Project Proposal

Towards Efficient Deep-Sea Organism Detection

2421G

October 2022

Project Originator: Dr Chris Town
Project Supervisor: Dr Chris Town, Dr Emily Mitchell
Director of Studies: Dr Chris Town
Overseers: Dr Sean Holden, Dr Neel Krishnaswami

Introduction

Deep-sea ecosystems are increasingly threatened by human activities. According to research from 2011, 19% of the world's coral reefs had been lost, with a further 75% under threat [1]. Regular monitoring of deep-sea ecosystems is key to conservation efforts. In recent years, deep dives by remotely operated underwater vehicles (ROVs) have recorded thousands of hours of video footage capturing deep-sea ecosystems. Detecting and classifying organisms found in these types of footage offers a valuable way to monitor deep-sea organisms, such as the sessile benthos, a class of immobile seabed creatures that include corals, sponges and sea anemones. However, wider adoption of this technique has been hindered by the cost of expert annotation, motivating efforts to automate this process. In particular, it will be valuable to do this computationally efficiently given the large amount of footage to analyse. A very efficient detection system could potentially also offer real-time control signals to ROVs, enabling greater dive automation.

My project aims to develop an efficient image recognition program that automatically detects and classifies sessile benthic deep-sea marine animals from video footage, outputting bounding-box localisations and organism classifications. At the core of this program lies an object detection model that I will train on the *FathomNet* [2] dataset, an open-source collection of over 78,000 annotated ocean-life images.

Challenges

Image Quality. The quality of underwater images depends on many factors. The number of visible particles in water (turbidity) can significantly affect the clarity of organisms. As very little sunlight reaches the ocean depth where many marine orgasms are found, artificial lumination is needed for photography. However, variations in lumination cause brightness to differ between images. Artificial lumination can often distort the natural colour of organisms.

High Intra-class Differences. Animal taxonomy is a complex discipline that classifies organisms based on many factors. As a result, two organisms in the same taxonomic rank often share very few physical characteristics such as colour and shape. Moreover, underwater imagery captures organisms from a variety of angles and distances, resulting in significant variation within each class. Intra-class differences also arise from different annotation techniques. For example, while some experts annotate each entire coral with a bounding box, others annotate each distinguishing feature on a coral's body.

High Inter-class Similarity. Organisms from different taxonomic classes can share similar visible features such as colour, pattern, shape and size. This has been a common problem for coral reef classifications. In addition, different classes of orgasms often live close to each other, resulting in optical occlusions and fuzzy borders between organisms.

Sparse Training Data. At higher granularity, available image data for organisms become rare. For example, while there are over 11,000 images of corals on the large *FathomNet* dataset, the number of samples per species is highly variable. Only a handful of annotated images are available for many species, genera or even families of deep-sea organisms. This significantly limits the potential granularity of classification and forces us to utilise the limited training data more intelligently.

Related Works

Many prior works used hand-picked features for deep-sea organisation classification, with varying degrees of success. Macros et al. [3] (2005) uses Normalized Chromaticity Coordinates (NCC) and Hue Saturation Value (HSV) for colour features, and Local Binary Patterns (LBP) for textual features on a classification task over 3 coral classes on 185 images, achieving 86.5% accuracy. Pizarro et al. [4] (2008) use NCC, hue-histograms and Scale-Invariant Feature Transforms (SIFT) on an image classification task over 8 classes of sessile benthos.

More recent papers have leveraged machine learning architectures such as Convolutional Neural Networks (CNNs) for automatic feature learning. Gomez-Rios et al. [5] (2018) trained various CNNs architectures on classifying corals. With transfer learning and data augmentations, they achieved 98.63% accuracy on the EILAT and RSMAS datasets. Rodrigues et al. [6] (2018) compared an AlexNet [7] (CNN) model pretrained on general images with one pretrained on images of planktons. The two models offered comparable performance at a downstream plankton classification task. This highlights the usefulness of pretrained weights on CNN tasks.

Project Methodology

My project draws significant inspiration from Gardar Ingvarsson's Part II project [8] (2021-2022). Gardar implemented a Faster-R-CNN-based deep-sea organism detector and trained it on the *FathomNet* Dataset. My project will be written from scratch and aims to improve on Gardar's implementation in three major areas:

1. **More efficient object detection architecture**.
   Faster-R-CNN is a two-stage object detection architecture [9]. It first uses a Region Proposal Network (RPN) to suggest suitable locations and sizes for bounding boxes. After this, selected region proposals are passed to a Fast-RCNN detector [10] for image classification. Due to data dependence, RPN and Fast-RCNN must be run sequentially, hence "two stages". In comparison, single-staged architectures predict bounding-boxes and object classes simultaneously. In general, one-stage architectures are more efficient than two-stage architectures but less performant. However, recent refinements in one-stage architectures have significantly closed the performance gap, with architectures such as EfficientDet [11] and You Only Look Once (YOLO) [12] achieving state of the art performance on object detection tasks

   My project seeks to improve detection efficiency by implementing and training a one-staged object detection model.

