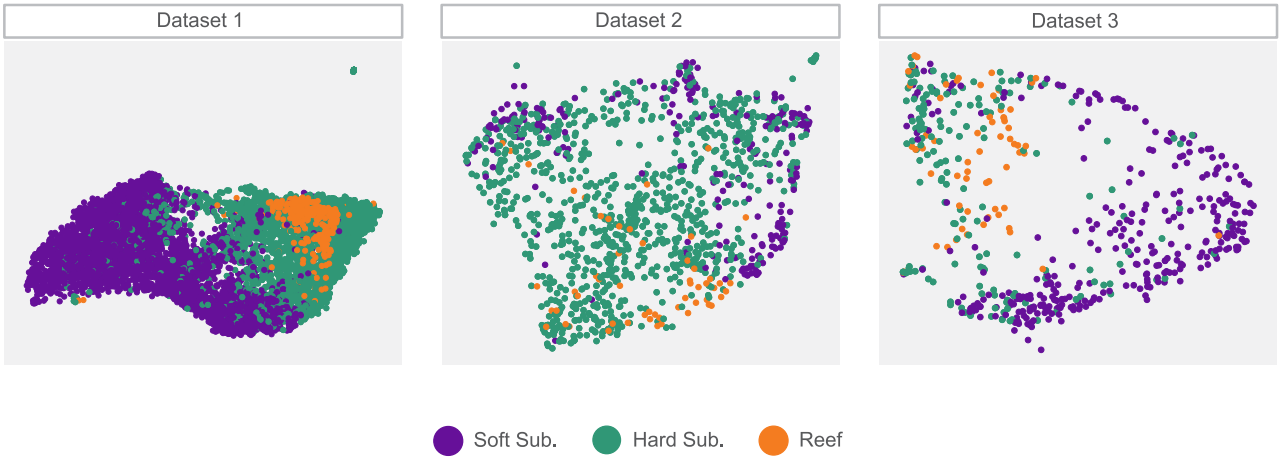


Fig. 5. 2-Dimensional visualisation of each datasets 3-Dimensional feature space created using PaCMAP.

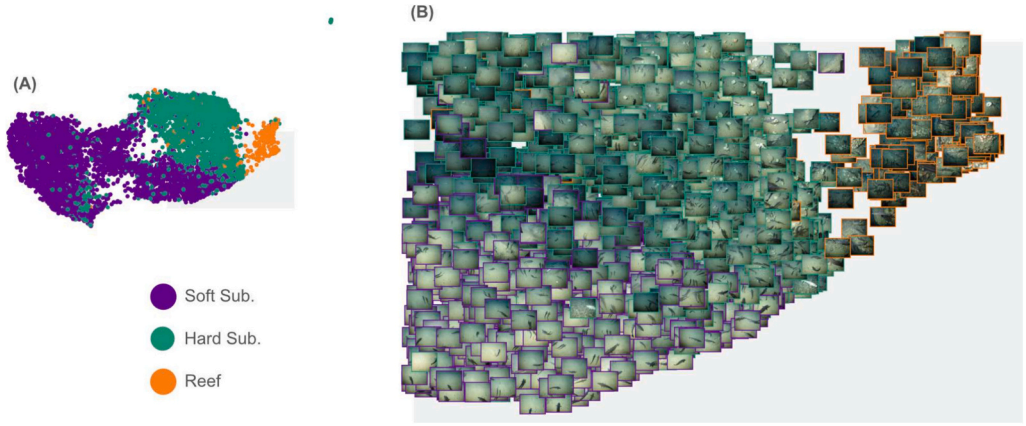


Fig. 6. PaCMAP 2-Dimensional visualisation of the Dataset 1 (Table 1) feature space: (A) 2D point cloud and (B) zoomed region in which points are replaced by image thumbnails for better conceptualization of feature variance.

Table 5
Classifier training performance: with tuned hyperparameters, mean cross-validation (CV) and final training accuracy.

				Accuracy	
Set	Images	Classifier	Hyperparameters	Mean CV	Train
1	6682	SVM:Linear$_d$	–	0.93 (± 0.003)	**1.00**
		SVM:Linear	$C = 2^{3.0}$	0.93 (± 0.003)	**1.00**
		SVM:RBF$_d$	–	0.95 (± 0.003)	0.97
		SVM:RBF	$C = 2^{3.0}, \gamma = 2^{-15.0}$	**0.96 (± 0.003)**	0.98
		CNN	$lr = 0.001$	0.92 (± 0.023)	0.98
2	992	SVM:Linear$_d$	–	0.87 (± 0.010)	**1.00**
		SVM:Linear	$C = 2^{3.0}$	0.87 (± 0.010)	**1.00**
		SVM:RBF$_d$	–	0.87 (± 0.018)	0.91
		SVM:RBF	$C = 2^{3.75}, \gamma = 2^{-15.0}$	**0.91 (± 0.016)**	0.97
		CNN	$lr = 0.001$	0.90 (± 0.015)	0.97
3	459	SVM:Linear$_d$	–	0.83 (± 0.021)	**1.00**
		SVM:Linear	$C = 2^{3.0}$	0.83 (± 0.021)	**1.00**
		SVM:RBF$_d$	–	0.78 (± 0.016)	0.84
		SVM:RBF	$C = 2^{5.25}, \gamma = 2^{-15.0}$	**0.86 (± 0.020)**	0.95
		CNN	$lr = 0.001$	0.82 (± 0.020)	0.95

[1]Subscript *d* denotes SVM with default hyperparameters.
[2]95% confidence intervals are shown in brackets. Bold font dictates best results.

feature space. A gradient of seabed complexity is clearly evident from the bottom left to the top right of the frame. The proximity of images, within and between classes, can be seen to not only correlate well with seabed characteristics but also shows some clustering based on illumination patterns.

3.2. Classification performance

We find that general features automatically extracted by the VGG16 network enabled automatic classification of benthic habitats with few errors. Of the SVMs compared, we found a non-linear RBF kernel was best suited across datasets, factoring in both validation and test accuracies. Performance was also improved with the optimisation of hyperparameters. In Table 5 we list final training accuracy and mean accuracy across the 5-fold cross-validation (following hyperparameter tuning), for each dataset and SVM approach. Training accuracy was always highest for the linear SVMs, regardless of whether the hyperparameters were tuned, however the comparative performance across validation sets was almost always lower, on average by 2.4%, than when employing a non-linear SVM. With the addition of hyperparameter tuning, mean validation accuracy was always highest with an RBF SVM.

In Table 6 we show the final test accuracy as well as the class-averaged Recall, Precision and F_1 scores. Overall accuracy on the test set was similar to the mean validation accuracy. In each dataset, best performance was found with a hyperparameter-tuned RBF SVM, with an average improvement of 3.3% over Linear SVMs and 1.6% over default RBF parameters. Performance was more variable across the class-averaged metrics. Linear SVM performance was fairly balanced across metrics. RBFs were found to be more precise in their predictions by comparison, sometimes at the expense of recall. For interpreting marine imagery, it is more important that predictions are correct (i.e. high

Table 6
Classifier testing performance: with overall test accuracy and class-averaged metrics (Recall, Precision and F_1 Score).

Set	Images	Classifier	Accuracy	Recall	Precision	F_1 Score
		SVM:Linear$_d$	0.93	0.93 (± 0.025)	0.93 (± 0.022)	0.93 (± 0.023)
		SVM:Linear	0.93	0.93 (± 0.025)	0.93 (± 0.022)	0.93 (± 0.023)
1	1671	SVM:RBF$_d$	**0.95**	0.92 (± 0.055)	**0.97 (± 0.026)**	0.94 (± 0.017)
		SVM:RBF	**0.95**	0.93 (± 0.040)	0.96 (± 0.016)	**0.95 (± 0.014)**
		CNN	0.95	**0.94 (± 0.018)**	0.93 (± 0.029)	0.93 (± 0.023)
		SVM:Linear$_d$	0.86	0.75 (± 0.140)	0.76 (± 0.123)	0.75 (± 0.130)
		SVM:Linear	0.86	0.75 (± 0.140)	0.76 (± 0.123)	0.75 (± 0.130)
2	248	SVM:RBF$_d$	0.88	0.63 (± 0.370)	**0.90 (± 0.098)**	0.67 (± 0.297)
		SVM:RBF	**0.89**	0.80 (± 0.106)	0.83 (± 0.070)	**0.82 (± 0.088)**
		CNN	0.88	**0.87 (± 0.039)**	0.78 (± 0.145)	**0.82 (± 0.085)**
		SVM:Linear$_d$	0.82	0.82 (± 0.041)	0.82 (± 0.127)	**0.82 (± 0.085)**
		SVM:Linear	0.82	0.82 (± 0.041)	0.82 (± 0.127)	**0.82 (± 0.085)**
3	115	SVM:RBF$_d$	0.83	0.67 (± 0.315)	**0.86 (± 0.126)**	0.70 (± 0.207)
		SVM:RBF	**0.87**	0.80 (± 0.138)	0.84 (± 0.049)	**0.82 (± 0.095)**
		CNN	0.84	**0.83 (± 0.168)**	0.80 (± 0.085)	0.80 (± 0.099)

