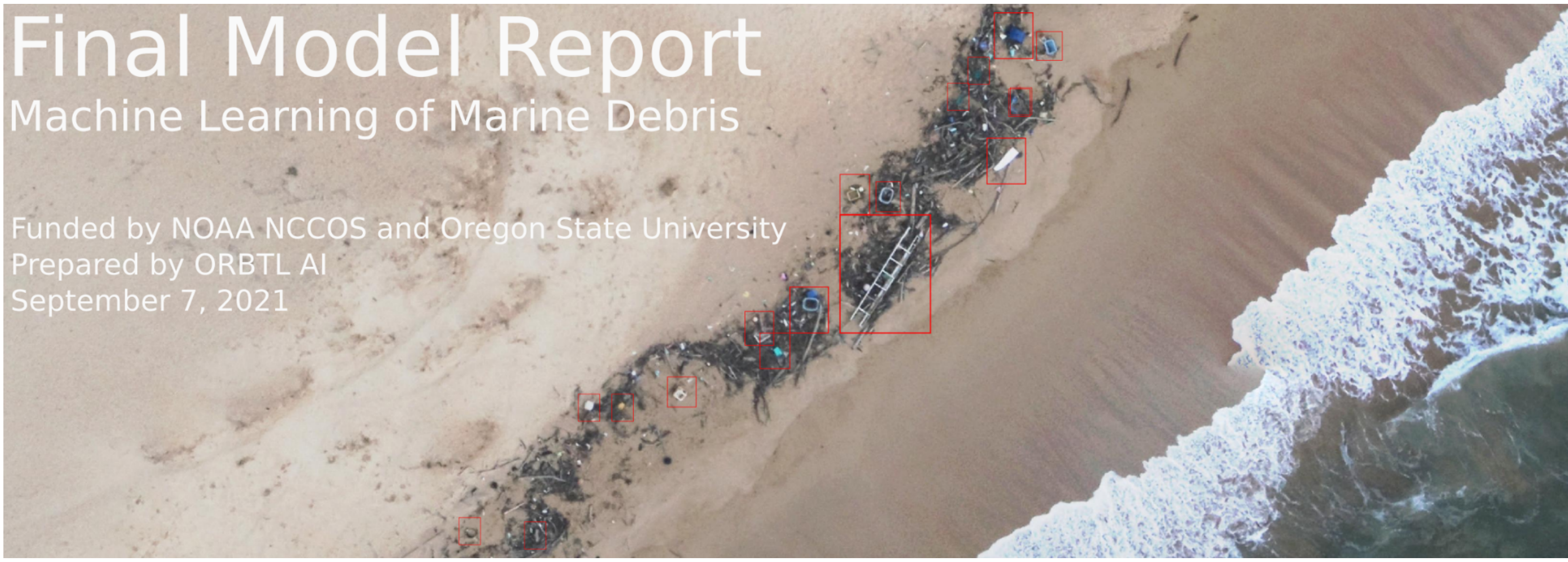

Final Model Report

Machine Learning of Marine Debris

Funded by NOAA NCCOS and Oregon State University
Prepared by ORBTL AI
September 7, 2021

Table of Contents

Introduction

If applied correctly, machine learning (and the subfield of deep learning) are a powerful technology for automating tedious tasks previously performed by hand. Many industries, including coastal management, have begun to see machine learning (ML) as a powerful tool to standardize and speed data collection (Winans 2021). This document outlines the training and evaluation procedures for a deep learning-based object detection model intended to detect large shoreline stranded marine debris (>20 cm) in remotely sensed imagery. The model requires the input imagery to have a ground spacing distance (GSD) of 2 cm, which is a typical resolution of imagery collected from crewed aircraft or uncrewed aerial systems. Given these parameters, this model should be capable of detecting shoreline stranded macro- to mega-debris at the regional-to-global scale.

This document, the trained model, and all associated data, reports, etc. were funded by the National Oceanic and Atmospheric Administration (NOAA) and Oregon State University as a component of the project, *Using Unmanned Aircraft Systems, Machine Learning, and Polarimetric Imaging to Develop a System for Enhanced Marine Debris Detection and Removal*[1]. The trained model, associated code, and data are considered "free and open-source" for public reuse, remix, and redistribution. Associated code for interacting with the trained model can be found at the project's Github repository[2].

