Scaling Performance Evaluation on FathomNet

Jakeb Milburn

Abstract

This project evaluates how a Faster R-CNN model, pretrained and fine-tuned, scales in performance on FathomNet, a large-scale marine dataset, with respect to dataset size and image resolution. The goal is to assess the impact of these scaling factors on accuracy (measured as Mean Average Precision, mAP), training time, GPU Usage during training, and inference time. Three hypotheses were tested: (1) increasing dataset size yields diminishing returns in accuracy while linearly increasing training time, (2) higher image resolution improves accuracy but significantly increases GPU memory usage, and (3) inference time is minimally impacted by resolution or model size. Experimental results provide partial validation of these hypotheses, revealing key findings about the relationships between dataset size, image resolution, and computational efficiency. While accuracy improved with both larger datasets and higher resolutions, significant trade-offs were observed in terms of increased training time and expected GPU usage. These findings highlight the challenges of balancing performance gains with resource constraints.

Introduction

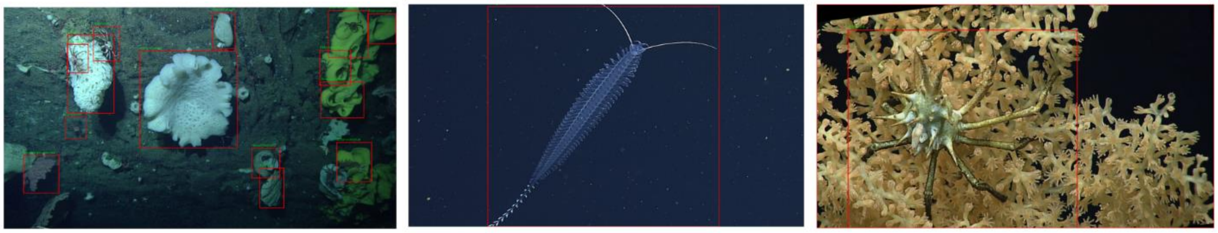
Marine ecosystems are under unprecedented stress from human activity, necessitating scalable and efficient tools for monitoring and conservation. Machine learning has emerged as a critical approach for processing and analyzing vast datasets in real-world applications, including ecological monitoring. FathomNet, an open-source dataset containing over 110,000 labeled images of marine species, provides a unique opportunity to develop object detection models capable of identifying marine life at scale. However, the computational demands of training and deploying such models necessitate careful consideration of scalability in terms of dataset size, image resolution, and computational resources.

This study leverages the Faster R-CNN framework to explore the challenges and opportunities of large-scale machine learning in marine ecology. Fine-tuning was applied to adapt the pretrained Faster R-CNN model to the FathomNet dataset. This approach enabled the model to learn domain-specific features while retaining the advantages of pretrained weights, balancing computational efficiency and performance.

The analysis focuses on scaling effects, investigating how variations in dataset size and resolution impact model accuracy, training time, and resource usage. By focusing on four ecologically significant classes—fishes, sponges, crinoids, and corals—the study ensures a meaningful benchmark for evaluating scalability. The findings contribute to the broader field of large-scale machine learning by addressing practical constraints while advancing scalable solutions for ecological monitoring and conservation.

Dataset and Class Selection

This study uses data from FathomNet, an open-source dataset containing over 110,000 images of marine species annotated with bounding boxes. FathomNet is specifically designed to support machine learning applications in marine science, offering high-quality labeled imagery for training and evaluation. The images are 1920x1080 pixels with a 16:9 aspect ratio, making them well-suited for object detection tasks that require high-resolution inputs.



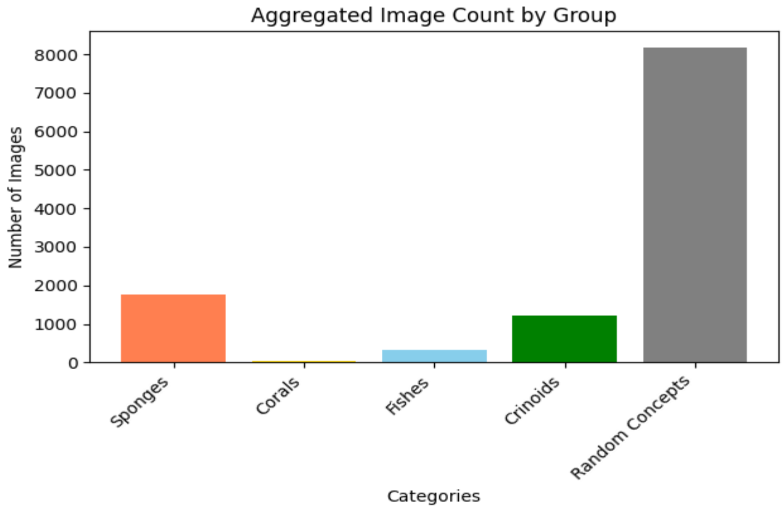
Example images and bounding boxes from the FathomNet dataset.

