A New Deep Learning Engine for CoralNet

Qimin Chen[1] Oscar Beijbom[2] Stephen Chan[1] Jessica Bouwmeester[3,4] David Kriegman[1]
qic003@ucsd.edu {oscar.beijbom, stephenjchan, jess.bouwmeester}@gmail.com kriegman@cs.ucsd.edu
[1]UC San Diego [2]Motional [3]Smithsonian Conservation Biology Institute
[4]Hawai'i Institute of Marine Biology

Abstract

CoralNet is a cloud-based website and platform for manual, semi-automatic and automatic analysis of coral reef images. Users access CoralNet through optimized web-based workflows for common tasks, and other systems can interface through API's. Today, marine scientists are widely using CoralNet, and nearly 3,000 registered users have uploaded 1,741,855 images from 2,040 distinct sources with over 65 million annotations. CoralNet is hosted on AWS, is free for users, and the code is open source [1]. *In January 2021, we released CoralNet 1.0 which has a new machine learning engine. This paper provides an overview of that engine, and the process of choosing the particular architecture, its training, and a comparison to some of the most promising architectures. In a nutshell, CoralNet 1.0 uses transfer learning with an EfficientNet-B0 backbone that is trained on 16M labelled patches from benthic images and a hierarchical Multi-layer Perceptron classifier that is trained on source-specific labelled data. When evaluated on a hold-out test set of 26 sources, the error rate of CoralNet 1.0 was 18.4% (relative) lower than CoralNet Beta.*

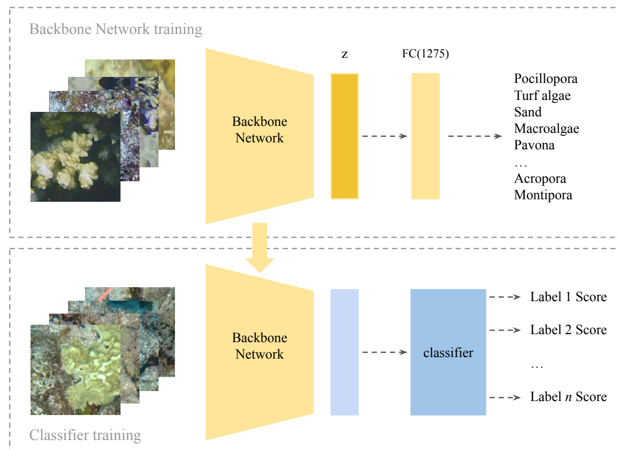

Figure 1. **CoralNet 1.0 machine learning system**. The first stage is to train a single deep backbone network pooling a large collection of data. After training, the backbone network is then used as a feature extractor. In the second stage, a per-source classifier is trained on feature vectors z extracted by the well-trained backbone network.

1. Introduction

It is commonly understood that coral reefs are in rapid decline globally due to confounding factors including rising temperatures, ocean acidification, pollution, over-fishing, disease, predators like crown of thorns, etc. [9, 13, 28, 20] Consequences include losses of up to 80% of coral cover in the Caribbean and 50% in the Indo-Pacific and Great Barrier Reef over the past 30 years [10, 19]. Quantifying the state of the reefs, determining the impact of the various causative factors, and measuring the benefit of remediation efforts requires carefully designed surveys, the means for acquiring image data, and methods for analysis, each of which must operate at large scale.

The goal of a typical survey is to measure the cover of study-specific taxa. If the goal is simply to estimate coral coverage, it may be sufficient to classify coral vs. non-coral. If the health of reefs is degrading, it may be relevant to also classify the state of the corals – healthy, bleached, diseased, dead. It may be desirable to classify the major functional groups on a reef – coral, algae, sand, rock. Perhaps finer granularity is needed to classify the corals to the genus or species level or to classify algae as crustose coralline algae, turf algae, or macro algae, as well as to categorize sponges and other common invertebrates on the reef. The set of labels varies with each study, and typically ranges from a few classes to a hundred.

Due to the vast scales (*e.g.*, reefs can be hundreds of square kilometers), it is impractical to survey an entire reef, and so studies rely on sampling. Typically, surveys [12] are done with downward pointing cameras, often photographing the reef through a frame or quadrat of a known size (*e.g.* 0.5m×0.5m) that is just above the bottom, so that the

[1]https://github.com/beijbom/coralnet

absolute scale and coverage in an area can be determined up to some experimental accuracy. The coverage area of each class within the image is traditionally done through labelling of a sample of point locations in the images. The point locations may be deterministic (*e.g.,* a 10×10 grid) or randomly placed. An analyst will then classify each point location using the specific set of labels for that study. Commonly used tools for manual annotation include Coral Point Count with Excel Extensions (CPCe) [23], PhotoQuad [41], PointCount99 [32], BIIGLE [24]. CoralNet was conceived as a tool for semi-automatic and automatic analysis of benthic images using this same methodology.

The astute computer vision researcher will recognize that even greater coverage and potentially accuracy could be achieved using semantic segmentation, where every pixel in the image is labeled using the study-specific label set rather than up to a few hundred pixel. In practice, the vast majority of marine scientists have opted for manual labelling of points rather than densely segmenting images because dense labelling of objects with complex, irregular boundaries is incredibly time consuming. Consider that a typical survey taken by a pair of divers on scuba may yield about a thousand images per week. Manual point annotation of that data may take one or two months. When scientists have produced manual semantic segmentation of images, it has usually been for a small number of classes, often for the purpose of measuring the growth of individual coral heads [31] or determining spatial distributions [11]. Given the widespread use and availability of labelled point data, CoralNet presently only supports point labelling although semantic segmentation is being developed.

CoralNet has had three major releases, and this paper documents the deep learning classification engine for CoralNet 1.0, released in January 2021. CoralNet Alpha was released in Fall 2011 and used a method reported in [3] based on texture and color features, textons, a Bag of Words model and an SVM classifier running on a deskside server. CoralNet Beta was released in December 2016 and used a convolutional neural network (VGG16) with transfer learning [37] that was trained on $63,000$ labeled images that had been uploaded and labelled on CoralNet Alpha. CoralNet Beta, like the current version, ran on Amazon Web Services (AWS). The contributions of this paper are to provide an overview of the CoralNet 1.0 deep learning engine, and the process of choosing particular architecture, its training and comparisons to some of the most promising architectures.