2. **Supervised Contrastive Learning.**
   Supervised contrastive learning techniques have improved performance across many image classification architectures. Intuitively, contrastive learning aims to separate different concepts' representations in latent space while bringing similar concepts together. Supervised contrastive learning takes advantage of supervised labels to define similar and different concepts. Supervised contrastive learning could mitigate the challenge of high inter-class similarity, as mentioned previously.

3. **Improved Data Augmentation**
   Image augmentation is a necessary component of contrastive learning and a popular technique to increase training data. Gardar's implementation includes horizontal flipping, image rotation and hue/brightness jitter. I will expand these with gaussian noise injection and random erasing [14]. I will also create a training data generator that overlays cut-outs of organisms into empty spaces in different images. The cut-outs will be generated from existing training images using segmentation techniques such as GrabCut [15] or U-Net [16]. The colours and size of the cut-outs will be adjusted to blend into their new backgrounds.

My supervisor, who is also my Director of Studies, has also supervised Gardar Ingvarsson in his project and is satisfied that this project is sufficiently different from his.

Project Structure

The core deliverable of the project is a program that can automatically detect and annotate deep-sea organisms captured in video footage. The organisms will be classified into at least 10 superclasses based on the top 30 families in the *fathom* dataset.

For the object detection model, I will implement a one-stage architecture based on a standard machine learning framework. A pretrained Convolutional network will be used as the backbone of this model. Supervised contrastive learning and improved data augmentation will be used to improve model performance.

Potential Extensions.
A natural extension is to increase the granularity of organism classification to family or genus level. If my proposed techniques have sufficiently improved classification performance over Gardar's implementation, then granular classification could be possible using one convolutional neural network (CNN) model. If this is not the case, an alternative approach is to train a hierarchical ensemble classifier that uses small, specialised classifiers for more granular classifications. Fine-Grained Image Analysis (FGIA) techniques could prove useful here.

Another extension is to include temporal information in the object detection model. This refers to utilising information from neighbouring video frames to improve object detection performance. Another potential extension is to train a Generative Adversarial Network (GAN) to generate training examples.

Success Criteria

Core:

- A software that automatically detects and classifies deep-sea organisms from input video footage, outputting bounding box annotations and associated classifications
- The object detection model should at least achieve comparable performance to Gardar's Faster-R-CNN model. Concretely, it should achieve a mAP > 0.55 (mean average precision) and mIoU > 0.23 (mean intersection over union) when evaluated on the *FathomNet* dataset
- The supervised contrastive approach should improve the performance of the image classification model relative to Gardar's Fast-R-CNN model on at least some of the classes

Extension:

- Implement an image classifier for a taxonomic class such as *Anthozoa* (corals) with at least order-level granularity
- Implement a hierarchical ensemble image classifier that iteratively applies more granular classifiers on an example
- Incorporate temporal information in object detection for video and improve mAP over frame-by-frame methods
- Implement a GAN-based training data generator

Starting Point

I have read the papers mentioned above. I never formally studied computer vision, but *Part IB Artificial Intelligence* gave me a foundational understanding of neural networks and *Part II Deep Neural Networks* will further my understanding of deep neural networks and model training.

I plan to use PyTorch to implement the machine learning models. I will implement each model from scratch except for the pretrained CNN backbone for the object detection model. I will only implement that from scratch if I can transfer the learned weights to it. Most of the code for this project will be written in Python, a language I am proficient in.

For a summer internship, I trained an object detection model for obstacles in construction sites. However, I did so through a high-level API with very little code. I have little prior experience with machine learning frameworks. I have completed several tutorials for TensorFlow with a Google Colab notebook.

Resources Required

Hardware. I plan to use my personal laptop (Windows 11 Home, AMD Ryzen 9 5900X, 16GB RAM, Nvidia RTX 3060 6GB, 1TB disk space) for dissertation writing and coding.

I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.

For model training, I will use the Cambridge HPC cluster. My access to the cluster is pending permission.

In case of hardware failure of my laptop, the MCS will be used as a backup. I will keep my data safe by synchronising daily with oneDrive and backing up weekly to an external hard drive. I will keep my codebase safe with frequent pushes to GitHub.

Datasets. The primary dataset for this project will be the *FathomNet* [2] dataset. *FathomNet* is licensed under a Creative Commons Attribution—No Derivatives 4.0 International License, allowing it to be used for non-commercial purposes with proper attribution. It can be accessed via a REST api[1] or through a python library[2].

Timetable and Milestones

After implementing each part of the project, the aim is to write a draft of the relevant section in the dissertation so that the writing is not left until the end of the project and details are not forgotten.