Key Challenges

The key challenges to deep learning-based object detection of marine debris objects are:

1. Marine debris objects tend to be small in size relative to the entire shoreline. Typically marine debris objects only occupy a small fraction of pixels within aerial images of shorelines.
2. Marine debris objects have a high variation in size, shape, color, and material. It is very difficult to say a single type of debris is "representative" of all potential marine debris.
3. Aerial images of shorelines are highly complex scenes with multiple land cover types (ocean, wet sand, dry sand, rock, dune, vegetation, etc.) and many naturally occurring non-debris objects (driftwood, footprints, beach-goers, etc.). Any model must be able to function over a wide range of geographies and imagery scenes.

Solutions

1. Crop input images to 512x512 pixels before performing object detection. This preserves the size and shape of marine debris objects while increasing the ratio of object pixels to background pixels in each individual image crop.

---

[1] https://coastalscience.noaa.gov/project/using-unmanned-aircraft-systems-machine-learning-and-polarimetric-imaging-to-develop-a-system-for-enhanced-marine-debris-detection-and-removal/

[2] https://github.com/orbtl-ai/md-ml-api

2. Choose an object detection framework with multi-scale feature maps. In such frameworks the multiple feature maps are able to identify objects of various sizes.
3. Robust data augmentation is applied to the training data set images. These operations will flip images, change their color, or otherwise change the image in some systematic way. The new “augmented” scenes are then used to artificially increase the size of the training data set. This helps the model make the most from the labeled training data.
4. Fine-tuning of the object detector’s aspect ratios during the object detector’s initial anchor box proposal phase. The aspect ratios are set based on values commonly seen in the training and evaluation data. This allows long, irregularly shaped objects to be localized properly within the aerial images.
5. Initialize our training from state-of-the-art object detectors that are pre-trained on much larger data sets (a technique commonly referred to as “transfer learning”). By transfer learning we save time during the training process, allowing more iteration and experimentation.

EfficientDet-d0 Object Detection Framework

After many experiments and research hours, the best overall object detection framework is **EfficientDet-d0**. This framework was selected based on high performance across many complex object detection challenges, high efficiency, high performance on marine debris evaluation data sets, and the ability to fine-tune multiple parameters of the model to suit multi-scale geospatial object detection.

The EfficientDet (Tan et al. 2020) architecture is a deep learning-based object detection whose dimensions were found by neural architecture search (NAS), an automated method of designing efficient neural networks that tends to outcompete the traditional hand-tailored neutral architectures (Tan et al. 2020). The design of EfficientDet-d0 means the network only has to “tune” 4 million parameters; far less than the 54 million parameters to be learned in the previous best performing model on the dar2015 marine debris data set (Winans 2021).

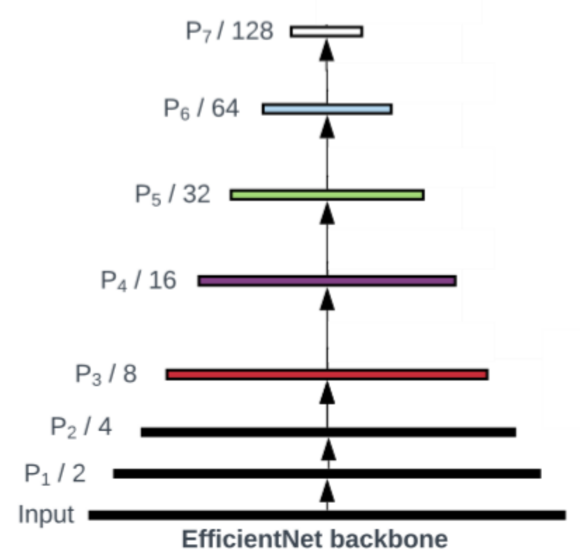

Figure 1: The multi-scale feature maps that form the backbone of EfficientNet/EfficientDet. The input image is of dimensions 512x512 pixels, and is subsequently resampled in each pyramid layer by a factor of 2. Note, this project’s implementation of EfficientDet-d0 only uses levels P3-P7.

In addition to efficiency, the EfficientDet-d0 architecture has many desirable traits for geospatial object detection. First, it ingests 512x512 pixel images, which is a desirable size for marine debris detection at the 2cm scale (Winans 2021). Furthermore, EfficientDet-d0 utilizes multi-scale feature maps which enable detection of variously sized objects (Figure 1).