On CoralNet, a study is referred to as a "source", and it contains a collection of images and set of user selected labels drawn from a globally defined label set. CoralNet's machine learning engine performs automatic point annotation - for a given pixel location in an image, the location is classified from a set of labels in a source's label set. The input to the classifier is an image patch cropped from the full size image about that point, and the output is score ranging from 0.0 to 1.0 over each of the n mutually exclusive labels in the label set. For a typical CoralNet source, there are between 5 and 80 labels with most sources having fewer than 20 labels. The label with the highest score is taken to be the annotation for the point. The n scores sum to 1.0 akin to a probability and provide a measure of the confidence of the decision. This score is used in the alleviation ability of CoralNet, that will be described below.

It should be noted that labelling of benthic coral reef images, even by experts, tends to be much more ambiguous than object category labelling typically used in object recognition. This holds when comparing labels by experts of previously unseen data or relabeling their own image data, and because of this inconsistency one expects a ceiling on accuracy by automatic systems such as CoralNet [2].

The CoralNet classification engine is built on a transfer learning approach with two stages to classification. The first stage is a feature extractor (convolutional neural network) that takes an image patch (usually 224×224 pixels) centered on the given point and outputs a feature vector with length that ranges from $1,024$ to $4,096$ depending upon the particular network. The classifier takes the feature vector as input and outputs the n scores.

The CoralNet feature extractor is a convolutional neural network that has been trained on large quantities of benthic images, feature vectors are computed when images are uploaded, and they are stored in a database. The classifier is trained repeatedly on a per source basis, using manually labelled points from that source. For a fresh source the labelset is created by the user. In a typical workflow, the user starts manually labelling points in images through a web-based graphical user interface. Once 20 images have been manually labelled, a classifier is trained using the precomputed features as input with the output compared to these manual labels. With a trained classifier for a source in hand, CoralNet can now use inference to suggest potential labels which the user can verify or manually correct. Every time a user labels 10% additional images, another classifier is trained. Using a validation set comprised of $1/8$ of the already labelled images, CoralNet will replace a previous classifier if the new classifier is at least 1% more accurate than the previous classifier.

Generally speaking, supervised networks require 1000's of examples of each class (label) to be effectively trained, which is typically not available for sources in the CoralNet scenario. The alternative that we developed on CoralNet is to train a single feature extractor pooling a large collection of data and then to train individual classifiers for each source. During the life cycle of a source, classifiers are repeatedly trained for a number of reasons:

- Users manually annotate additional points.
- Users confirm automatic annotations.

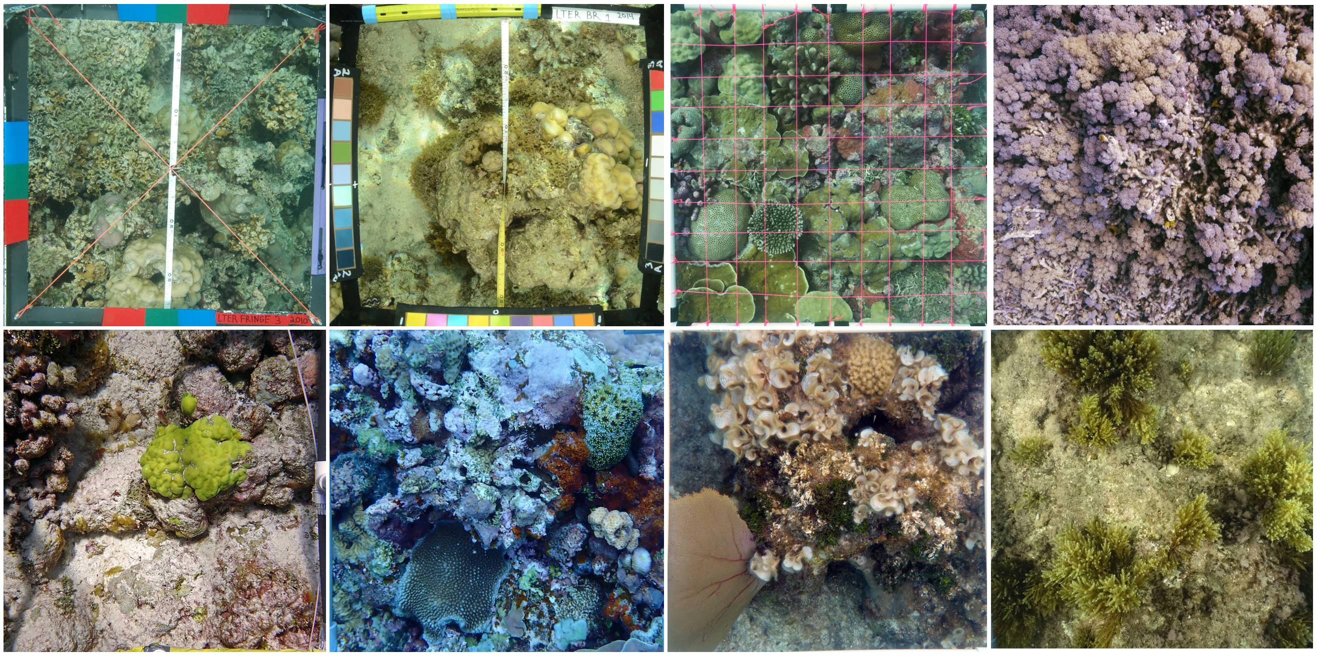

Figure 2. **Examples from different coral sources**. Original high-resolution (varies from 1948×1980 pixels to 3556×3264 pixels) coral images from 8 different sources. First row from left to right: *Moorea Labeled Corals*, *LTER Back Reef*, *KI Benthic Analysis 2014*, *OV1_362*. Second row from left to right: *GU CESU AAFB SPP*, *Taiping Islands*, *Barbuda Reef Cover*, *Canopy-forming Species Levantine Basin*. For each coral image, the number of annotated points might vary from 20 to 200 depending upon the source. Figure 4 shows an example from *Moorea Labeled Corals* and patches centered at each annotated point with a window size of 224 × 224. We point out that images from this figure are hand picked to clearly demonstrate the diversity of each coral source.