For this study, four classes were selected from a related work based on their ecological significance as indicators of vulnerable marine ecosystems, and the ability to compare these models to a baseline model trained on the same classes. These classes were identified using their scientific class labels, grouped as follows:

- **Sponges**: Includes the classes *Demospongiae*, *Hexactinellida*, *Calcarea*, and *Homoscleromorpha*.
- **Corals**: Corresponds to the class *Anthozoa*.
- **Fishes**: Includes the classes *Agnatha*, *Chondrichthyes*, *Osteichthyes*, *Sarcopterygii*, and *Actinopterygii*.
- **Crinoids**: Represents the class *Crinoidea*.

For each dataset configuration, the images were distributed such that 20% of the data came from each of these four categories, with an additional 20% allocated to a fifth category representing images that did not contain any of these classes. This fifth category was added to reflect the diversity of marine environments captured in FathomNet and to account for real-world deployment scenarios where non-target objects or empty scenes may be present.

However, the data for the four selected classes was highly imbalanced and insufficient to meet the required number of images for the dataset.



Distribution of classes before dataset expansion

To address this, image augmentation techniques were applied to artificially expand the dataset and reach the 20% allocation for each class. These augmentations included the following operations:

- **Horizontal Flip**: Applied with a probability of 50% to simulate natural variability in orientation.
- **Affine Transformations**: Random rotations between -20 and 20 degrees to mimic different viewing angles.
- **Brightness Adjustments**: Multiplying pixel values by a random factor between 0.8 and 1.2 to simulate changes in lighting conditions.
- **Random Cropping**: Cropping up to 10% of the image to introduce variability in framing.

Problem Formulation

The object detection task is formalized as follows:

1. **Input**: $X = \{x_1, x_2, \ldots, x_n\}$, where x_i is an image of dimensions $h \times w$.
2. **Output**: $Y = \{y_1, y_2, \ldots, y_n\}$, where $y_i = \{(b_{ij}, c_{ij}) \mid j = 1 \ldots k_i\}$, b_{ij} are bounding boxes, and c_{ij} are class labels for the j-th object in x_i.
3. **Model**: $f_\theta : X \rightarrow Y$, where θ represents pretrained parameters fine-tuned on FathomNet.

Approach

Planned Scope and Constraints

Initially, the experimental design aimed to include multiple object detection architectures, specifically YOLO and SSD, to evaluate their performance and scalability alongside Faster R-CNN. These architectures represent state-of-the-art approaches that are optimized for real-time applications (YOLO) and lightweight deployment (SSD). Testing these models would have provided a comprehensive comparison and deeper insights into architectural trade-offs.

Additionally, the study sought to investigate **class scaling**, which involves varying the number of detected classes to assess how model performance and computational costs are affected. This would have extended the scope to explore the impact of increasing ecological complexity on detection tasks.

For dataset scaling, the original plan was to use subsets of 10K, 50K, and 100K images to evaluate how performance scales with significantly larger datasets. However, due to computational constraints, particularly limited GPU resources available through Google Colab, the dataset sizes had to be scaled down to 1K, 5K, and 10K images. Similarly, testing multiple architectures, scaling classes, and using larger datasets would have required substantial computational power, memory, and time, which exceeded the resources available for this project. Consequently, the focus was narrowed to fine-tuning Faster R-CNN and analyzing scaling effects across reduced dataset sizes and resolutions for a fixed set of four classes.

Classification Model

This study utilizes the state-of-the-art Faster R-CNN framework for object detection, chosen for its robust two-stage architecture. Its region proposal network (RPN) efficiently generates candidate regions, while the subsequent classification and bounding box refinement stages ensure high detection accuracy. This makes Faster R-CNN particularly effective for tasks requiring precise object localization, such as detecting species in underwater imagery.

Faster R-CNN was selected over alternatives like YOLO and SSD due to its superior performance in scenarios where accuracy is prioritized over speed. Unlike single-stage detectors, which optimize for real-time inference by combining region proposal and classification into a single step, Faster R-CNN's two-stage process allows for more refined predictions, making it better suited for applications with complex imagery and subtle features.

Independent Variables

Dataset size scaling and image resolution scaling were chosen as independent variables because they directly influence both model performance and computational demands.