Hyperparameters

Hyperparameters are settings that can be tuned before training a deep learning model. Poorly tuned hyperparameters can have a major impact on the performance of a deep learning model so it is important to find the right combination to maximize performance. The full set of hyperparameters for the tuned EfficientDet-d0 can be found in the associated pipeline.config file. Below we detail the key hyperparameters whose calibration had an especially large impact on our model's performance.

Transfer Learning

The model was initialized from a set of pre-trained model weights served by Google in the Tensorflow 2 Model Zoo[3]. This model contains pre-trained weights that were learned during the Microsoft Common Objects in Context Object (COCO) Detection Challenge (Lin et al. 2015). These pre-trained model weights allow our model to more quickly generalize to new types and classes of objects.

Multi-scale feature maps

The default settings were left for the multi-scale feature maps after hyperparameter tuning. Adding coarse scale feature maps did not aid in large object detection.

Anchor Box Aspect Ratios

EfficientDet begins the object detection routine by specifying a set of default "anchor boxes" spread across the image. These anchor boxes come in all shapes and sizes. The ratio of an anchor box's height and width is it's aspect ratio, and we can control our default anchor boxes' aspect ratio during object detection.

Below in Figure 2 we have the count of each anchor box in our training data, which gives an idea of a real-world range of aspect ratios for marine debris objects. The aspect ratios of our training data largely cluster around 1.0 (a perfect square). However, the aspect ratio can be as high as 15 and as low as 0.02, which represent very skinny horizontal and vertical rectangles. Therefore, we modify our input aspect_ratio hyperparameters from the values of [0.5, 1.0, 2.0] to a larger range designed to properly capture the complete range of aspect ratios seen in our training data set **[0.1, 0.3, 0.5, 1.0, 2.0, 4.0, 7.0, 15.0]**.

---

3 https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

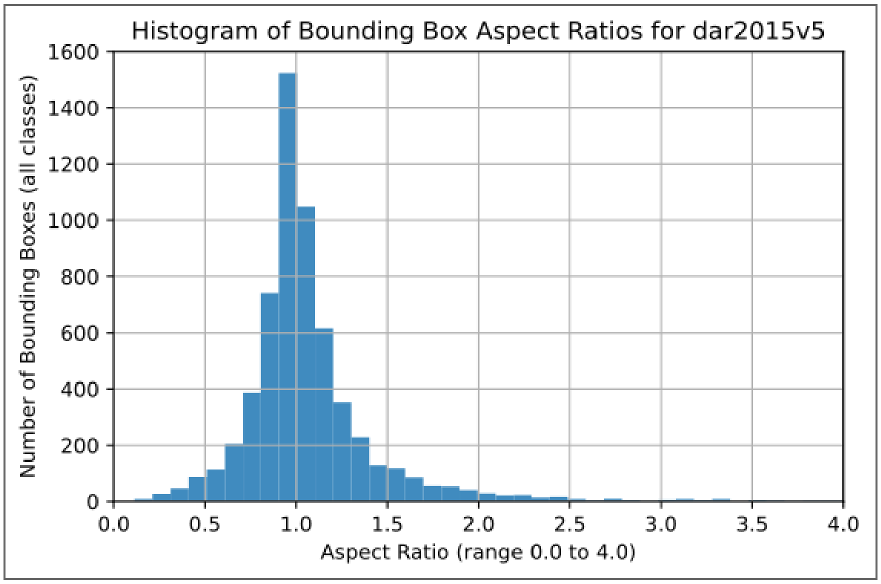

Figure 2: A histogram showing the peak aspect ratios of our training bounding box dataset. Not shown on this histogram are several extreme outliers in the 7.0-15.0 range that cannot be easily seen on a histogram plot.

Batch Size

Batch size controls the number of images loaded into the computer's memory on each training step. A higher batch size is generally preferable as it allows for higher throughput and more stable training. Batch size should be raised to the highest possible number that will fit within the computer's memory.

The final EfficientDet-d0 model detailed in this report utilized a **batch size of 16 images** when training on a NVIDIA Tesla T4 Graphics Processing Unit (GPU) with 15 gigabytes of VRAM. If this model is ever retrained on a different GPU in the future the batch size hyperparameter will need to be raised or lowered based on the future GPU's available VRAM.