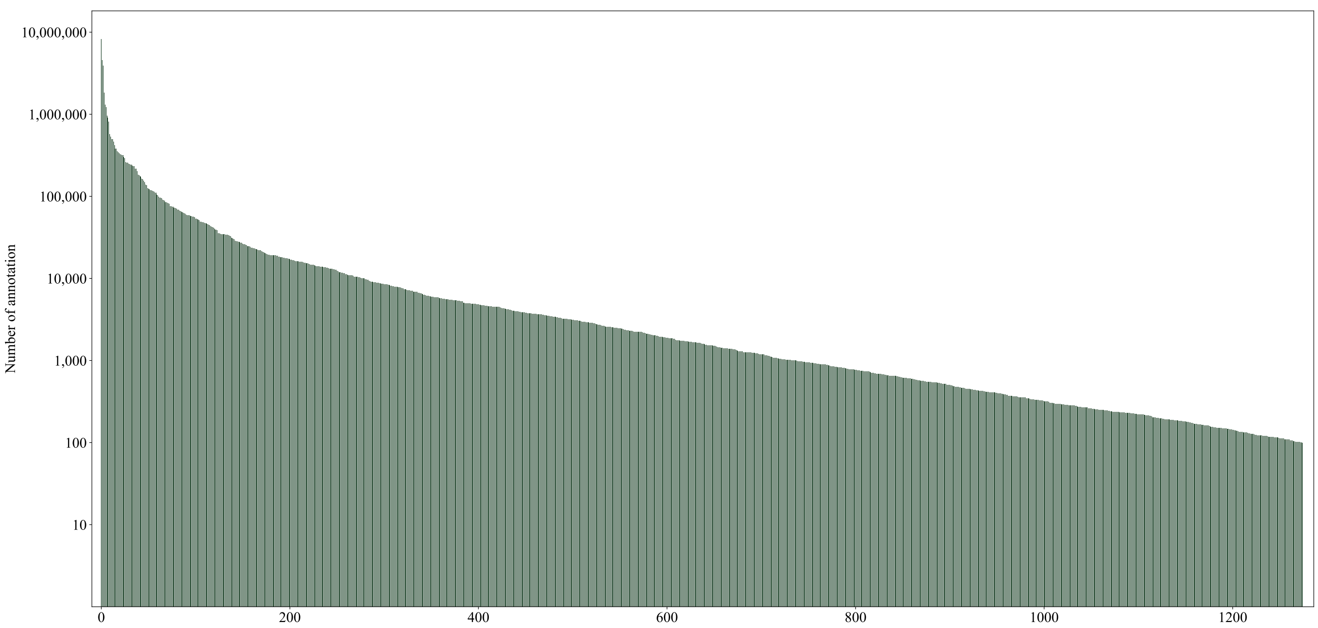

Figure 3. **Distribution of Labels**. The number of time a label is used to annotate a point varies significantly, and the most common label (turf) occurs almost 10M times whereas other labels are far rarer. A log plot of the number of annotations (sorted in descending order) for each of the 1,275 labels.

- Users may decide to add or remove labels from the source's label set.

Once a classifier is deemed to be sufficiently well-trained, the CoralNet user may choose to fully rely on all suggested labels or those where the confidence is greater than a user-specified threshold; the rest are then verified or labelled manually. This significantly alleviates the manual burden with little impact on accuracy. CoralNet also offers a web API that lets users programmatically classify images using a previously trained classifier on CoralNet by simply passing a URL to an image along with a set of row, column locations to be classified. In one study, the API was used to classify over five million points in more than 175,000 images [29].

2. Related Work

There are many potential choices of deep network feature extractors in the computer vision literature and many potential classifiers from the machine learning literature. CoralNet Beta's feature extractor was based on the VGG [37] classifier and was trained in 2017 on 2.4M annotations from $62,000$ images. CNN's usage for image classification are well known and publicized, and full literature review is beyond the scope of this paper. In the course of selecting a next generation feature extractor for CoralNet, we examined the ResNet [17] and EfficientNet [39] families.

2.1. Machine learning for coral analysis

An early research method proposed by Beijbom *et al.* [3] used a 'bag of visual words' approach together with a Support Vector Machine and evaluated it across 9 classes on images from the island of Moorea in French Polynesia. Other earlier works including [26, 36] also relied on combinations of hand engineered feature extractors and machine learning. Treibitz *et al.* used fluorescence signals to utilize the chlorophyll signature present in living organisms and further improve classification performance [40].

More recent works demonstrate significant performance improvement using deep convolutional neural networks [30, 18, 15, 27]. These studies fine-tune deep networks for each study site which requires an abundance of annotations. In contrast, this work performs a shared "pre-tuning" of a large network on the joint CoralNet data corpus and then uses that network in turn as a feature extractor. This allows a more feasible cloud deploy and provides good operating performance even on surveys with very few samples available.

Deep learning has also been extended to semantic segmentation of both 2D images [1, 21, 33] and 3D reconstructions [18].

2.2. Impact on large scale surveys

CoralNet is now being widely used by marine scientists in the course of their research and conservation efforts. An independent 2018 NOAA study, using images from about 1400 coral reef survey sites, demonstrated that CoralNet Beta was capable of estimating coral cover data with accuracy comparable to human analysts, although estimates of some coral genera and algal groups were less accurate [42]. Another large scale study compiled and examined images from five different global regions in the period from 2012 to 2016: Central Pacific Ocean, Western Atlantic Ocean, Central Indian Ocean, Southeast Asia and Eastern Australia. They used the same deep learning based classifier as is used in CoralNet and found unbiased and high agreement between the ecological compositions calculated from expert and automated observations [16, 35].