Dataset size impacts the model's ability to generalize and learn robust features, with larger datasets typically improving accuracy but incurring significant computational costs. Understanding how performance scales with dataset size helps identify the optimal balance between accuracy and resource efficiency.

Similarly, image resolution determines the level of detail available for feature extraction, where higher resolutions can enhance detection performance but drastically increase memory usage and training time. By investigating these two variables, this study addresses the trade-offs between performance and scalability, enabling more informed decisions for deploying machine learning models in large-scale ecological monitoring tasks.

These independent variables formed the basis for the following **hypotheses:**

1. **Dataset Size Scaling**: Increasing dataset size yields diminishing returns in mAP, with linear increases in training time.
2. **Image Resolution Scaling**: Higher resolutions improve mAP but significantly increase GPU memory usage during training
3. **Inference Time and Model Size**: Inference time is minimally impacted by resolution or model size

Experimental Design and Performance Metrics

Configurations

To focus the evaluation, A subset of 9 configurations across all combinations of 3 dataset sizes and 3 resolutions were tested:

- **Dataset Sizes**: 1K, 5K, and 10K images
- **Resolutions**: 480x270, 960x540, and 1920x1080

This structured evaluation allows for a systematic analysis of how performance and computational efficiency scale across dataset sizes and resolutions.

Training Split

Each dataset was split into three subsets:

- Training Set: 70% of the data, used to train the model.
- Validation Set: 10% of the data, used to tune hyperparameters and evaluate model performance during training.
- Test Set: 20% of the data, reserved for final evaluation of the model's performance.

Training Environment and Hyperparameters

The model was trained on an NVIDIA A100 GPU using Google Colab.

- **Learning Rate**: 0.005
- **Weight Decay**: 0.0005, included to prevent overfitting by penalizing large weights.
- **Momentum**: 0.9

- **Number of Epochs**: 3, providing sufficient iterations to assess the impact of dataset size and resolution without overburdening computational resources.

Performance Metrics

To comprehensively evaluate model performance and computational cost, the following metrics were employed:

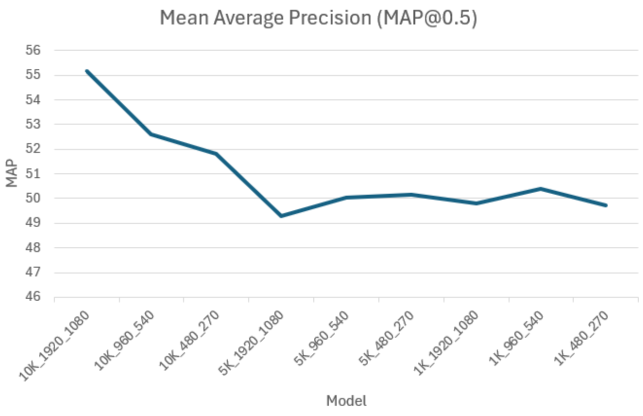
- **Mean Average Precision (mAP)**: Evaluated at IoU=0.5 to measure detection accuracy for each model.
- **Training Time**: Total time taken to train each model, measured in seconds.
- **GPU Memory Usage**: Memory utilized during the training period for each model, measured in megabytes (MB).
- **Inference Time**: Time taken to process a single image during inference, measured in seconds.

The metrics chosen for this study were selected to provide a comprehensive evaluation of both model performance and computational feasibility. mAP at IoU=0.5 is a standard metric in object detection, offering a robust measure of detection accuracy across classes. Training time was included to assess the scalability of the models with respect to dataset size and resolution, reflecting the practical considerations of time efficiency during model development. GPU memory usage was measured to evaluate the resource demands of training each configuration, as limited GPU memory can constrain model scalability and deployment feasibility. Finally, inference time was tested per image to assess the model's efficiency in real-time, providing insights into the feasibility of deploying the trained models for ecological monitoring tasks

Results

Impact of Dataset Size on Accuracy:

- **10K Dataset**: The highest mAP values were achieved with the 10K dataset across all resolutions, with a maximum of **55.2** for the 1920x1080 resolution.
- **5K and 1K Dataset**: mAP did not improve between the 1K and 5K datasets. At 480x270, the highest mAP for the 5K dataset was **50.1**, compared to **50.3** for the 1K dataset at 960x540.



The lack of significant improvement in mAP between the 1K and 5K datasets suggests that the model required the full 10K dataset to effectively learn the task. Despite extensive image augmentation, the smaller datasets likely lacked the diversity necessary to overcome the challenges posed by noisy images and subtle visual patterns in the FathomNet data. This noise and limited variability may have restricted the model's ability to generalize effectively, leading to only marginal performance gains until the larger 10K dataset provided the necessary scale to improve accuracy.