Training and Evaluation Data

Source Data

The source imagery data for this project was collected in 2015 and provided by the State of Hawaii and can be found at the Hawaii Statewide GIS Program[4]. Resource Mapping Hawaiʻi performed the initial collection of the imagery data under funding from the Government of Japan PISCES. The initial marine debris annotations were digitized by Moy et al. 2017, adapted for deep learning by Winans 2021, and further improved under this project. Please refer to these two studies for in-depth processing history on the source labels and imagery.

Data Counts

The model was trained on 5,733 hand-labeled marine debris objects spread across 1,224 aerial

[4] http://geodata.hawaii.gov/arcgis/rest/services/SoH_Imagery/Coastal_2015/ImageServer

image chips of the Hawaiian Islands. The model was evaluated on 993 hand-labeled marine debris objects spread across 184 aerial image chips of the Hawaiian Islands. The training data was used to create the associated Tensorflow saved_model.pb file, while the evaluation data was used to calculate the saved model's performance (detailed in the Model Performance section below). Tables 1 and 2 below provide a per-class count of marine debris object labels.

Table 1: Per-class counts of marine debris objects in the dar2015v6 training dataset.

TRAINING DATA COUNTS		
1	unidentified fragment	2426
2	plastic object	588
3	buoy	1269
4	fishing net	333
5	fishing line	211
6	metal	232
7	tire donut	298
8	wood board	334
9	wood pallet	17
10	vessel	25
	TOTAL:	**5733**

Table 2: Per-class counts of marine debris objects in the dar2015v6 evaluation dataset.

EVALUATION DATA COUNTS		
1	unidentified fragment	351
2	plastic object	103
3	buoy	238
4	fishing net	55
5	fishing line	28
6	metal	36
7	tire donut	72
8	wood board	61
9	wood pallet	24
10	vessel	25
	TOTAL:	**993**

Data Augmentation

Data augmentation flips, crops, distorts, the color, or otherwise changes the training images to create "synthetic images" and artificially enlarge our training data set. This is a common technique in deep learning, which greatly benefits from large training data sets.

The trained EfficientDet-d0 utilizes a full suite of data augmentations empirically proven to enhance geospatial object detection:

- **random_horizontal_flips**: flips the image horizontally, so objects that initially faced east in the image now face west.
- **random_vertical_flips**: flips the image vertically, so objects that initially faced north in the image now face south.
- **random_rotation90**: randomly rotates the image between 0 and 90 degrees.
- **random_distort_color**: randomly distorts the color of an image, which helps the model be invariant to color.

- **random_scale_crop_and_pad_to_square**: randomly changes the scale of the image, cropping and padding the re-scaled image so the shape of objects are preserved. This benefits EfficientDet-d0's multiscale feature maps.

Classification Scheme

The current model will make marine debris predictions according to a streamlined 10-class scheme. The streamlined class scheme is designed to balance current technological capabilities with real world management needs. See Table 3 below for a detailed breakdown of the new streamlined 10-class scheme (called "v6" in this documentation) and how it differs from the previously tested and more complicated classification scheme.

Table 3: A crosswalk from the original 16-class v5 class scheme to the streamlined 10-class scheme deployed in this project's final model. This crosswalk shows which classes were binned together to produce a model with the least amount of class confusion while still providing as much detail as possible for marine managers.

	v5 Class Scheme		**v6 Class Scheme**	**v6 Description**
1	Unidentified Fragment	1	Unidentified Fragment	Indiscript fragments of objects. Predominantly plastic, foam, or cloth; but could be any man-made material.
4	Foam			
7	Cloth or Tarp			
14	Wood Fragment			
16	Vessel Personal			
2	Plastic Object	2	Plastic Object	Whole plastic objects such as large water jugs, beach toys, etc.
3	Buoy	3	Buoy	Fishing buoys of all shapes, sizes, and colors. Typically round but occasionally oblong. With and without tails.
5	Fishing Net	4	Fishing Net	Entangled fishing nets of all sizes and colors. Typically green.
6	Line Bundle	5	Fishing Line	Large bundles or individual fragments of fishing line of every size and color. Typically white.
8	Line Fragment			
9	Metal	6	Metal	Metal objects. Whole and fragment. Often rusted.
10	Tire Donut	7	Tire	A whole, intact tire (with or without inner rim). Could also contain obvious tire fragments (typically in crescent-moon shape).
11	Tire Fragment			
12	Wood Board	8	Wood Board	An individual piece of plywood lumber (i.e. 2x4). The defining trait of Wood Board is long, rectangular shape, even edges, and hard corners.
13	Wood Pallet	9	Wood Pallet	A wooden shipping pallet. Whole and fragment.
15	Vessel	10	Vessel	A large, powered vessel such as a boat.