[1]Subscript *d* denotes SVM with default hyperparameters.
[2]95% confidence intervals are shown in brackets. Bold font dictates best results.

precision) than identifying all images of a class (i.e. high recall). These results therefore suggest that a good SVM approach for our datasets is a non-linear RBF kernel. We note that highest RBF precision across datasets is found using default hyperparameters, however the recall (for dataset 2 & 3) is extremely low by comparison. By tuning RBF hyperparameters we yield higher performance than a Linear SVM and do not compromise the recall as much. For the remainder of this section, we compare the SVM classification performance to the CNN. Note that when referring to the **CNN + SVM** henceforth, we refer explicitly to this *best* SVM case, the hyperparameter-tuned RBF.

As with the SVM, the CNN classified general deep features well. Across both classification methods, training and validation accuracy were high, ranging from 0.95 to 0.98 ($\mu = 0.97 \pm 0.01$) and 0.82–0.96 ($\mu = 0.9 \pm 0.04$) respectively, see Table 5. Here, μ indicates the mean and $\pm$ the 95% CI. Final test accuracy, detailed in Table 6, was comparable to mean validation accuracy. Test accuracy even marginally exceeded validation performance (by 2–3%) in Datasets 1 & 2 when using a CNN and by 1% in Dataset 3 when using an SVM classifier. Across all datasets, classifier test accuracy ranged between 0.84 and 0.95 ($\mu = 0.9 \pm 0.03$). We find that in general the simpler classification approach, **CNN + SVM**, competes well with its complex counterpart, **CNN**, increasing accuracy up to 4% across all test and validation sets.

Compared to test accuracy, mean F_1 score was more varied across datasets, ranging between 0.8 and 0.95 ($\mu = 0.86 \pm 0.05$) across both methods. The lowest values were associated with CNN classification (in Dataset 3). SVM mean F_1 score was always >0.82; either matching or exceeded CNN performance between up to 2%. Of the components that contributed to the mean F_1 score, namely recall and precision, precision was always greatest for the SVM classifiers, and higher than their scores for recall. By comparison recall was always highest for CNN classifiers and lower than their scores for precision. This indicates the different priorities of each classification approach. The SVM is more conservative in its predictions whereas the CNN favours over-estimation - ensuring that more of each class is captured. Across both methods, mean precision scored between 0.78 and 0.96 ($\mu = 0.86 \pm 0.05$). Use of an SVM increased mean precision by between 3 and 5%. Whereas mean recall ranged between 0.8 and 0.94 ($\mu = 0.86 \pm 0.05$), with SVM decreasing performance by between 1 and 7%.

Considering the habitats individually, dataset-averaged performance metrics were relatively consistent across habitats and between classifiers (CNN or SVM) for each habitat, see Fig. 7. Regarding the consistency of performance scores across metrics, for each habitat and classifier, more variation was present in the classification of reef. Average performance metrics for Reef varied between 0.77 and 0.89 ($\mu = 0.84 \pm 0.05$) when using the CNN classifier and 0.77–0.87 ($\mu = 0.81 \pm 0.04$) with an SVM classifier. Soft Sub. and Hard Sub. performance was more consistent across metrics by comparison. For Soft Sub., average performance metrics ranging from 0.85 to 0.93 ($\mu = 0.9 \pm 0.03$) for CNN classification compared to 0.88–0.9 ($\mu = 0.89 \pm 0.01$) when using an SVM. Hard Sub. scored between 0.81 and 0.89 ($\mu = 0.84 \pm 0.03$) with a CNN and 0.86–0.9 ($\mu = 0.87 \pm 0.01$) with an SVM. These figures indicate that regardless of class, dataset-averaged performance metrics are more similar to each other (and thus more stable) when using an SVM. However, the average of these performance metrics do not vary significantly between the two classifiers or between classes.

Comparing datasets, we see that the variability in performance metrics is somewhat reflective of the number of training images. In Fig. 8 we show the overall and class-averaged metric scores on the test datasets (as detailed in Table 6) and the corresponding number of training images. Note that the size of training sets in Fig. 8 have been square root transformed for easier visual interpretation. Across datasets there is typically a general trend of decreasing performance with shrinking training set size, with Dataset 1 (the largest) scoring best followed by smaller datasets, 2 & 3. However, when focusing on individual classes we see that the number of training images is not always sufficient alone to explain variation in performance. Training sets were typically dominated by Soft Sub. followed by Hard Sub. & Reef, however for each of these, performance often declined significantly between the smaller datasets in which class representation was roughly equivalent. This was particularly pronounced for Soft Sub. and Reef and common to both classifiers.

To uncover any patterns that may explain imperfect model accuracy and identify classifications that were perhaps more challenging, we visually inspected model decisions across the datasets. Interrogation of the misclassified images found that Hard Sub. was most prevalent followed by Soft Sub. and then Reef. Soft and Hard Sub. were largely mistaken for each other, whilst Reef was typically classified as Hard Sub. Analysis of these images first highlighted that the classifiers had corrected erroneous annotations. In the remaining images, unfamiliar objects and characteristics, as well as features associated with another class, could be confusing the classifiers. In Fig. 9(A-C) for example, we show Soft Sub. images that have been mislabelled as Hard Sub. In (A) we see anthropogenic debris, rare and unique in appearance and in (B) and (C) sponges that occur in Hard Sub. communities such as *Phakellia stet.* and Geodiidae, which appear vaguely boulder-like. Overlap of habitat features may also be causing some difficulty in the remaining habitats, for example in (D) we see a Hard Sub. image with some gravel, but largely sediment with evidence of bioturbation, features typical of Soft Sub. images. Additionally, in (H) a Reef image is colonised by Hard Sub. sponges. Obscured views and augmented feature appearance as shown in (E,F,G & I) could also be another cause of the misclassified images. This is caused by sediment disturbance and blurring, as well as inconsistent altitude and angle of the imaging platform, which subsequently affects illumination patterns and the size and perspective of habitat features.

3.3. Time considerations

Aside from variation with respect to classification performance, the two modelling approaches yielded notable differences in training time. In Fig. 10 we demonstrate final training, and testing time, for each approach. For visual clarity, time (minutes) has been natural log transformed. However, to facilitate better comprehension we discuss time in minutes throughout this section. Training was consistently faster across

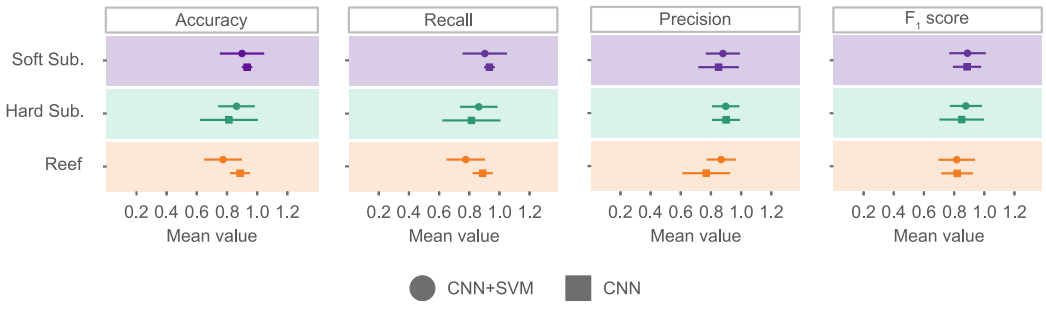


Fig. 7. Mean performance across test datasets for each habitat. Error bars demonstrate 95% confidence intervals.