2.3. Software services

Other efforts related to automatic analysis of coral include Squidle (`https://squidle.org/`) and Reefcloud (`https://reefcloud.ai/`). Both provide similar capabilities as CoralNet for importing, annotating and analyzing benthic surveys. MAIA [43], an extension of BIIGLE [24], also provides an annotation platform with automatic annotation.

Squidle [4] is developed by Greybits Engineering [2] and the public data view suggests wide site adoption across the world in particular by larger programs. One such example is the Australian Center of Field Robotics which rely on Autonomous Underwater Vehicles to collect large amounts of survey data.

Reefcloud is an open access platform developed by the Australian Institute of Marine Science and is supported by the Australian government.

3. Dataset

Marine scientists have uploaded 1.7 million images from over $2,040$ ecological surveys ("sources") from around the world since the release of CoralNet Alpha. As the name implies, CoralNet was originally created for the annotating of coral reefs, but scientists have found value in a broader range of habitats and classes from sea grasses and cold water rocky habitats to oil rigs, pier pilings and autonomous reef monitoring structures (ARMS). The vast majority of sources are from the tropics, but uploaded images range from as far south as Antarctica to as far north as Scotland.

Since there is no universally agreed upon set of labels or taxonomy, and since most sources are created by different groups of marine scientists, users have defined a total of $4,489$ labels. This includes duplicates - different labels and names for the same taxa. With the help of coral biologists,

[2] `https://greybits.com.au/`

we identified 315 duplicate labels covering $5,436,343$ annotations, and merged corresponding duplicates into a common label. We selected 280 representative sources for training the deep learning engine, and these contain $432,489$ images with $15,137,977$ annotated points. These sources are randomly divided into 254 sources for training and testing the backbone networks and 26 sources for training and testing the classifiers. We selected $1,279$ labels that 1) are used in at least 3 sources, 2) are used to annotate at least 100 points, and 3) which do not designate "unsure", "dark", or similar catch-all categories. We designate this first set **V1**. In a later version of the dataset, designated as **V2**, we exported 50 more sources for training and further removed 4 more catch-all type labels. In summary:

- **V1**: 280 sources, $432,489$ images, $15,137,977$ annotated points; 254 sources for training the backbone network with $1,279$ classes in common; 26 sources for training the classifiers, each randomly split into 80/20 for training and testing.
- **V2**: 330 sources, $591,604$ images, $16,533,651$ annotated points; 304 sources for training the backbone network with $1,275$ classes in common; the same 26 sources and splits as in **V1** for training and testing the classifiers.

Figure 2 shows examples from different coral sources, and Figure 4 shows an example of a coral image and patches cropped from the original image centered on the given annotated points with a window size of 224×224 pixels. Figure 3 shows a log histogram of the distribution of the number images for $1,275$ labels. Note that while some labels occur more than a million times, almost half of the labels are only used hundreds of times.

4. Training Setup

In this section, we describe two components of the CoralNet 1.0 machine learning system – the Backbone Network and the Classifier – and how these are trained. Figure 1 shows an overview of CoralNet's transfer learning approach with two stages.

4.1. Feature extraction

In the first stage we train a convolutional neural network (backbone) to function as a feature encoder. We follow a standard transfer learning procedure [14] and replace the final backbone layer with a new linear layer which maps to the number of classes used. The backbone network is trained end to end using cross entropy loss and the final layer represents the probability of this patch being classified as each of the $1,275$ classes. After training, we remove the final layer and use the activation of the second-to-last layer

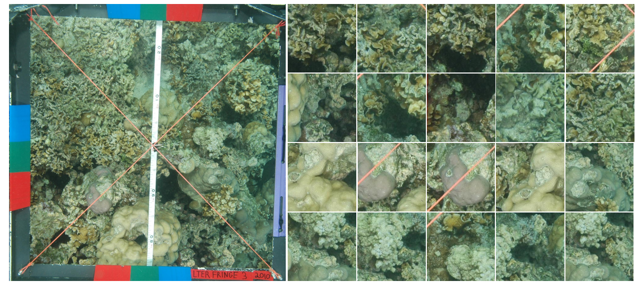

Figure 4. **Moorea Labeled Corals example**. An original 1948 × 1980 pixel image from *Moorea Labeled Corals* on the left and patches centered at annotated points with a window size of 224 × 224 on the right. Each class is represented by a row with five example patches. First row: *Pavona*. Second row: *CCA*. Third row: *Porites*. Forth row: *Montipora*.

as a feature vector of $1,024$ to $4,096$ dimensions depending upon the backbone. We conducted experiments using networks from the VGG [37], ResNet [17] and EfficientNet [39] families.

4.2. Classification

For each source, a classifier is trained that takes as input a feature vector extracted by the backbone network and outputs a classification score for each of the n labels in that source. We evaluated different classifiers including Logistic Regression Classifier (LR), Multi-layer Perceptron (MLP), Support Vector Machine (SVM) [8], Random Forest [5] and Gaussian Naive Bayes [6].

Hyper-parameters tuning is generally required for training Multi-layer Perceptron (MLP) in order to avoid overfitting and achieve better performance. Therefore we first review the definition of MLP and define the hierarchical MLP. The multi-layer perceptron network is composed of K hidden layers with hidden units $h_1, ..., h_K$ followed by an activation function f that maps the input data to output domain (*e.g.* coral labels).

4.2.1 Hierarchical MLP

We explored a number of different full connected networks (Multi-layer Perceptron) with a softmax layer as a classifier, varying the number of layers, the number of hidden units per layer, and the activation function. We also explored different learning rates for training these networks. In the process of hyper-parameter tuning, we found that a 3-layer network outperformed a 2-layer network when the amount of training data was large (e.g., more than $50,000$ annotations), and we felt that it was desirable to provide users with higher accuracy when they had gone to the trouble of manually annotating or verifying so many points. Since we are not restricted to use the same MLP architecture for all sources, we decided on a hierarchical structure. For large sources with more annotations than a threshold $(50,000)$, we would use one network; for sources with fewer annotations, we would use a smaller network.