The motivation behind the v6 classification scheme was to consolidate classes which simply lack a large enough number of training labels to support reliable classification. Also, the current model only utilizes three bands of spectral information (Red, Green, and Blue). It has been empirically shown that both humans and machines have trouble reliably distinguishing materials from true color imagery alone (Winans 2021). For these reasons, we are attempting to implement a classification scheme that is as reliable as possible while providing sufficient detail for management purposes. See the “Performance by Class” below for more information about the class-by-class performance of the trained EfficientDet-d0 model on an evaluation dataset.

Model Performance

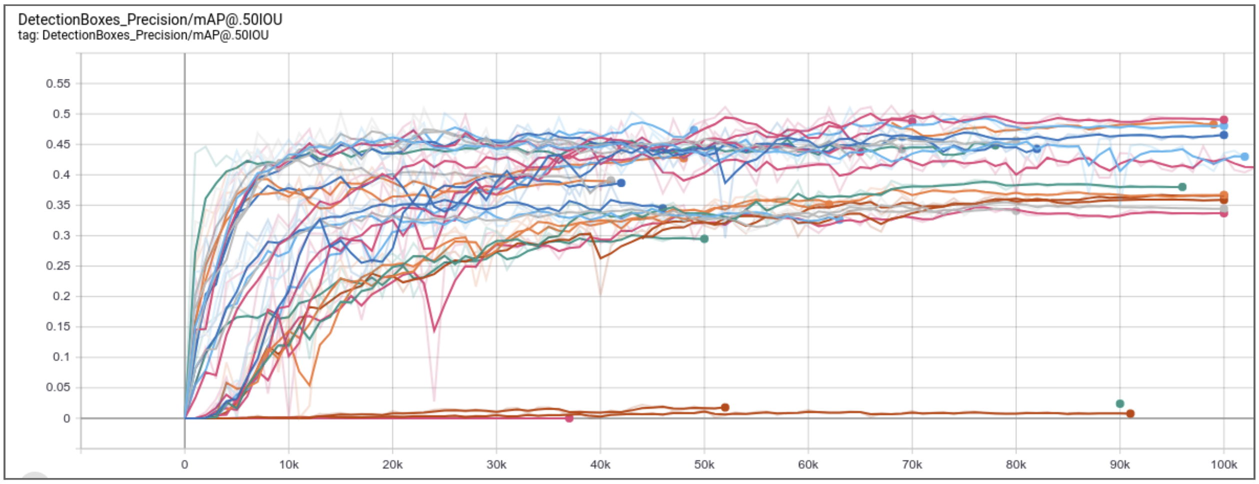

Figure 3: A line plot of every experimental model trained in the course of ORBTL AI’s experiments. Overall over 30 models were trained, evaluated, and documented to arrive at the final model detailed below.

Quantitative Results

Accurately describing the benefits and drawbacks of a multi-class object detection model can be difficult, as we are dealing with both localization and classification of objects within an image. Therefore it is often helpful to use compound metrics which summarize the overall performance of an object detector. Here are the primary metrics used to gauge the real-world performance of the trained model:

- **Intersection Over Union (IOU)** is used to determine whether a prediction bounding box properly localizes an object during evaluation. Our project utilizes a 50% IOU threshold (0.5), this means our prediction must fit the real-world object’s shape and size.
- **Mean average precision (mAP)** is often used to describe the relevance of our model’s positive predictions averaged over a series of recall values. This is the primary compound metric used to evaluate overall model performance.
- **Average recall for 100 detections (AR@100)** computes the rate of positive detections over all possible positive predictions. This metric is computed with up to 100 of the highest confidence predictions in each photo.

Table 4 provides details of our model’s key performance measures at the top performing training checkpoint.