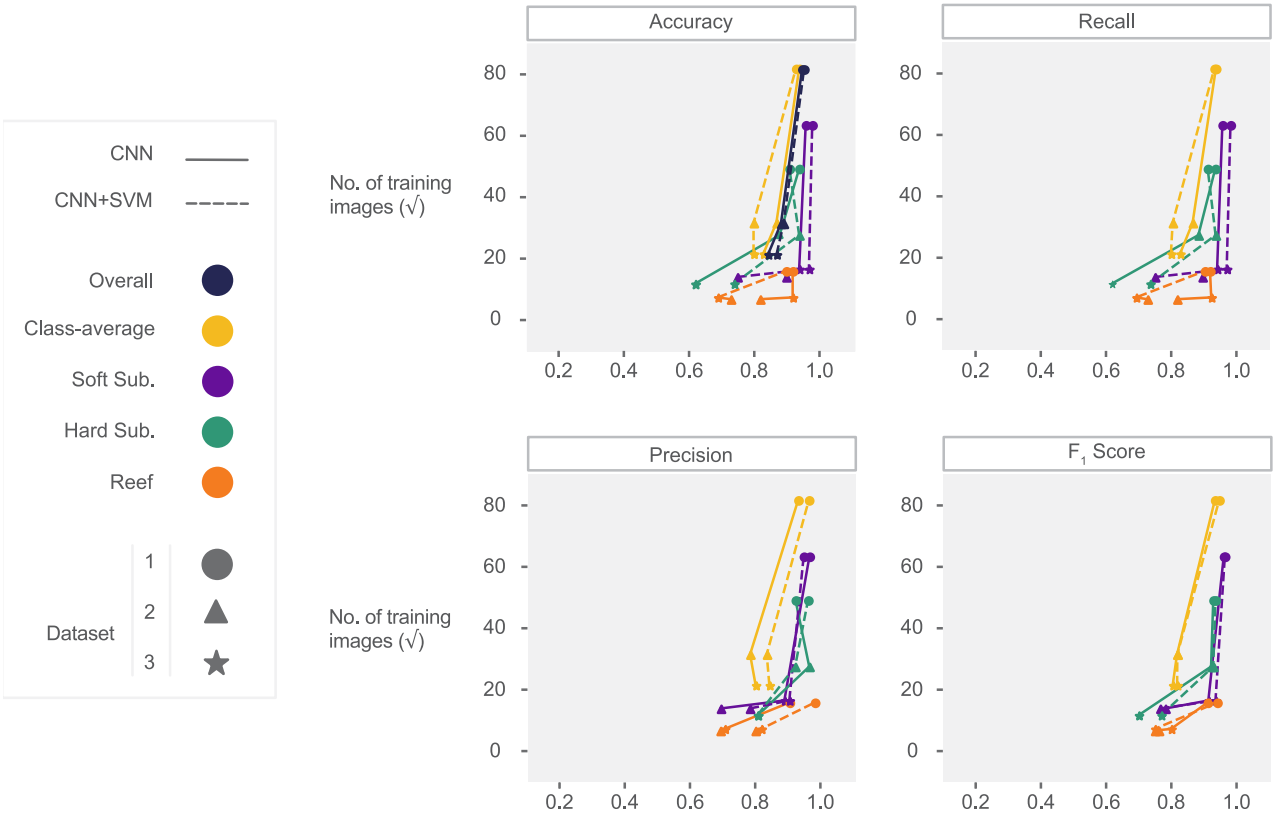


Fig. 8. Overall, class-averaged and individual class performance across test datasets in relation to training set size.

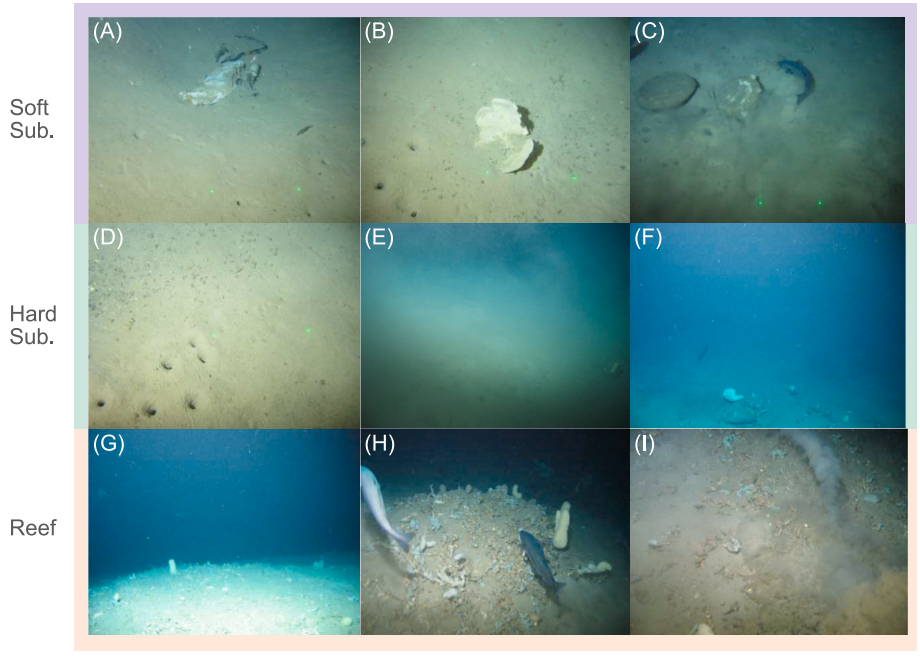


Fig. 9. Examples of misclassified images: (A-C) depict Soft Sub. mislabelled as Hard Sub., (D—F) show Hard Sub. classified as Soft Sub. and lastly Reef in (G-I) predicted as Hard Sub.

datasets when using an SVM. With a GPU, SVM training took between 0.58 and 16.21 ($\mu = 6 \pm 8.17$) minutes, 2.4–5× faster than training a CNN, which took 2.89–38.78 ($\mu = 15.37 \pm 18.74$) minutes. In Fig. 10, we also demonstrate the increased training speed when relying solely on a CPU. On average, training time took 3.6 (± 3.92) minutes longer for the **CNN + SVM** when utilising a CPU, ranging between 1.35 and 24.69 ($\mu = 9.6 \pm 12.09$) minutes. However, CNN training required, on average, 387.09 (± 472.32) minutes more, ranging between 67.78 and 1015.3 ($\mu = 402.46 \pm 491.07$) minutes; a significant deterioration in training time. In addition, this was approximately 40–50× slower than the **CNN + SVM**. Notably, training each **CNN + SVM** with only a CPU was still faster than training each **CNN** with a GPU.

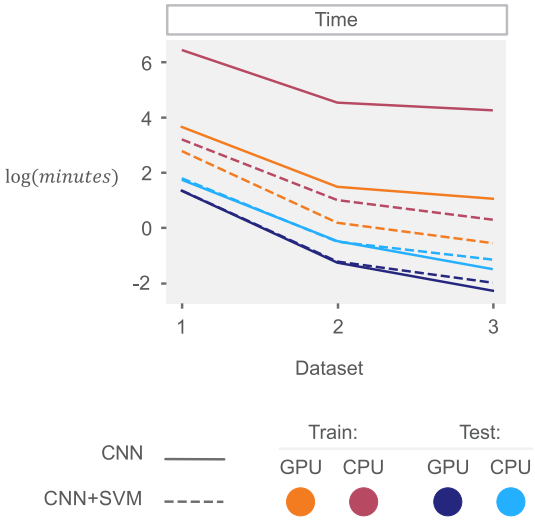


Fig. 10. Time required for training and testing phase, in order of decreasing dataset size.

Testing time was found to be equivalent between the two classification approaches, but vary with hardware. Using a GPU completed the testing phase, on average, 0.87 (± 0.7) minutes faster across both methods. GPU-enabled testing time ranged between 0.1 and 3.88 $(\mu = 1.43 \pm 1.38)$ minutes compared to 0.28–5.87 $(\mu = 2.26 \pm 2.9)$ minutes using just the CPU.

4. Discussion

In this work we have demonstrated that automatic classification of habitats from benthic imagery is possible with high accuracy. We use multiple datasets (of varying sizes) to confirm this, captured by a range of imaging platforms in varying geographic regions.

In line with other research (Mahmood et al., 2016; Mahmood et al., 2019; Mahmood et al., 2020a; Mahmood et al., 2020b; Mohamed et al., 2020), we find that general features extracted from imagery by 'off-the-shelf' convolutional neural networks are suitable for classifying benthic imagery. They also exhibit relatively clear associations with the broad habitat classes. Most notably, the general feature spaces enabled the distinction of soft and hard substrate habitats, aligning with the distinct visual contrast of these habitat classes. The overlap of Hard Sub. and Reef feature spaces following dimension reduction, particularly within Datasets 2 & 3, is unsurprising given the ecological association of Reef and hard substrates (Howell et al., 2011; Roberts et al., 2009). Larvae of coral species such as *Desmophyllum pertusum*, responsible for creating the reef framework in the featured datasets, require hard substrate to settle and build their communities (Roberts et al., 2009; Wilson, 1979). Reef is also more commonly associated with topographic highs (created by boulders and bedrock for example) that enable elevation into currents for enhanced filter feeding (Mienis et al., 2007; Roberts et al., 2009; Thiem et al., 2006). Imagery classified as Reef will thus likely contain evidence of hard substrates. Some images would also have featured the sub-class coral rubble, as described in Table 2. In a technical sense, the texture and edge distributions of coral rubble in imagery, may exhibit more visual similarity to seabed areas dominated by gravel.