More specifically, we performed a hyper-parameter search of single network and hierarchical networks, by considering the parameters of the number of layers (2 or 3), the number of hidden units per layer h_i, the learning rate n, the activation function $f \in \{ReLU, tanh\}$, and whether it is a hierarchical MLP denoted by the presence of a $threshold$, itself a hyper-parameter. A sample network's hyper-parameters are:

$$mlp(h_1; n = 10^{-3}; f = ReLU; s \leq threshold)$$
$$mlp(h_2, h_3; n = 10^{-4}; f = ReLU; s > threshold)$$

4.3. Implementation details

For training the backbone networks, we used a batch size of 128, initial learning rate of 10^{-3} and Adam [22] optimizer with one cycle learning policy [38] to adjust the learning rate throughout the experiments.

For training the classifiers, we used the default hyper-parameters setup provided by [34] for Logistic Regression Classifier, SVM [8] and Gaussian Naive Bayes [6]. We set the number of trees in the forest as 30 when training Random Forest [5]. We provide a detailed hyper-parameter tuning for Multi-layer Perceptron in Table 4.

5. Empirical Evaluation of Design Choices

In this section, we provide experimental results that lead to the design decisions for the deployed deep learning engine for CoralNet 1.0. We first consider the accuracy and speed of different backbone networks and then consider different classifiers for transfer learning. We then explore the effect of floating point precision, and conclude the evaluation with a comparison of CoralNet 1.0 to CoralNet Beta.

5.1. Backbone accuracy

We first need to select a backbone network to use for CoralNet 1.0. To evaluate performance after training the backbone on dataset **V1**, we performed transfer learning using Logistic Regression on the 26 CoralNet sources that were not used for training the backbone. Table 2 shows the mean and standard deviation of accuracy over the 26 sources for VGG16 [37], ResNet50/101 [17], EfficientNet-B0/B1/B4 [39] as backbone feature extractors.

Since the size of feature vector might significantly affect accuracy, we also trained three variations of EfficientNet-B0 [39] with embedding sizes of $1,280$ (default), $2,048$ and $4,096$, and accuracy is also shown in Table 2.

The slight benefit to accuracy for a larger embedding size was outweighed by longer training time and larger storage costs. It is worth noting that the standard deviation in accuracy reflects the difficulty of the classification task for different sources.

5.2. Backbone inference time

A pragmatic decision for deploying a free service such as CoralNet that is expected to operate at large scales is the cost of inference. GPU's are faster than CPU's, but cost more. Backbone networks have different computation cost. For CPU timing, we use an AMD Ryzen 7 3700x 8-core processor × 16, and for GPU we use a GeForce RTX 2080 Ti/PCIe/SSE2. EfficientNet-B0 [39] outperforms other network architectures in both CPU and GPU inference time. Figure 5 shows the trade-off of inference

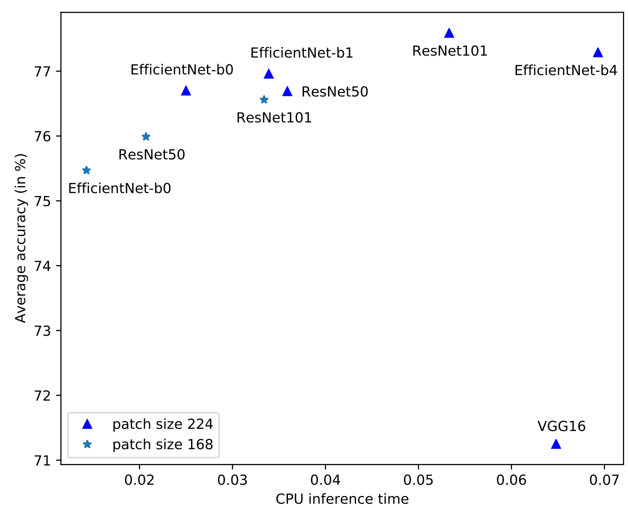

Figure 5. **Inference time vs. Accuracy**. X-axis represents CPU inference time in second (s) and y-axis represents Average accuracy on 26 test sources in percentage (%). Patch size corresponds to the input image size.

time vs. accuracy for each backbone network. From this we conclude that nearly all of the networks are faster than CoralNet Beta (VGG16), potentially twice as fast. The accuracy of CoralNet Beta is 71.36%, and all of the other networks are more accurate by a solid margin. Two networks stand out: ResNet101 [17] is the most accurate network at 77.54%. EfficientNet-B0 [39] pays a small price on accuracy (76.50%), but it is more than twice as fast as ResNet101 [17]. Consequently, we chose EfficientNet-B0 for CoralNet 1.0.

5.3. Classifiers

Having chosen EfficientNet-B0 [39] as the backbone network, we needed to select a classifier architecture for transfer learning. We evaluated Logistic Regression (LR), Multi layer Perceptron (MLP), Support vector machines (SVM)

Source ID	# Total images	# Patches per image	# Total Annotation	# Labels	CoralNet Beta	CoralNet 1.0
s16	2,055	199	408,945	16	78.60	84.15
s338	1,970	200	394,000	43	54.90	64.52
s472	2,336	100	233,600	21	78.50	82.33
s111	1,571	100	157,100	46	71.50	85.37
s1579	15,258	10	152,580	34	71.30	77.83
s586	2,209	50	110,450	37	69.40	76.23
s407	700	100	70,000	11	76.90	81.00
s1070	687	100	68,700	28	68.10	87.13
s1504	303	225	68,175	19	79.10	81.33
s1091	2,270	30	68,100	25	77.50	86.29
s843	2,820	20	56,400	42	72.90	76.99
s1055	1,701	30	51,030	10	86.80	92.16
s699	914	50	45,700	25	80.90	85.74
s1074	211	200	42,200	9	54.80	53.00
s294	201	200	40,200	9	80.00	86.35
s521	350	100	35,000	16	72.80	76.52
s656	1,279	24	30,696	49	59.10	66.31
s614	1,201	25	30,025	32	56.20	59.10
s609	598	50	29,900	23	77.70	88.42
s1559	453	50	22,650	21	68.00	73.02
s1591	2,048	10	20,480	10	69.70	72.50
s601	315	50	15,750	12	70.20	67.41
s367	312	50	15,600	13	64.60	66.63
s672	387	25	9,675	19	59.80	63.18
s1396	600	10	6,000	4	89.80	90.00
s1420	300	20	6,000	16	66.20	73.13
Mean±STD	-	-	-	-	71.36±9.16	76.79±10.14