The improvement seen with the 10K dataset, where the highest mAP of 55.2 was achieved at 1920x1080 resolution, demonstrates that increasing the amount of training data can lead to better results. The additional data in the 10K dataset likely provided more unique and representative examples, helping the model better handle the inherent variability and noise in the dataset. This highlights the importance of larger datasets in leveraging the full capacity of high-performance models like Faster R-CNN, suggesting that further increases in dataset size could potentially yield even better results.

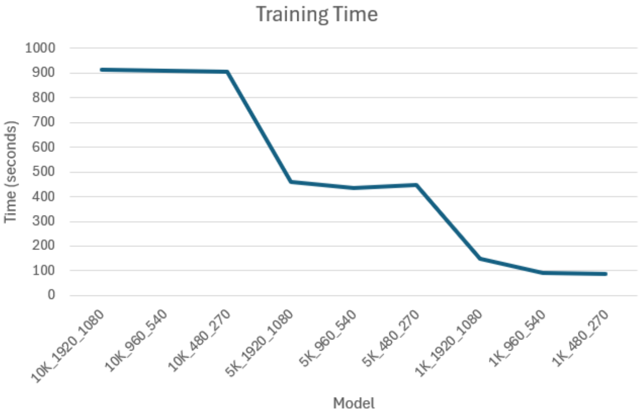
Impact of Image Resolution on Accuracy:

The impact of resolution varied significantly across dataset sizes:

- For the **10K dataset**, higher resolutions consistently improved mAP, with 3.3 mAP gain from the lowest resolution to the highest. This suggests that with sufficient training data, the model can leverage the finer details captured in high-resolution images to improve detection accuracy.
- For the **5K and 1K datasets**, higher resolutions did not consistently lead to improved mAP values. Again, it is likely that the 5k and 1k datasets did not effectively learn the task.

Impact of Dataset Size on Training Time:

- The training time was longest for the **10K dataset**, where models consistently required approximately 900 seconds across all resolutions.
- For the **5K dataset**, training times were noticeably shorter, around 450 seconds for each resolution.
- The **1K dataset** had the shortest training times, around 100 seconds for each resolution.



The training time increases linearly with dataset size, reflecting efficient scaling behavior in the training process. This linearity is a desirable property, as it ensures that computational resources are utilized predictably as the dataset grows.

Impact of Image Resolution on Training Time:

Image resolution did not have a significant impact on training time. These findings suggest that resolution alone does not significantly affect training time in large-scale tasks. This highlights the feasibility of using higher resolutions in training when sufficient data is available to justify their benefits in model performance.

Impact of Image Resolution and Dataset Size on GPU Usage

The GPU memory usage results were too varied and inconsistent to be reliable due to the nature of running experiments on Google Colab. Colab dynamically allocates GPU resources and operates in a shared environment, leading to fluctuations in available memory and resource contention from other users.

In a controlled environment with a dedicated GPU, the expected GPU memory usage would likely show:

- **Image Resolution Dependence**:
  - GPU memory usage would increase with higher resolutions, as larger image dimensions require more memory for processing and storage of intermediate tensors. For instance, **1920x1080** images would consume significantly more memory than **480x270** images.
  - Memory usage would likely scale linearly with resolution.
- **Dataset Size Independence**:
  - For a fixed batch size, dataset size would not directly impact GPU memory usage per batch. Larger datasets would only affect the total training time but not the memory footprint for a single batch.

Impact of Image Resolution and Dataset Size on Inference Time

The inference time remained constant at **0.03 seconds per image** across all models, regardless of image resolution or dataset size. The observed constant inference time demonstrates the robustness and scalability of Faster R-CNN in handling variations in resolution and dataset size. Therefore, inference time should not be a factor in determining how large a dataset should be or what resolution the images in it are.

Training Time Vs. Accuracy for Dataset Size:

Training time scaled linearly with dataset size. However, the accuracy did not improve significantly between the 1K and 5K datasets. This indicates that the additional training time required for the 5K dataset did not yield proportional gains in accuracy, likely due to insufficient data diversity or task complexity in the smaller datasets.

In contrast, the 10K dataset demonstrated a marked improvement in mAP. The longer training time for the 10K dataset proved worthwhile, as the larger dataset provided sufficient examples for the model to effectively generalize and handle the inherent noise and variability in the FathomNet data.

Training Time Vs. Accuracy for Image Resolution:

Image resolution did not significantly impact training time, as the training times for different resolutions within the same dataset size remained relatively constant. This suggests that the Faster R-CNN model processes images efficiently regardless of resolution, with the primary determinant of training time being the number of samples in the dataset.