Although useful as a broad survey tool, the discrete portrayal of habitats in this work thus strays somewhat from a realistic representation. In actuality, there is a transition between these benthic habitats and they can co-occur. It is in these 'grey' areas, and subsequent overlap of class feature spaces, that the classifiers may be more subject to error, as seen in our visual analysis and other benthic habitat applications (Gómez-Ríos et al., 2019a; Jackett et al., 2023). The means by which these images are annotated only exacerbates this challenge. For practical and environmentally protective purposes, the habitats in these datasets are annotated hierarchically rather than by dominance i.e. presence of any Hard Sub. or Reef characteristics categorizes it as such, even if small. Using a pre-trained feature extractor, that is not optimized to the training data, already risks the importance of these characteristics not being recognized. However, this may be impeded further if these non-dominant characteristics are obscured by natural factors (sediment, fauna) and poor image quality or missed due to placement i.e. if they occur at the edge of images and are reduced or excluded due to the reshaping of images during feature extraction. For example if a small cobble and associated sponges are not captured in the feature space, then the image is likely to be classified as Soft Sub. A finer-scale classification with image patches would better account for spatial heterogeneity in an image, however the classification errors associated with mixed habitat characteristics do still occur (Jackett et al., 2023).

Working with image patches has the benefit of generating large training datasets, with potentially clearer class distributions. Their use in training can also be used to classify a full-sized image (Vega et al., 2024). Despite the comparatively small size of some of our datasets and our broad structure-based classification, we find that our performances compare and compete well with those that classify benthic imagery from a textural perspective, featuring highly-zoomed image patches (Beijbom et al., 2012; Gómez-Ríos et al., 2019a; Gómez-Ríos et al., 2019b; Jackett et al., 2023; Mahmood et al., 2020a; Mahmood et al., 2020b; Yamada et al., 2021; Yamada et al., 2022; Yamada et al., 2023). We also find this extends to full-image classification of benthic habitats, by various automated methods (Mohamed et al., 2020; Rao et al., 2017; Seiler et al., 2012; Vega et al., 2024). Across these benthic classification studies, overall performances average around 90% accuracy in the best-case scenarios, though this can significantly fluctuate within classes and according to the dataset used. These methodologies are comparatively complex and although optimized for more classes, the baseline approach in this study achieves performances within the same magnitudes, offering a viable solution for simpler classification tasks.

Inconsistent appearance of classes is widely understood to affect performance in machine learning and this work is no exception. Whether variation is naturally occurring (morphology) or a result of the imaging process, the classifiers must make a decision and will thus make errors. There is no unknown category when the uncertainty level reaches a specific threshold. In fact, classifying images as *unknown* is a ML problem in itself, as you must train a model to recognize what it doesn't know (Blair et al., 2022; Eerola et al., 2024; Gawlikowski et al., 2023). Regardless of the cause, it is difficult to put an exact value on what an acceptable level of error is in classification tasks. Ideally, ML should, at a minimum, match the current performance baselines set by manual analysis. However, manual accuracies will vary with respect to the annotation task and individual (both intra- & interpersonally) (Beijbom et al., 2015; Culverhouse et al., 2003; Culverhouse et al., 2014; Durden et al., 2016; Schoening et al., 2012) and do not necessarily conform to a 'gold standard' (Schoening et al., 2016). We note that performance of our habitat recognition pipelines was found comparable or better than accuracies following manual annotation of marine imagery (43–95%) (Culverhouse et al., 2003; Durden et al., 2016; Schoening et al., 2012). These studies focus on tasks dissimilar to our own, such as phytoplankton or benthic megafauna classification and they possess a higher number of classes. However they highlight that whilst there is room for improvement, our results meets reported standards in marine image analysis pipelines. Additionally, the fact that the trained models in this work identified some manual annotation errors demonstrates that even an imperfect model, with <100% accuracy, could provide a useful screening tool to improve quality of the ground-truth. Here although the task is a simple one, its time-consuming nature means that for many large datasets it would be impossible to complete this task fully due to manual analysis constraints. Thus there is a suitable trade-off with the error-rate and efficiency of the approaches used. Regardless, the accuracy and reliability of annotations should always be interrogated,

whether manual or automatically generated and judged against the annotation purpose.

4.1. Simplicity vs. complexity

In this application, we note that SVM classifiers offer a good advantage over the more complex CNN classifier, for the following reasons: 1) performance, 2) time constraints and 3) ease of implementation.

SVM classifiers performed well, competing with CNN classifiers across datasets. They were also more conservative in their predictions, and thus more suited to this application in which prediction accuracy is more important than capturing all instances of a habitat. Performance was also not *particularly* sensitive to alterations in training set size, handling poorly represented classes such as Reef reasonably well. SVMs always converge to an optimum, thus irrespective of training set size, if class distinction is possible in the underlying data they are likely to perform well. Although the sensitivity of the CNN classifier to training set size appeared similar to the SVM, typically they require larger quantities of data to achieve higher performance. This is due to their iterative training approach in which they gradually improving their ability to fit complex relationships to the high-dimensional feature data to enable class prediction.

We find that SVM classifiers train much faster than CNN classifiers and demand fewer computational resources. Considering that much of the heavy computation in the **CNN + SVM** approach is within the CNN feature extraction phase, this accounts for the less pronounced increase in training time across datasets when using a CPU. Likely training time could be further improved if the GPU was employed for SVM training. Classifying images in the testing phase is computationally inexpensive and thus no advantage was found between the two models. Although these CPU/GPU comparisons are an extreme example and time demands will vary with hardware specifications, they emphasize well the importance of hardware in selecting appropriate models in image classification. For example, GPUs are an expected requirement for CNN training given the success of CNNs in image classification is partially linked to the development and availability of GPUs (LeCun et al., 2015).

From a coding perspective, training an SVM is extremely straightforward, with training, optimisation and testing executed within only a few simple lines of code. The relative complexity with the **CNN + SVM** approach, and by extension the **CNN** approach, rather lies in data management and feature extraction. The complexity of these steps is not so much related to the actions required, but navigating the extensive literature and knowing "where to start". As one of the contributions of this paper we hope to better guide the user with clear and detailed descriptions of these steps, that are applicable across ML frameworks (Table 3). Aside from these preparations, training the **CNN** classifier is further complicated since an optimal is not automatically found. Instead performance metrics must be monitored across epochs and decisions made on when to stop training - typically the point at which training and test accuracy are near equivalent, to minimize over- and under- fitting. In addition, with each application, time-consuming tuning and exploration of hyperparameters is essential to maximise their performance.

4.2. Improving performance

The approaches presented in this study were designed to minimize complexity and reflect a realistic performance baseline that the reader could expect when attempting a similar classification task on their own data. That being said, enhancing the suitability of the underlying dataset and features could improve model performance.

Classifiers highlighted both inaccuracies in the ground-truth and the difficulty of overlapping habitats and generalizing to unseen characteristics in the training data. Verifying the accuracy of manual annotations and removing images that are not clear examples of habitat could thus improve performance. Dataset cleaning such as this has been shown to increase model accuracy by ~12% in benthic habitat classification, but risks impeding performance on novel uncleaned data (Jackett et al., 2023). Increasing the amount of training data, as echoed in our own work, could also increase performance metrics, both traditional (Durden et al., 2021; Piechaud et al., 2019; Yamada et al., 2021) and ecological (Durden et al., 2021). In the absence of additional suitable data, data augmentation to re-sample and transform existing training data, such that it generalizes to datasets more widely, is also commonly undertaken (Jackett et al., 2023; Vega et al., 2024). However, care must be taken since many common augmentations may not in fact be applicable to marine datasets (Tan et al., 2022). Data augmentation can also be used to solve class imbalance (Langenkämper et al., 2019b), which encourages models, particularly CNNs, to develop biases such as over-predicting common classes. Class imbalance can alternatively be handled by down-weighting common classes or up-sampling rare classes (Durden et al., 2021; Langenkämper et al., 2019b). Increasing availability, and quality, of the ground-truth would also allow for further tuning or scratch-based learning of the CNN perhaps creating feature representations even more suitable for automatic classification of benthic habitats. Whereas reducing the size of feature spaces could aid classification by lessening noise within the feature space and reducing the sparsity of data points, which makes finding groups with similar properties challenging (curse of dimensionality). This can be achieved through feature importance selection, dimensionality reduction or employing a network which outputs less features, such as GoogleNet (Szegedy et al., 2015) and Inception V3 (Szegedy et al., 2015).