Table 1. **CoralNet 1.0 vs. CoralNet Beta**. Summary of 26 coral sources including 1) number of total images, 2) number of patches (annotated points) per image, 3) number of total annotations and 4) number of labels (classes) and the accuracy (shown in %) on each source of CoralNet Beta vs CoralNet 1.0. For CoralNet 1.0, we use EfficientNet-B0 [39] as backbone network trained on dataset **V2** and hierarchical MLP setup ② shown in Table 4. The table is sorted by the number of total annotations in descending order.

Source ID	CoralNet Beta	VGG16	ResNet50	ResNet101	EfficientNet (network version - embedding size)				
					b0 - 1280	b0 - 2048	b0 - 4096	b1 - 1280	b4 - 1280
Mean±STD	71.36±9.16	73.72±11.13	76.81±10.29	77.54±10.36	76.50±10.36	76.86±10.09	77.06±9.82	77.02±10.04	77.34±10.45

Table 2. **Performance of different backbone networks**. Logistic Regression (LR) classifier accuracy (shown in %) with backbone networks Beijbom *et al*. [3], VGG16 [37], ResNet50 [17], ResNet101 [17], EfficientNet-B0 [39], EfficientNet-B1 [39] and EfficientNet-B4 [39] on 26 test sources. We further show the results of EfficientNet-B0 with different embedding sizes (default 1280, 2048 and 4096). All backbone networks are trained on dataset **V1**.

[8], Random Forest [5] and Gaussian Naive Bayes [6], and these are shown in Table 3. We found that even a simple 2-layer MLP was more accurate than the other classifiers, and so we explored different MLP configurations in a second set of experiments (Table 4). As shown in Figure 3, there can be severe class imbalance, and we tried a variety of data balancing methods from sub-sampling the dominant class to augmentation of the sparser classes. We also applied SMOTE [7] to address the imbalance (See Table 4), but none of these methods had a significant effect on accuracy. Ultimately, we chose the Hierarchical MLP architecture shown in bold in Table 4 for transfer learning.

5.4. Floating point precision

Floating point precision can affect accuracy and run time, and we show this trade-off in Table 5. Accuracy and F1-score do not change considerably when the data type varies from half precision to double precision. However, half precision and single precision are considerably faster.

5.5. CoralNet 1.0 vs. CoralNet Beta

Having chosen and trained EfficientNet-B0 as a backbone and selected a hierarchical MLP as a classifier for transfer learning for CoralNet 1.0, we now compare CoralNet 1.0 to CoralNet Beta using 26 sources. Table 1 summarizes the 26 sources and the accuracy on each source of CoralNet 1.0 vs. CoralNet Beta and Figure 6 shows a histogram of accuracy improvement of CoralNet 1.0 compared to CoralNet Beta across these 26 sources. The new classifiers outperform those from CoralNet Beta by a solid margin on most sources.

Source ID	CoralNet Beta	EfficientNet-B0						ResNet101	
		LR	MLP	SVM	RF	GNB	MLP-SMOTE	LR	MLP
Mean±STD	71.36±9.16	76.35±10.48	76.79±10.14	76.39±10.49	75.13±10.51	65.91±11.87	76.69±10.23	75.97±10.3	76.2±9.89

Table 3. **Performance of different classifiers**. Accuracy (shown in %) of different classifiers: Logistic Regression (LR), Multi-layer Perceptron (same hierarchical MLP setup as Table 4 ②), Support Vector Machine (SVM) [8], Random Forest (RF) [5], Gaussian Naive Bayes (GNB) [6] with backbone networks EfficientNet-B0 [39] and ResNet101 [17] both trained on dataset **V2**.

Source ID	LR		① $(100; n = 10^{-3}; f = ReLU)$		② $(100; n = 10^{-3}; f = ReLU; s \leq 50,000)$ $(200, 100; n = 10^{-4}; f = ReLU; s > 50,000)$		③ $(100, n = 10^{-3}; f = tanh; s \leq 50,000)$ $(200, 100; n = 10^{-4}; f = tanh; s > 50,000)$		④ $(200, n = 10^{-4}; f = ReLU; s \leq 10,000)$ $(200, 100; n = 10^{-4}; f = ReLU; s > 10,000)$		⑤ $(200, n = 10^{-4}; f = tanh; s \leq 10,000)$ $(200, 100; n = 10^{-4}; f = tanh; s > 10,000)$	
	Acc (%)	F1-score	Acc (%)	F1-score	Acc (%)	F1-score	Acc (%)	F1-score	Acc (%)	F1-score	Acc (%)	F1-score
Mean±STD	76.35±10.48	0.7456±0.1130	76.67±9.84	0.7559±0.1023	**76.79±10.14**	**0.7565±0.1047**	76.77±9.96	0.7562±0.1038	76.64±10.17	0.7553±0.1058	76.62±10.12	0.7534±0.1071

Table 4. **Performance of different MLP hyper-parameters**. Accuracy and F1-Score of different Multi-layer Perceptron hyper-parameters settings. ① Single layer. ②③④⑤ Hierarchical setup depending on the total number of samples s of each source, n denotes the learning rate and f denotes the activation function. Adam optimizer is used in all setups. Take *s16*, *s521* and setup ② for example, as *s16* has $s = 408,945$ samples, $(200, 100; n = 10^{-4}; f = ReLU; s > 50,000)$ is used. Similarly, *s521* has $s = 35,000$ samples, $(100; n = 10^{-3}; f = ReLU; s \leq 50,000)$ is used instead.