Table 4: The two key quantitative measures of our model's performance at the best training checkpoint (44,000).	
Metric	**Value at model checkpoint 44,000**
mAP@0.5IOU	0.4914
AR@100	0.4943

Performance by Object Size

The addition of additional multi-scale feature maps improves EfficientDet-d0's performance on small and medium object detection localization. However, EfficientDet-d0's performance on large objects still lags that of other object detection frameworks, such as Faster R-CNN. This tradeoff is worthy, as a majority of the marine debris objects in the data set fall in the small and medium size categories. It is also notable that large object classes tend to contain the least amount of labels, meaning it is possible that large object performance is constrained by the smaller than average training data set. See Table 5 below for a full description of the object size categories.

Table 5: A summarization of mAP@0.5IOU across the three object size categories. The mAP values describe how relevant each prediction is across each size class.	
Size Class	**mAP@0.5IOU**
Small Object (0.4 m^2 or smaller)	.2064
Medium Object (0.4 m^2 - 3.7 m^2)	.2966
Large Object (3.7 m^2 or larger)	.1932

Performance by Class

During final evaluation (and on subsequent evaluations of new data) we can specify the confidence threshold and IOU threshold at which to keep the model's predictions. After empirical study it was found that an IOU threshold of 0.5 (50%) and a confidence threshold of 0.3 (30%) provide the best balance in precision and recall rates. See Table 6 below for a detailed breakdown of per-class precision and recall scores.

Table 6: The precision and recall scores for each class during the final model evaluation. Only model predictions that exceeded 50% IOU and 30% confidence were kept.

	class	precision@ 0.5IOU/0.3CONF	recall@ 0.5IOU/0.3CONF
1	unidentified fragment	0.40	0.56
2	plastic object	0.51	0.33
3	buoy	0.65	0.54
4	fishing net	0.69	0.73
5	fishing line	0.40	0.36
6	metal	0.25	0.42
7	tire donut	0.72	0.85
8	wood board	0.34	0.34
9	wood pallet	0.67	0.50
10	vessel	0.52	0.48
	Average Precision/Recall:	**0.51**	**0.51**

False positives occur when the model flags a natural object as marine debris. False negatives occur when the model misses a piece of marine debris. The false positive rate tends to increase as the confidence threshold is lowered, while the number of false negatives drop. The opposite occurs when the confidence threshold is raised. Therefore, the confidence threshold may be lowered if it is desirable to lean towards overcounts of marine debris. Conversely, the confidence threshold can be raised if it is desirable to lean towards undercounts of marine debris. Overall there were 350 false positives and 254 false negatives at the 30% confidence level.

Further, certain objects may be localized correctly in the imagery but their classification may be wrong (i.e. a plastic object is misidentified as a metal object). These occurrences are referred to as class confusion, which negatively impacts model performance metrics. Below is a class confusion matrix, which shows both the total counts and percentages of classified objects (Figure 4). Note that Figure 4's right column and bottom row both represent an eleventh "nothing" category. This category only exists in the confusion matrix for the purpose of displaying false positive and false negative rates on a class-by-class basis.