4.3. Future work & concluding remarks

Although these techniques have been around a long time, and there seems to be a willingness to conduct more AI among non-specialists, its usage within marine science is lagging. Studies such as this, which focus on methods conceptually and practically targeted at non-specialists, will encourage further uptake and development. In addition, research needs to move past 'proof of concept' studies on singular datasets alone (where possible), excluding those comparing benchmark datasets.

It is important to have realistic expectations when designing and employing machine learning pipelines. Not all tasks will be suitable for automation and human interaction will always be required at some stage. Equally, the task at hand will govern the complexity of the machine learning approach required. In this study we focused on a relatively simple but time-consuming manual task that would benefit marine scientists, promoting efficient broad-scale monitoring. We also prioritised simplicity of the modelling approach over any potential gain in habitat resolution. We have presented one of a few cases of automatic habitat classification from imagery and demonstrated that SVM (tuned RBF) classifiers paired with a VGG16 feature extractor offer a simple, fast and consistent framework. There may be desire to implement state of the art Deep Learning approaches in the ecological community, such as CNNs, or even Vision Transformers (Dosovitskiy et al., 2020; Vaswani et al., 2017) where feasible. However, given our results, we believe it is useful to first employ the **CNN + SVM** approach, for tasks of a similar complexity or application, as it may prove sufficient. For example, an exploratory application of the **CNN + SVM** approach to compiled image datasets from the North Sea yielded 90% accuracy. Although this was a large dataset (~19k training images), it was highly imbalanced and uncleaned, with annotations often made per station rather then per image. It also contained several EUNIS habitat classes (MC12, MC2211, MD32, MC/D42, MC/D52, MC/D62) (European Environment Agency, 2022) and a class for poor quality imagery i.e. where the seabed was obscured due to sediment disturbance. Though it should be noted that more sophisticated methods exist to handle unknown classes (Blair et al., 2022; Eerola et al., 2024; Gawlikowski et al., 2023).

Despite the resolution of the broad habitats used in this study, the **CNN + SVM** approach can be seen as a good option for ecologists seeking classification methods that yield a high return on minimal

investment, with classes that have clear characteristics. It forms an important component in developing a hierarchical, or ensemble, analysis tool for benthic imagery. A 'first-pass' method which can identify the next suitable analysis steps and automatic tools to use, such as classifiers for the next node in an hierarchy (Gómez-Ríos et al., 2019a; Mahmood et al., 2020a; Vega et al., 2024) or object detectors for specific taxa found within each habitat. This could then support higher resolution habitat classifications through recognizing sea pens, boulders or reef status for example.

The approach presented is suited to offshore use; offering near real-time decision making in the field and the development of sampling protocols. Data collection can be triaged and quick, albeit crude, insights into habitat presence provided. It can be used to screen old-datasets and support manual annotation by grouping similar imagery. The reduced manual constraints may also support monitoring at a higher resolution, by allowing image analysts to instead focus manual efforts on quality control and more challenging annotation tasks.

Future work could look at model transferability (domain adaptation) between datasets and testing the **CNN + SVM** approach as a fine-scale monitoring tool, classifying habitat patches within an image. Another interesting progression would be to look at better integration of acoustic, optical and environmental data to support contextual-based automation (Ellen et al., 2019; Rao et al., 2017; Shields et al., 2020; Yamada et al., 2021).

Funding

This work was supported by NERC and EPSRC grants: NE/N012070/1 and EP/S028730/1, respectively.

CRediT authorship contribution statement

Chloe A. Game: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Michael B. Thompson:** Writing – review & editing, Supervision, Resources, Conceptualization. **Graham D. Finlayson:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization.

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors thank Gardline Ltd. and their environmental reporting team for providing annotated imagery used within this project.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ecoinf.2024.102619.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283. URL. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi?ref=https://githubhelp.com.

Abad-Uribarren, A., Prado, E., Sierra, S., Cobo, A., Rodríguez-Basalo, A., Gómez-Ballesteros, M., Sánchez, F., 2022. Deep learning-assisted high resolution mapping of vulnerable habitats within the Capbreton canyon system, Bay of Biscay, Estuarine. Coast. Shelf Sci. 275, 107957 https://doi.org/10.1016/j.ecss.2022.107957. URL. https://www.sciencedirect.com/science/article/pii/S0272771422002153.

Abosaq, H.A., Ramzan, M., Althobiani, F., Abid, A., Aamir, K.M., Abdushkour, H., Irfan, M., Gommosani, M.E., Ghonaim, S.M., Shamji, V.R., Rahman, S., 2023. Unusual driver behavior detection in videos using deep learning models. Sensors 23 (1), 311. https://doi.org/10.3390/s23010311. URL. https://www.mdpi.com/1424-8220/23/1/311.

Althubiti, S.A., Alenezi, F., Shitharth, S., S. K, Reddy, C.V.S., 2022. Circuit manufacturing defect detection using VGG16 convolutional neural networks. Wirel. Commun. Mob. Comput. 2022, e1070405 https://doi.org/10.1155/2022/1070405. RL. https://www.hindawi.com/journals/wcmc/2022/1070405/.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al- Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8 (1), 53. https://doi.org/10.1186/s40537-021-00444-8.

Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S., 2015. From generic to specific deep representations for visual recognition. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 36–45. https://doi.org/10.1109/CVPRW.2015.7301270.

Baker, E.K., Harris, P.T., 2020. Chapter 2 - habitat mapping and marine management. In: Harris, P.T., Baker, E. (Eds.), Seafloor Geomorphology as Benthic Habitat, Second edition. Elsevier, pp. 17–33. https://www.sciencedirect.com/science/article/pii/B9780128149607000026.

Beijbom, O., Edmunds, P.J., Kline, D.I., Mitchell, B.G., Kriegman, D., 2012. Automated annotation of coral reef survey images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1170–1177. https://doi.org/10.1109/CVPR.2012.6247798.

Beijbom, O., Edmunds, P.J., Roelfsema, C., Smith, J., Kline, D.I., Neal, B.P., Dunlap, M.J., Moriarty, V., Fan, T.-Y., Tan, C.-J., 2015. Towards automated annotation of benthic survey images: variability of human experts and operational modes of automation. PLoS One 10 (7), e0130312. https://doi.org/10.1371/journal.pone.0130312.

Beijbom, O., Treibitz, T., Kline, D.I., Eyal, G., Khen, A., Neal, B., Loya, Y., Mitchell, B.G., Kriegman, D., 2016. Improving automated annotation of benthic survey images using wide-band fluorescence. Sci. Rep. 6 (1), 23166 https://doi.org/10.1038/srep23166.

Bewley, M., Friedman, A., Ferrari, R., Hill, N., Hovey, R., Barrett, N., Marzinelli, E.M., Pizarro, O., Figueira, W., Meyer, L., Babcock, R., Bellchambers, L., Byrne, M., Williams, S.B., 2015. Australian sea-floor survey data, with images and expert annotations. Scient. Data 2 (1), 150057. https://doi.org/10.1038/sdata.2015.57. URL. https://www.nature.com/articles/sdata201557.

Blair, J., Weiser, M.D., de Beurs, K., Kaspari, M., Siler, C., Marshall, K.E., 2022. Embracing imperfection: machine-assisted invertebrate classification in real-world datasets. Eco. Inform. 72, 101896 https://doi.org/10.1016/j.ecoinf.2022.101896. URL. https://www.sciencedirect.com/science/article/pii/S1574954122003466.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. https://doi.org/10.1023/A:1010933404324.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3) https://doi.org/10.1145/1961189.1961199. Article 27.

Chen, Q., Beijbom, O., Chan, S., Bouwmeester, J., Kriegman, D., 2021. A New Deep Learning Engine for CoralNet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3693–3702. https://doi.org/10.1109/ICCVW54120.2021.00412.

Christin, S., Hervet, N., 2019. Lecomte, Applications for deep learning in ecology. Methods Ecol. Evol. 10 (10), 1632–1644. https://doi.org/10.1111/2041-210X.13256.