	Accuracy (%)	F1-score	Training time (s)
FP16	76.91 ± 10.25	0.7583 ± 0.1049	35.0504
FP32	77.08 ± 10.15	0.7599 ± 0.1059	32.9096
FP64	76.85 ± 10.25	0.7576 ± 0.1072	373.0312

Table 5. **Performance of floating-point data types**. Mean and standard deviation of accuracy and F1-score of MLP with setup ② shown in Table 4 over 26 test sources. Feature vector in different floating-point types (half precision as FP16, single precision as FP32 and double precision as FP64) are extracted by EfficientNet-B0 [39] trained on dataset **V2**.

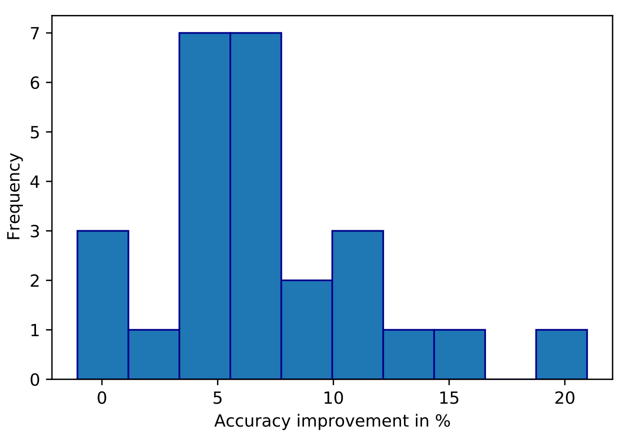

Figure 6. **Accuracy improvement**. Accuracy a_1 of EfficientNet-B0 with MLP ② shown in Table 4 compared to accuracy a_β of CoralNet Beta across 26 test sources. X-axis represents the absolute accuracy improvement $a_1 - a_\beta$, for example, 6 test sources have more than **10**% improvement in accuracy compared to current CoralNet Beta.

6. Discussion and Conclusion

We provided an overview of CoralNet's new deep learning engine named CoralNet 1.0. We show that EfficientNet-B0 [39] as the backbone network with Multi-layer Perceptron (MLP) classifier provides better performance considering both accuracy and speed and therefore serves as the new engine.

As the CoralNet 1.0 engine provides faster and more accurate automatic coral analysis, some future directions would be worth investigating. One would be to use a well-trained backbone network and classifier for direct semantic segmentation without further training. The backbone network can be adapted into fully a convolutional network [25] and transfer the learned representations to the segmentation task. Another one would be to train the backbone with multi-domain classification objective. The label set in CoralNet is flat, and there would be benefits of allowing generic and user defined taxonomies, such as genera-species-subspecies or morphologies. It would also be useful to support multiple labels per point (*e.g.* coral type as one label, and coral health as a second label). Finally, domain transfer and semi-supervised learning are made possible because of the rich feature information and large amount of labelled and unlabelled image data provided by CoralNet.

Acknowledgements We thank the anonymous reviewers for their valuable comments. Funding for this work was provided by NOAA to The Cooperative Institute For Marine Ecosystems and Climate (CIMEC) under award NA15OAR4320071.

References

[1] Inigo Alonso, Matan Yuval, Gal Eyal, Tali Treibitz, and Ana C Murillo. Coralseg: Learning coral segmentation from

sparse annotations. *Journal of Field Robotics*, 36(8):1456–1477, 2019. 4

[2] O. Beijbom, P.J. Edmunds, C. Roelfsema, J. Smith, D.I. Kline, B.P. Neal, M.J. Dunlap, Moriarty V., T. Fan, C. Tan, S. Chan, T. Treibitz, A. Gamst, G.B. Mitchell, and D. Kriegman. Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation. *PLOS One*, July 2015. 2

[3] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *CVPR*, 2012. 2, 4, 7

[4] Michael Bewley, Ariell Friedman, Renata Ferrari, Nicole Hill, Renae Hovey, Neville Barrett, Ezequiel M Marzinelli, Oscar Pizarro, Will Figueira, Lisa Meyer, et al. Australian sea-floor survey data, with images and expert annotations. *Scientific data*, 2(1):1–13, 2015. 4

[5] Leo Breiman. Random forests. *Machine learning*, 2001. 5, 6, 7, 8

[6] Tony F Chan, Gene H Golub, and Randall J LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. In *COMPSTAT 1982 5th Symposium held at Toulouse 1982*, 1982. 5, 6, 7, 8

[7] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 2002. 7

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995. 5, 6, 7, 8

[9] Courtney S Couch, John HR Burns, Gang Liu, Kanoelani Steward, Tiffany Nicole Gutlay, Jean Kenyon, C Mark Eakin, and Randall K Kosaki. Mass coral bleaching due to unprecedented marine heatwave in papahānaumokuākea marine national monument (northwestern hawaiian islands). *PloS one*, 12(9):e0185121, 2017. 1

[10] Glenn De'ath, Katharina E Fabricius, Hugh Sweatman, and Marji Puotinen. The 27–year decline of coral cover on the great barrier reef and its causes. *Proceedings of the National Academy of Sciences*, 109(44):17995–17999, 2012. 1

[11] Clinton B Edwards, Yoan Eynaud, Gareth J Williams, Nicole E Pedersen, Brian J Zgliczynski, Arthur CR Gleason, Jennifer E Smith, and Stuart A Sandin. Large-area imaging reveals biologically driven non-random spatial patterns of corals at a remote reef. *Coral Reefs*, 36(4):1291–1305, 2017. 2

[12] Susan A English, V J Baker, and Clive R Wilkinson. Survey manual for tropical marine resources. http://catalog.hathitrust.org/api/volumes/oclc/37934814.html, 1999. 1

[13] Katharina E Fabricius. Effects of terrestrial runoff on the ecology of corals and coral reefs: review and synthesis. *Marine pollution bulletin*, 50(2):125–146, 2005. 1