Michaelmas

- Week 2-3 *(15/10/22 - 23/10/22)*
  Setup development environment (libraries, Github repositories)
  Download the *FathomNet* dataset, begin data cleaning

- Week 4-5 *(24/10/22 - 06/11/22)*
  Research on object detection models
  Finish data cleaning
  Research and start developing image augmentation

- Week 6-7 *(07/11/22 - 20/11/22)*
  Start implementing object detection model without contrastive learning
  Continue to refine image augmentation
  Other Deadline: *(09/11/22)* Cloud Computing Assignment 1

- Week 8 *(21/11/22 - 27/11/22)*
  Research on contrastive learning
  Finish up object detection model without contrastive learning
  Finish up image augmentation
  Milestone: Image augmentation method

- Week 9 *(28/11/22 - 04/12/22)*
  Buffer period
  Other Deadline: *(28/11/22)* Cloud Computing Assignment 2

---

1 http://fathomnet.org:8080
2 https://pypi.org/project/fathomnet/

Winter Holiday

- Week 1 *(05/12/22 - 11/12/22)*
  Break for exam revision
  Research on Cambridge HPC cluster
  Try to train on the Cambridge HPC cluster

- Week 2-3 *(12/12/22 - 23/12/22)*
  Implement a training pipeline with contrastive learning
  Milestone: Complete all components needed to start training

- Week 3-4 *(24/12/22 - 01/01/23)*
  Holiday break
  Potentially research on extensions

- Week 5-6 *(02/01/23 - 15/01/23)*
  Start training on object detection model

Lent

- Week 1-2 *(16/01/23 - 29/01/23)*
  Continue training of object detection model
  Work on the progress report and oral presentation

- Week 3-4 *(30/01/23 - 12/02/23)*
  Finish up training of object detection model
  Evaluate object detection model, keep detailed records of test results
  Milestone: Object detection model training completed

  Milestone: *(03/02/23 12 noon)* Progress report deadline
  Milestone: *(08/02/23 - 15/02/23)* Progress Report presentation

- Week 5-6 *(13/02/23 - 26/02/23)*
  Study evaluation results
  Start writing the evaluation chapter of the dissertation
  Work on extensions if time allows

- Week 7-8 *(27/02/23 - 12/03/23)*
  Work on extensions if time allows
  Finish up the evaluation chapter of the dissertation

- Week 9 *(13/03/23 - 19/03/23)*
  Combine various components into a unified pipeline for video organism detection
  Tidy up the code base, improve documentation
  Milestone: Main software deliverable completed

Easter Holidays

- Week 1-2 *(20/03/23 - 02/04/23)*
  Combine sections of writing that have been completed so far in a coherent manner
  Work on the evaluation chapter of the dissertation
  Milestone: A complete dissertation with every component filled up

- Week 3 *(03/04/23 - 09/04/23)*
  Buffer period, may be used for extensions

- Week 4-5 *(10/04/23 - 23/04/23)*
  Complete the remaining parts of the dissertation
  Review the dissertation as a whole, rewrite segments if necessary
  Milestone: Submit dissertation draft 1 for DoS and supervisors to review

Easter

- Week 1 *(24/04/23 - 30/04/23)*
  Incorporate feedback from DoS supervisors
  Milestone: Submit dissertation draft 2 for DoS and supervisors to review

- Week 2 *(01/05/23 - 07/05/23)*
  Incorporate feedback from DoS supervisors, proofread dissertation and verify citations

- Week 3 *(08/05/23 - 12/05/23)*
  Final modifications to the dissertation
  Milestone: Submit dissertation!

Bibliography

[1] Hedley, John D., et al. "Remote sensing of coral reefs for monitoring and management: a review." *Remote Sensing* 8.2 (2016): 118.

[2] Boulais, Océane, et al. "FathomNet: An underwater image training database for ocean exploration and discovery." *arXiv preprint arXiv:2007.00114* (2020).

[3] Marcos, Ma Shiela Angeli C., Maricor N. Soriano, and Caesar A. Saloma. "Classification of coral reef images from underwater video using neural networks." *Optics express* 13.22 (2005): 8766-8771.

[4] Pizarro, Oscar, et al. "Towards image-based marine habitat classification." *OCEANS 2008*. IEEE, 2008.
e
[5] Gómez-Ríos, Anabel, et al. "Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation." *Expert Systems with Applications* 118 (2019): 315-328.
[6] Rodrigues, Francisco Caio Maia et al. "Evaluation of Transfer Learning Scenarios in Plankton Image Classification." *VISIGRAPP* (2018).

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* 60.6 (2017): 84-90.

[8] Ingvarsson, Gardar. “Deep-sea organism detection”. *Part II Project Dissertation* (2022).

[9] Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[10] Girshick, Ross. "Fast R-CNN." *Proceedings of the IEEE international conference on computer vision*. 2015.

[11] Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and efficient object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

[12] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[13] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *arXiv preprint arXiv:2207.02696* (2022).

[14] Zhong, Zhun, et al. "Random erasing data augmentation." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. No. 07. 2020.

[15] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. ""GrabCut" interactive foreground extraction using iterated graph cuts." *ACM transactions on graphics (TOG)* 23.3 (2004): 309-314.

[16] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.