Cogan, C.B., Todd, B.J., Lawton, P., Noji, T.T., 2009. The role of marine habitat mapping in ecosystem-based management. ICES J. Mar. Sci. 66 (9), 2033–2042. https://doi.org/10.1093/icesjms/fsp214. https://doi.org/10.1093/icesjms/fsp214.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297. https://doi.org/10.1007/BF00994018.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.

Crosby, A., Orenstein, E.C., Poulton, S.E., Bell, K.L., Woodward, B., Ruhl, H., Katija, K., Forbes, A.G., 2023. Designing Ocean Vision AI: An investigation of community needs for imaging-based ocean conservation. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23. Association for Computing Machinery, New York, NY, USA, pp. 1–16. https://doi.org/10.1145/3544548.3580886.

Culverhouse, P.F., Williams, R., Reguera, B., Herry, V., González- Gil, S., 2003. Do experts make mistakes? A comparison of human and machine indentification of dinoflagellates. Mar. Ecol. Prog. Ser. 247, 17–25. https://doi.org/10.3354/meps247017. URL. https://www.int-res.com/abstracts/meps/v247/p17-25/.

Culverhouse, P.F., Macleod, N., Williams, R., Benfield, M.C., Lopes, R.M., Picheral, M., 2014. An empirical assessment of the consistency of taxonomic identifications. Mar. Biol. Res. 10 (1), 73–84. https://doi.org/10.1080/17451000.2013.810762. https://doi.org/10.1080/17451000.2013.810762.

Davies, C.E., Moss, D., Hill, M.O., 2004. EUNIS Habitat Classification Revised 2004. Report to: European Environment Agency European Topic Centre on Nature Protection and Biodiversity. URL. https://www.eea.europa.eu/data-and-maps/data/eunis-habitat-classification-1/documentation/eunis-2004-report.pdf.

Dawkins, M., Sherrill, L., Fieldhouse, K., Hoogs, A., Richards, B., Zhang, D., Prasad, L., Williams, K., Lauffenburger, N., Wang, G., 2017. An open-source platform for underwater image and video analytics. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 898–906. https://doi.org/10.1109/WACV.2017.105.

Deng, J., Dong, W., Socher, R., Li, L.J., Kai, L., Li, F.-F., 2009. ImageNet: a large-scale hierarchical image database. IEEE Confer. Comp. Vision Pattern Recogn. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

Diaz, R.J., Solan, M., Valente, R.M., 2004. A review of approaches for classifying benthic habitats and evaluating habitat quality. J. Environ. Manag. 73 (3), 165–181. https://doi.org/10.1016/j.jenvman.2004.06.004.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations (ICLR) 2021. URL. https://openreview.net/forum?id=YicbFdNTTy, 21 pages.

Downie, A.-L., Noble-James, T., Sperry, J., White, S., Mar. 2022. Automated Detection of Sea Pens in Video Footage: Applications for Time-Series Monitoring of Vulnerable Marine Ecosystems (VME), Tech. Rep..

J. M. Durden, B. J. Bett, T. Schoening, K. J. Morris, T. W. Nattkemper, H. A. Ruhl, Comparison of image annotation data generated by multiple investigators for benthic ecology, Mar. Ecol. Prog. Ser. 552 (2016) 61–70. doi:https://doi.org/10.3354/meps11775. URL https://www.int-res.com/abstracts/meps/v552/p61-70/.

Durden, J.M., Hosking, B., Bett, B.J., Cline, D., Ruhl, H.A., 2021. Automated classification of fauna in seabed photographs: the impact of training and validation dataset size, with considerations for the class imbalance. Prog. Oceanogr. 196, 102612 https://doi.org/10.1016/j.pocean.2021.102612.

Eerola, T., Batrakhanov, D., Barazandeh, N.V., Kraft, K., Haraguchi, L., Lensu, L., Suikkanen, S., Seppälä, J., Tamminen, T., Kälviäinen, H., 2024. Survey of automatic plankton image recognition: challenges, existing solutions and future perspectives. Artif. Intell. Rev. 57 (5), 114. https://doi.org/10.1007/s10462-024-10745-y.

Ellen, J.S., Graff, C.A., Ohman, M.D., 2019. Improving plankton image classification using context metadata. Limnol. Oceanogr. Methods 17 (8), 439–461. https://doi.org/10.1002/lom3.10324. https://onlinelibrary.wiley.com/doi/abs/10.1002/lom3.10324.

European Environment Agency, 2022. EUNIS habitat classification. Available at: https://www.eea.europa.eu/data-and-maps/data/eunis-habitatclassification-1 [Accessed 19 Dec. 2022]. URL. https://www.eea.europa.eu/data-and-maps/data/eunis-habitat-classification-1.

European Parliament, 2008. Directive 2008/56/EC of the European Parliament and of the Council of 17 June 2008 Establishing a Framework for Community Action in the Field of Marine Environmental Policy (Marine Strategy Framework Directive).

Evans, D., Aish, A., Boon, A., Condé, S., Connor, D., Gelabert, E.M.N., Parry, M., Richard, D., Salvati, E., Tunesi, L., 2016. Revising the Marine Section of the EUNIS Habitat Classification-Report of a Workshop Held at the European Topic Centre on Biological Diversity, 12 & 13 May 2016. ETC/BD report to the EEA., Tech. Rep..

Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., Zhu, X.X., 2023. A survey of uncertainty in deep neural networks. Artif. Intell. Rev. 56 (1), 1513–1589. https://doi.org/10.1007/s10462-023-10562-9.

Gómez-Ríos, A., Tabik, S., Luengo, J., Shihavuddin, A.S.M., Herrera, F., 2019a. Coral species identification with texture or structure images using a two-level classifier based on convolutional neural networks. Knowl.-Based Syst. 184, 104891 https://doi.org/10.1016/j.knosys.2019.104891.

Gómez-Ríos, A., Tabik, S., Luengo, J., Shihavuddin, A.S.M., Krawczyk, B., Herrera, F., 2019b. Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. Expert Syst. Appl. 118, 315–328. https://doi.org/10.1016/j.eswa.2018.10.010.

González-Rivero, M., Beijbom, O., Rodriguez-Ramirez, A., Bryant, D.E.P., Ganase, A., Gonzalez-Marrero, Y., Herrera-Reveles, A., Kennedy, E.V., Kim, C.J.S., Lopez-Marcano, S., Markey, K., Neal, B.P., Osborne, K., Reyes-Nivia, C., Sampayo, E.M., Stolberg, K., Taylor, A., Vercelloni, J., Wyatt, M., Hoegh-Guldberg, O., 2020. Monitoring of coral reefs using artificial intelligence: a feasible and cost-effective approach. Remote Sens. 12 (3), 489. https://doi.org/10.3390/rs12030489. URL. https://www.mdpi.com/2072-4292/12/3/489.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. URL. http://www.deeplearningbook.org.

Harris, P.T., Baker, E.K., 2020. Chapter 1 - why map benthic habitats? In: Harris, P.T., Baker, E. (Eds.), Seafloor Geomorphology as Benthic Habitat, Second edition. Elsevier, pp. 3–15. RL. https://www.sciencedirect.com/science/article/pii/B9780128149607000014.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W.Wang, T.Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (Apr. 2017). Doi:10.48550/arXiv.1704.04861. URL http://arxiv.org/abs/1704.04861.

Howell, K.L., Holt, R., Endrino, I.P., Stewart, H., 2011. When the species is also a habitat: comparing the predictively modelled distributions of Lophelia pertusa and the reef habitat it forms. Biol. Conserv. 144 (11), 2656–2665. https://doi.org/10.1016/j.biocon.2011.07.025.

Howell, K.L., Hilário, A., Allcock, A.L., Bailey, D.M., Baker, M., Clark, M.R., Colaço, A., Copley, J., Cordes, E.E., Danovaro, R., 2020. A blueprint for an inclusive, global deep-sea ocean decade field program. Front. Mar. Sci. 999. https://doi.org/10.3389/fmars.2020.584861.

Hsu, C.W., Chang, C.C., Lin, C.J., 2016. A Practical Guide to Support Vector Classication. Available at: https://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf [Accessed 22 Dec. 2022]. URL. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Jackett, C., Althaus, F., Maguire, K., Farazi, M., Scoulding, B., Untiedt, C., Ryan, T., Shanks, P., Brodie, P., Williams, A., 2023. A benthic substrate classification method for seabed images using deep learning: application to management of deep-sea coral reefs. J. Appl. Ecol. https://doi.org/10.1111/1365-2664.14408.

Kandimalla, V., Richard, M., Smith, F., Quirion, J., Torgo, L., Whidden, C., 2022. Automated detection, classification and counting of fish in fish passages with deep learning. Front. Mar. Sci. 8 https://doi.org/10.3389/fmars.2021.823173.