Despite having a minimal impact on training time, higher resolutions contributed to improved accuracy for the 10K dataset. This indicates that, when sufficient data is available, higher resolutions can enhance the model's ability to extract fine-grained features without incurring additional training time.

GPU Usage Vs. Accuracy:

Although we could not obtain reliable results for GPU Usage vs. accuracy due to the variability in resource allocation on Google Colab, it is likely that GPU usage would scale predictably based on resolution and remain consistent across dataset sizes.

While there are accuracy improvements observed at higher resolutions, these gains must be weighed against the increased resource requirements. The scalability of such models in real-world deployments will depend on the availability of high-memory GPUs or the feasibility of optimizing model architectures to handle high-resolution inputs more efficiently.

On the other hand, it is likely dataset size can be increased to gain additional model accuracy without incurring additional GPU usage for a fixed batch size.

Comparison to Existing Baseline Model:

Researchers at the Monterey Bay Aquarium Research Institute trained a model on the same dataset and classes, using the Ultralytics YOLOv8x architecture, achieving a higher mAP of **0.713**. Unfortunately, the details of their experimental setup, such as dataset preprocessing, augmentation strategies, or resolution scaling, were not provided, making it challenging to determine the exact reasons for their improved performance.

The discrepancy highlights the importance of exploring alternative architectures in future work, as well as optimizing training conditions to better understand how various factors—such as dataset size, resolution, and preprocessing—contribute to performance differences.

Conclusion

The results of this study provide insights into the relationships between dataset size, image resolution, training time, GPU usage, and inference time, allowing us to evaluate the initial hypotheses:

- **Dataset Size Scaling**:
  - The hypothesis that increasing dataset size yields diminishing returns in mAP with linear increases in training time was **partially supported**.
  - Training time scaled linearly with dataset size, with the 10K dataset requiring approximately 900 seconds, compared to 450 seconds for the 5K dataset and 100 seconds for the 1K dataset.
  - However, mAP improvements were negligible between the 1K and 5K datasets, suggesting that the smaller datasets lacked the diversity necessary for effective learning. A significant improvement in mAP (from 50.1 to 55.2) was observed with the 10K dataset, demonstrating that accuracy increases with dataset size when

sufficient data is provided. This result highlights the importance of dataset scale while emphasizing that increased training time is a key constraint for scaling dataset size.

- **Image Resolution Scaling**:
    - The hypothesis that higher resolutions improve mAP but significantly increase GPU memory usage during training could not be fully validated due to unreliable GPU memory usage data. However, based on expected trends, it is likely that GPU memory usage increases predictably with higher resolutions.
    - Accuracy improved with higher resolutions, as shown by a 3.3-point gain in mAP for the 10K dataset between the lowest resolution (480x270) and the highest resolution (1920x1080). This supports the hypothesis that higher resolutions enhance detection accuracy by capturing finer details. However, the increased GPU memory requirements for higher resolutions present a key constraint for scalability, particularly in resource-limited environments.
- **Inference Time and Model Size**:
    - The hypothesis that inference time is minimally impacted by resolution or model size was **confirmed**. Inference time remained constant at 0.03 seconds per image across all resolutions and dataset sizes. This consistency demonstrates the efficiency of the Faster R-CNN architecture in handling varying input sizes and dataset scales, making it suitable for large-scale applications requiring predictable inference performance.
    - Moreover, the results show that inference time should not be a constraint for scaling either dataset size or image resolution. The model’s consistent inference time ensures scalability for large-scale deployments, making it well-suited for real-time or batch-processing scenarios regardless of dataset size or resolution.

Future Works

For future works, I would like to further increase the dataset size, as the results indicate that the model did not effectively learn until the 10K dataset. Scaling beyond 10K could reveal additional performance improvements and clarify whether diminishing returns occur at higher dataset sizes. Additionally, I plan to experiment with different object detection architectures, such as YOLO or SSD, to evaluate their efficiency and performance compared to Faster R-CNN. Exploring class scaling would also be a priority, as it could improve the model’s ability to handle imbalanced datasets and better represent underrepresented categories. Finally, using a dedicated GPU instead of Google Colab would provide consistent resources, allowing for reliable GPU usage measurements and a more accurate assessment of computational requirements.

Final Remarks

The findings demonstrate that accuracy increases with both dataset size and image resolution. However, the constraints for scaling dataset size include increased training time, while the constraints for scaling image resolution include increased GPU usage. These trade-offs highlight the importance of balancing computational resources and performance gains in large-scale machine learning applications.