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 5

[15] Anabel Gómez-Ríos, Siham Tabik, Julián Luengo, ASM Shihavuddin, Bartosz Krawczyk, and Francisco Herrera. Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation. *Expert Systems with Applications*, 118:315–328, 2019. 4

[16] Manuel Gonzalez-Rivero, Oscar Beijbom, Alberto Rodriguez-Ramirez, Dominic EP Bryant, Anjani Ganase, Yeray Gonzalez-Marrero, Ana Herrera-Reveles, Emma V Kennedy, Catherine JS Kim, Sebastian Lopez-Marcano, et al. Monitoring of coral reefs using artificial intelligence: A feasible and cost-effective approach. *Remote Sensing*, 12(3):489, 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4, 5, 6, 7, 8

[18] Brian M Hopkinson, Andrew C King, Daniel P Owen, Matthew Johnson-Roberson, Matthew H Long, and Suchendra M Bhandarkar. Automated classification of three-dimensional reconstructions of coral reefs using convolutional neural networks. *PloS one*, 15(3):e0230671, 2020. 4

[19] Terry P Hughes, Kristen D Anderson, Sean R Connolly, Scott F Heron, James T Kerry, Janice M Lough, Andrew H Baird, Julia K Baum, Michael L Berumen, Tom C Bridge, et al. Spatial and temporal patterns of mass bleaching of corals in the anthropocene. *Science*, 359(6371):80–83, 2018. 1

[20] Terry P Hughes, James T Kerry, Andrew H Baird, Sean R Connolly, Andreas Dietzel, C Mark Eakin, Scott F Heron, Andrew S Hoey, Mia O Hoogenboom, Gang Liu, et al. Global warming transforms coral reef assemblages. *Nature*, 556(7702):492–496, 2018. 1

[21] Andrew King, Suchendra M Bhandarkar, and Brian M Hopkinson. A comparison of deep learning methods for semantic segmentation of coral reef survey images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1394–1402, 2018. 4

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Kevin E Kohler and Shaun M Gill. Coral point count with excel extensions (cpce): A visual basic program for the determination of coral and substrate coverage using random point count methodology. *Computers & geosciences*, 32(9):1259–1269, 2006. 2

[24] Daniel Langenkämper, Martin Zurowietz, Timm Schoening, and Tim W Nattkemper. Biigle 2.0-browsing and annotating large marine image collections. *Frontiers in Marine Science*, 2017. 2, 4

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 8

[26] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous Sohel, Farid Boussaid, Renae Hovey, Gary Kendrick, and Robert B Fisher. Coral classification with hybrid feature representations. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 519–523. IEEE, 2016. 4

[27] Ammar Mahmood, Mohammed Bennamoun, Senjian An, Ferdous A Sohel, Farid Boussaid, Renae Hovey, Gary A Kendrick, and Robert B Fisher. Deep image representations for coral image classification. *IEEE Journal of Oceanic Engineering*, 44(1):121–131, 2018. 4

[28] J Miller, E Muller, C Rogers, R Waara, A Atkinson, KRT Whelan, M Patterson, and B Witcher. Coral disease following massive bleaching in 2005 causes 60% decline in coral cover on reefs in the us virgin islands. *Coral Reefs*, 28(4):925, 2009. 1

[29] Scott Miller. Automatically annotating 175,000+ images with the CoralNet API. *CoralNet Blog*, Sept. 2020. 4

[30] Md Modasshir and Ioannis Rekleitis. Enhancing coral reef monitoring utilizing a deep semi-supervised learning approach. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1874–1880. IEEE, 2020. 4

[31] Benjamin Paul Neal, Adi Khen, Tali Treibitz, Oscar Beijbom, Grace O'Connor, Mary Alice Coffroth, Nancy Knowlton, David Kriegman, B Greg Mitchell, and David I Kline. Caribbean massive corals not recovering from repeated thermal stress events during 2005–2013. *Ecology and Evolution*, 7(5):1339–1353, 2017. 2

[32] Dustan P., Leard J., Meier O., Brill M., and Kosmynin V. Pointcount99 software. http://www.cofc.edu/~coral/pc99/pc99.htm, 1999. 2

[33] G Pavoni, M Corsini, M Callieri, M Palma, and R Scopigno. Semantic segmentation of benthic communities from orthomosaic maps. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019. 4

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011. 6

[35] Alberto Rodriguez-Ramirez, Manuel González-Rivero, Oscar Beijbom, Christophe Bailhache, Pim Bongaerts, Kristen T Brown, Dominic EP Bryant, Peter Dalton, Sophie Dove, Anjani Ganase, et al. A contemporary baseline record of the world's coral reefs. *Scientific data*, 7(1):1–15, 2020. 4

[36] ASM Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur CR Gleason, and Brooke Gintert. Image-based coral reef classification and thematic mapping. *Remote Sensing*, 5(4):1809–1841, 2013. 4

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 4, 5, 6, 7

[38] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. International Society for Optics and Photonics, 2019. 6

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 4, 5, 6, 7, 8

[40] Tali Treibitz, Benjamin P Neal, David I Kline, Oscar Beijbom, Paul LD Roberts, B Greg Mitchell, and David Kriegman. Wide field-of-view fluorescence imaging of coral reefs. *Scientific reports*, 5(1):1–9, 2015. 4

[41] Vasilis Trygonis and Maria Sini. photoquad: a dedicated seabed image processing software, and a comparative error analysis of four photoquadrat methods. *Journal of Experimental Marine Biology and Ecology*, 424:99–108, 2012. 2

[42] I. D. Williams, C. Couch, O. Beijbom, T. Oliver, B. Vargas-Angel, B. Schumacher, and R Brainard. Leveraging automated image analysis tools to transform our capacity to assess status and trends on coral reefs. *Frontiers in Marine Science*, 2019. 4

[43] Martin Zurowietz, Daniel Langenkämper, Brett Hosking, Henry A Ruhl, and Tim W Nattkemper. Maia—a machine learning assisted image annotation method for environmental monitoring and exploration. *PloS one*, 2018. 4