Katija, K., Orenstein, E., Schlining, B., Lundsten, L., Barnard, K., Sainz, G., Boulais, O., Cromwell, M., Butler, E., Woodward, B., 2022. FathomNet: a global image database for enabling artificial intelligence in the ocean. Sci. Rep. 12 (1), 15914.

Kaur, T., Gandhi, T.K., 2019. Automated Brain Image Classification Based on VGG-16 and Transfer Learning. In: 2019 International Conference on Information Technology (ICIT), pp. 94–98. https://doi.org/10.1109/ICIT48102.2019.00023.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, San Diego, USA. https://doi.org/10.48550/arXiv.1412.6980. URL. http://arxiv.org/abs/1412.6980.

Kloster, M., Langenkämper, D., Zurowietz, M., Beszteri, B., Nattkemper, T.W., 2020. Deep learning-based diatom taxonomy on virtual slides. Sci. Rep. 10 (1), 14416 https://doi.org/10.1038/s41598-020-71165-w.

Krishnaswamy Rangarajan, A., Purushothaman, R., 2020. Disease classification in eggplant using pre-trained VGG16 and MSVM. Sci. Rep. 10 (1), 2322 https://doi.org/10.1038/s41598-020-59108-x.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc. URL. https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Langenkämper, D., Zurowietz, M., Schoening, T., Nattkemper, T.W., 2017. BIIGLE 2.0 - browsing and annotating large marine image collections, Frontiers in marine. Science 4, 83. https://doi.org/10.3389/fmars.2017.00083.

Langenkämper, D., Simon-Lledó, E., Hosking, B., Jones, D.O.B., Nattkemper, T.W., 2019a. On the impact of citizen science-derived data quality on deep learning based classification in marine images. PLoS One 14 (6), e0218086. https://doi.org/10.1371/journal.pone.0218086.

Langenkämper, D., van Kevelaer, R., Nattkemper, T.W., 2019b. Strategies for tackling the class imbalance problem in marine image classification. In: Zhang, Z., Suter, D., Tian, Y., Branzan Albu, A., Sidère, N., Jair Escalante, H. (Eds.), Pattern Recognition and Information Forensics, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 26–36. https://doi.org/10.1007/978-3-030-05792-3_3.

Langenkämper, D., van Kevelaer, R., Purser, A., Nattkemper, T.W., 2020. Gear-induced concept drift in marine images and its effect on deep learning classification. Front. Mar. Sci. 7 (506) https://doi.org/10.3389/fmars.2020.00506.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324. https://doi.org/10.1109/5.726791.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.

Liu, J., Zhang, L., Li, Y., Liu, H., 2023. Deep residual convolutional neural network based on hybrid attention mechanism for ecological monitoring of marine fishery. Eco. Inform. 77, 102204 https://doi.org/10.1016/j.ecoinf.2023.102204. URL. https://www.sciencedirect.com/science/article/pii/S1574954123002339.

Liu, Y., Wang, S., 2021. A quantitative detection algorithm based on improved faster R-CNN for marine benthos. Eco. Inform. 61, 101228 https://doi.org/10.1016/j.ecoinf.2021.101228. URL. https://www.sciencedirect.com/science/article/pii/S1574954121000194.

A. Lumini, L. Nanni, Deep learning and transfer learning features for plankton classification, Eco. Inform. 51 (2019) 33–43. doi:https://doi.org/10.1016/j.ecoinf.2019.02.007. URL https://www.sciencedirect.com/science/article/pii/S1574954118303054.

Mahmood, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Kendrick, G., Fisher, R.B., 2016. Coral classification with hybrid feature representations. In: IEEE International Conference on Image Processing (ICIP), pp. 519–523. https://doi.org/10.1109/ICIP.2016.7532411.

Mahmood, A., Bennamoun, M., An, S., Sohel, F.A., Boussaid, F., Hovey, R., Kendrick, G. A., Fisher, R.B., 2019. Deep image representations for coral image classification. IEEE J. Ocean. Eng. 44 (1), 121–131. https://doi.org/10.1109/JOE.2017.2786878.

Mahmood, A., Ospina, A.G., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Fisher, R.B., Kendrick, G.A., 2020a. Automatic hierarchical classification of kelps using deep residual features. Sensors 20 (2), 447. https://doi.org/10.3390/s20020447. URL. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7013955/.

Mahmood, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., 2020b. ResFeats: residual network based features for underwater image classification. Image Vis. Comput. 93, 103811 https://doi.org/10.1016/j.imavis.2019.09.002. URL. https://www.sciencedirect.com/science/article/pii/S0262885619301362.

Marburg, A., Bigham, K., 2016. Deep learning for benthic fauna identification. In: Oceans 2016 MTS/IEEE Monterey, pp. 1–5. https://doi.org/10.1109/OCEANS.2016.7761146.

Mienis, F., de Stigter, H.C., White, M., Duineveld, G., de Haas, H., van Weering, T.C.E., 2007. Hydrodynamic controls on cold water coral growth and carbonate mound development at the SW and SE Rockall trough margin, NE Atlantic Ocean. Deep-Sea Res. Part 1 Oceanogr. Res. Papers 54, 1655–1674 doi:urn:nbn:nl:ui:31-1871/30349.

Mohamed, H., Nadaoka, K., Nakamura, T., 2020. Semiautomated mapping of benthic habitats and seagrass species using a convolutional neural network framework in shallow water environments. Remote Sens. 12 (23), 4002 https://doi.org/10.3390/rs12234002. URL. https://www.mdpi.com/2072-4292/12/23/4002.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: an imperative style, highperformance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc p. Article 721.

Piechaud, N., Howell, K.L., 2022. Fast and accurate mapping of fine scale abundance of a VME in the deep sea with computer vision. Eco. Inform., 101786 https://doi.org/10.1016/j.ecoinf.2022.101786. URL. https://www.sciencedirect.com/science/article/pii/S1574954122002369.

Piechaud, N., Hunt, C., Culverhouse, P.F., Foster, N.L., Howell, K.L., 2019. Automated identification of benthic epifauna with computer vision. Mar. Ecol. Prog. Ser. 615, 15–30. https://doi.org/10.3354/meps12925. URL. https://www.int-res.com/abstracts/meps/v615/p15-30/.

PyTorch, 2023. PyTorch. Available at: https://pytorch.org/ [Accessed 02 Mar. 2023]. URL. https://www.pytorch.org.

Rao, D., De Deuge, M., Nourani-Vatani, N., Williams, S.B., Pizarro, O., 2017. Multimodal learning and inference from visual and remotely sensed data. Intern. J. Robot. Res. 36 (1), 24–43. https://doi.org/10.1177/0278364916679892.

Razavian, S., Ali, H. Azizpour, Sullivan, J., Carlsson, S., 2014. CNN Features off-the-shelf: An astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813.

Roberts, J.M., Wheeler, A., Freiwald, A., Cairns, S., 2009. Cold-Water Corals: The Biology and Geology of Deep-Sea Coral Habitats. Cambridge University Press, Cambridge.

Rubbens, P., Brodie, S., Cordier, T., Destro Barcellos, D., Devos, P., Fernandes-Salvador, J.A., Fincham, J.I., Gomes, A., Handegard, N.O., Howell, K., Jamet, C., Kartveit, K.H., Moustahfid, H., Parcerisas, C., Politikos, D., Sauzède, R., Sokolova, M., Uusitalo, L., Van den Bulcke, L., van Helmond, A.T.M., Watson, J.T., Welch, H., Beltran-Perez, O., Chaffron, S., Greenberg, D.S., Kühn, B., Kiko, R., Lo, M., Lopes, R.M., Möller, K.O., Michaels, W., Pala, A., Romagnan, J.-B., Schuchert, P., Seydi, V., Villasante, S., Malde, K., J.-O., 2023. Irisson, machine learning in marine ecology: an overview of techniques and applications. ICES J. Mar. Sci. 80 (7), 1829–1853. https://doi.org/10.1093/icesjms/fsad100.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 115 (3), 211–252. https://doi.org/10.1007/s11263-015-0816-y.

Sala, E., Lubchenco, J., Grorud-Colvert, K., Novelli, C., Roberts, C., Sumaila, U.R., 2018. Assessing real progress towards effective ocean protection. Mar. Policy 91, 11–13. https://doi.org/10.1016/j.marpol.2018.02.004. URL. https://www.sciencedirect.com/science/article/pii/S0308597X17307686.

Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., Harvey, E., 2016. Fish species classification in unconstrained underwater environments based on deep learning. Limnol. Oceanogr. Methods 14 (9), 570–585. https://doi.org/10.1002/lom3.10113.

Schoening, T., Bergmann, M., Ontrup, J., Taylor, J., Dannheim, J., Gutt, J., Purser, A., Nattkemper, T.W., 2012. Semi-automated image analysis for the assessment of Megafaunal densities at the Arctic Deep- Sea observatory HAUSGARTEN. PLoS One 7 (6), e38179. https://doi.org/10.1371/journal.pone.0038179.

Schoening, T., Osterloff, J., Nattkemper, T.W., 2016. Reco-MIA—recommendations for marine image annotation: lessons learned and future directions. Front. Mar. Sci. 3 https://doi.org/10.3389/fmars.2016.00059.

scikit-learn, 2023. scikit-learn: machine learning in Python. Available at: https://scikit-learn.org/stable/index.html [Accessed 2 Mar. 2023]. URL. https://scikit-learn.org/stable/index.html.

Seiler, J., Friedman, A., Steinberg, D., Barrett, N., Williams, A., Holbrook, N.J., 2012. Image-based continental shelf habitat mapping using novel automated data extraction techniques. Cont. Shelf Res. 45, 87–97. https://doi.org/10.1016/j.csr.2012.06.003. URL. https://www.sciencedirect.com/science/article/pii/S0278434312001513.

Shields, J., Pizarro, O., Williams, S.B., 2020. Towards Adaptive Benthic Habitat Mapping, pp. 9263–9270. https://doi.org/10.1109/ICRA40945.2020.9196811.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for largescale image recognition. In: 3rd International Conference on Learning Representations, San Diego, USA. https://doi.org/10.48550/arXiv.1409.1556. URL. http://arxiv.org/abs/1409.1556.

skorch, 2022. skorch 0.12.1 documentation. Available at: https://skorch.readthedocs.io/en/stable/ [Accessed 22 Dec. 2022]. URL. https://skorch.readthedocs.io/en/stable/.

Song, W., Zheng, N., Liu, X., Qiu, L., Zheng, R., 2019. An improved U-net convolutional networks for seabed mineral image segmentation. IEEE Access 7, 82744–82752. https://doi.org/10.1109/ACCESS.2019.2923753.

Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

Tan, M., Langenkämper, D., Nattkemper, T.W., 2022. The impact of data augmentations on deep learning-based marine object classification in benthic image transects. Sensors 22 (14), 5383. https://doi.org/10.3390/s22145383. URL. https://www.mdpi.com/1424-8220/22/14/5383.

Thiem, E., Ravagnan, J.H., Foss, A., Berntsen, J., 2006. Food supply mechanisms for cold-water corals along a continental shelf edge. J. Mar. Syst. 60 (3), 207–219. https://doi.org/10.1016/j.jmarsys.2005.12.004. URL. https://www.sciencedirect.com/science/article/pii/S0924796305002071.

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B.R., Zuffi, S., Risse, B., Mathis, A., Mathis, M.W., van Langevelde, F., Burghardt, T., Kays, R., Klinck, H., Wikelski, M., Couzin, I.D., van Horn, G., Crofoot, M.C., Stewart, C.V., Berger-Wolf, T., 2022. Perspectives in machine learning for wildlife conservation. Nat. Commun. 13 (1), 792. https://doi.org/10.1038/s41467-022-27980-y. URL. https://www.nature.com/articles/s41467-022-27980-y.

United Nations, 2018. Revised Roadmap for the UN Decade of Ocean Science for Sustainable Development. Available at: https://unesdoc.unesco.org/ark:/48223/pf0000265141 [Accessed 23 Dec. 2022]. URL. https://unesdoc.unesco.org/ark:/48223/pf0000265141.

United Nations General Assembly, 2015. Transforming Our World: the 2030 Agenda for Sustainable Development. Available at: https://sdgs.un.org/2030agenda [Accessed 23 Dec. 2022]. RL. https://sdgs.un.org/2030agenda.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, I. Polosukhin, 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc.. URL. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Vega, P.J.S., Papadakis, P., Matabos, M., Van Audenhaege, L., Ramiere, A., Sarrazin, J., da Costa, G.A.O.P., 2024. Convolutional neural networks for hydrothermal vents substratum classification: An introspective study. Eco. Inform. 80, 102535 https://doi.org/10.1016/j.ecoinf.2024.102535. URL. https://www.sciencedirect.com/science/article/pii/S1574954124000773.

Wang, Y., Huang, H., Rudin, C., Shaposhnik, Y., 2021. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. J. Mach. Learn. Res. 22 (201), 1–73. URL. http://jmlr.org/papers/v22/20-1061.html.

Weinstein, B.G., 2018. A computer vision for animal ecology. J. Anim. Ecol. 87 (3), 533–545. https://doi.org/10.1111/1365-2656.12780.

Williams, S.B., Pizarro, O.R., Jakuba, M.V., Johnson, C.R., Barrett, N.S., Babcock, R.C., Kendrick, G.A., Steinberg, P.D., Heyward, A.J., Doherty, P.J., Mahon, I., Johnson-Roberson, M., Steinberg, D., Friedman, A., 2012. Monitoring of benthic reference sites: using an autonomous underwater vehicle. IEEE Robot. Automat. Magaz. 19 (1), 73–84. https://doi.org/10.1109/MRA.2011.2181772.

Wilson, J.B., 1979. 'Patch' development of the deep-water coral *Lophelia Pertusa* (L.) on Rockall Bank. J. Mar. Biol. Assoc. U. K. 59 (1), 165–177. https://doi.org/10.1017/S0025315400046257. URL. https://www.cambridge.org/core/journals/journal-of-the-marine-biological-association-of-the-united-kingdom/article/abs/patch-development-of-the-deepwater-coral-lophelia-pertusa-l-on-rockall-F1EB7A6071BDE772C6E98FD27CACC5BD.

Wynn, R.B., Huvenne, V.A.I., Le Bas, T.P., Murton, B.J., Connelly, D.P., Bett, B.J., Ruhl, H.A., Morris, K.J., Peakall, J., Parsons, D.R., Sumner, E.J., Darby, S.E., Dorrell, R.M., Hunt, J.E., 2014. Autonomous underwater vehicles (AUVs): their past, present and future contributions to the advancement of marine geoscience. Mar. Geol. 352, 451–468. https://doi.org/10.1016/j.margeo.2014.03.012.

Yamada, T., Prügel-Bennett, A., Thornton, B., 2021. Learning features from georeferenced seafloor imagery with location guided autoencoders. J. Field Robot. 38 (1), 52–67. https://doi.org/10.1002/rob.21961.

Yamada, T., Prügel-Bennett, A., Williams, S.B., Pizarro, O., Thornton, B., 2022a. GeoCLR: Georeference contrastive learning for efficient seafloor image interpretation. ArXiv. https://doi.org/10.55417/fr.2022037 abs/2108.06421. URL. http://eprints.soton.ac.uk/id/eprint/456914.

Yamada, T., Massot-Campos, M., Prugel-Bennett, A., Pizarro, O., Williams, S.B., Thornton, B., 2023 Jan. Guiding Labelling Effort for Efficient Learning With Georeferenced Images. IEEE Trans Pattern Anal Mach Intell. 45 (1), 593–607. https://doi.org/10.1109/TPAMI.2021.3140060.

Yang, D., Martinez, C., Visuña, L., Khandhar, H., Bhatt, C., Carretero, J., 2021b. Detection and analysis of COVID-19 in medical images using deep learning techniques. Sci. Rep. 11 (1), 19638 https://doi.org/10.1038/s41598-021-99015-3.

Yang, H., Ni, J., Gao, J., Han, Z., Luan, T., 2021a. A novel method for peanut variety identification and classification by improved VGG16. Sci. Rep. 11 (1), 15756 https://doi.org/10.1038/s41598-021-95240-y.

Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks?. In: Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc.. URL. https://papers.nips.cc/paper/2014/hash/375c71349b295fbe2dcdca9206f20a06-Abstract.html

Zhang, B., Xie, F., Han, F., 2019. Fish population status detection based on deep learning system. In: 2019 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 81–85. https://doi.org/10.1109/ICMA.2019.8816263.

Zhang, J., Yongpan, W., Xianchong, X., Yong, L., Lyu, L., Wu, Q., 2022. YoloXT: a object detection algorithm for marine benthos. Eco. Inform. 72, 101923 https://doi.org/10.1016/j.ecoinf.2022.101923. URL. https://www.sciencedirect.com/science/article/pii/S1574954122003739.