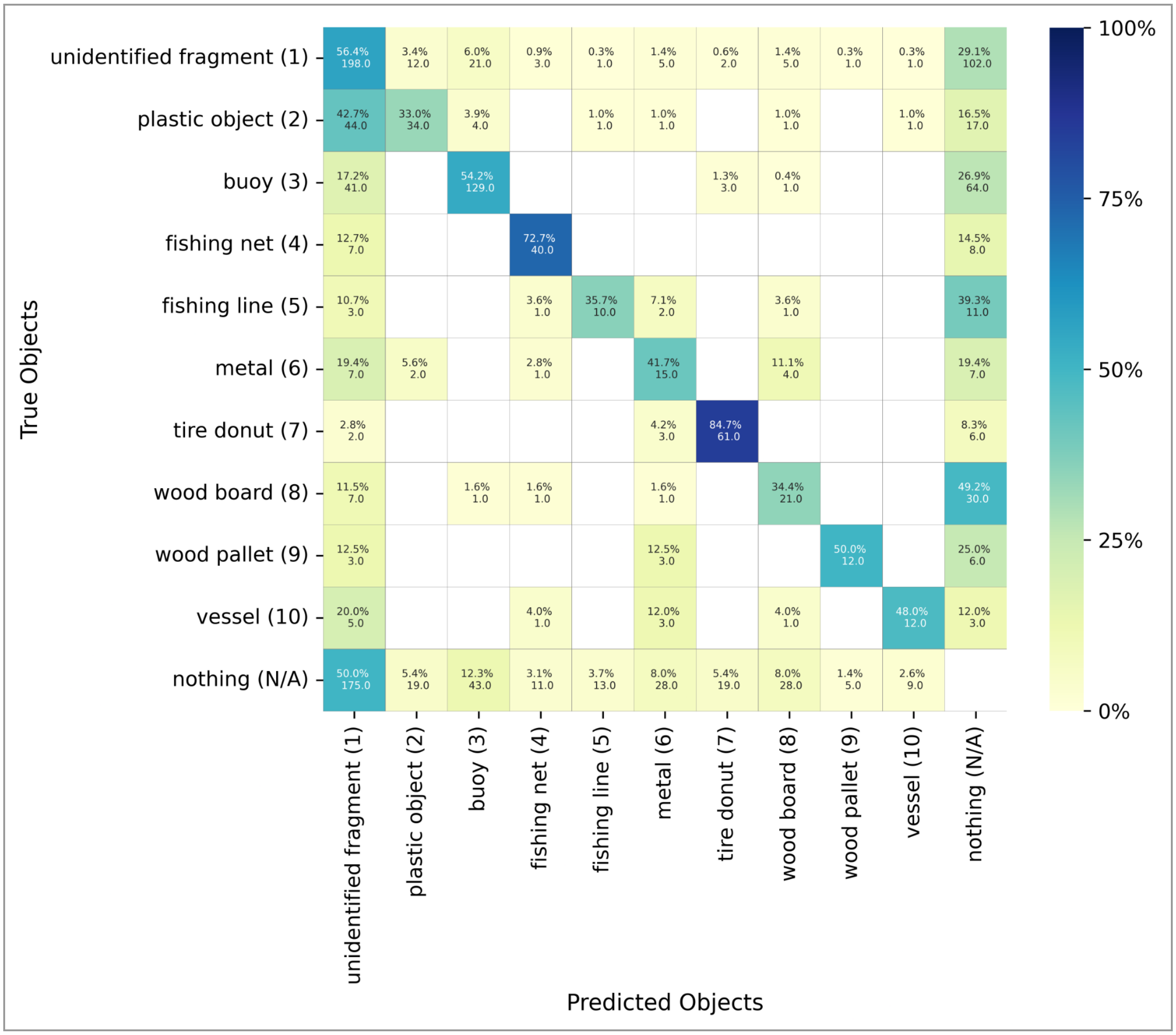

Figure 4: A class confusion matrix showing the final count and percentages of correctly and incorrectly classified objects per class. Note that the right column and bottom row represent false negatives and false positives respectively.

Qualitative Results

The associated evaluation-plots/ folder contains plots of all 184 evaluation images with model predictions. In the plots the red boxes are ground truth, while the predictions are colored by class with class name and confidence score included. These plots can be used to get a qualitative feel for the model's performance. See Figure 5 below, which is an example of an evaluation plot.

Figure 5: Evaluation plot with model predictions for the image maui_16644_31_40.jpg. Ground truth (hand labels) are drawn in red.

Loading the Model

The model was trained and evaluated in Tensorflow Object Detection API v2.4 (Huang et al. 2017). The format of the saved model is the Tensorflow saved_model format. Refer to the official Tensorflow documentation for further instructions on loading the saved model[5] to run evaluation jobs.or fine-tune the training.

[5] https://www.tensorflow.org/guide/saved_model

Citations

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv:1611.10012 [cs].

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P., 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs].

Moy, K., Neilson, B., Chung, A., Meadows, A., Castrence, M., Ambagis, S., Davidson, K., 2018. Mapping coastal marine debris using aerial imagery and spatial analysis. Marine Pollution Bulletin, SI: Japanese Tsunami Debris 132, 52–59. https://doi.org/10.1016/j.marpolbul.2017.11.045

Tan, M., Pang, R., Le, Q.V., 2020. EfficientDet: Scalable and Efficient Object Detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079

Winans, W.R., 2021. Automatic Detection of Hawai’i’s shoreline stranded mega-debris using deep learning-based object detection (Master thesis). University of Hawaii at Manoa, Honolulu, HI. https://scholarspace.manoa.hawaii.edu/bitstream/10125/75937/Winans_hawii_0085O_11020.pdf