Deep Learning for Detecting Amphoras in Ancient Shipwrecks

by Tianyao Chen

Bachelor Thesis in Computer Science

Jacobs University Bremen | Supervisor: Prof. Dr. Andreas Birk

Table of Contents

Introduction - Motivation

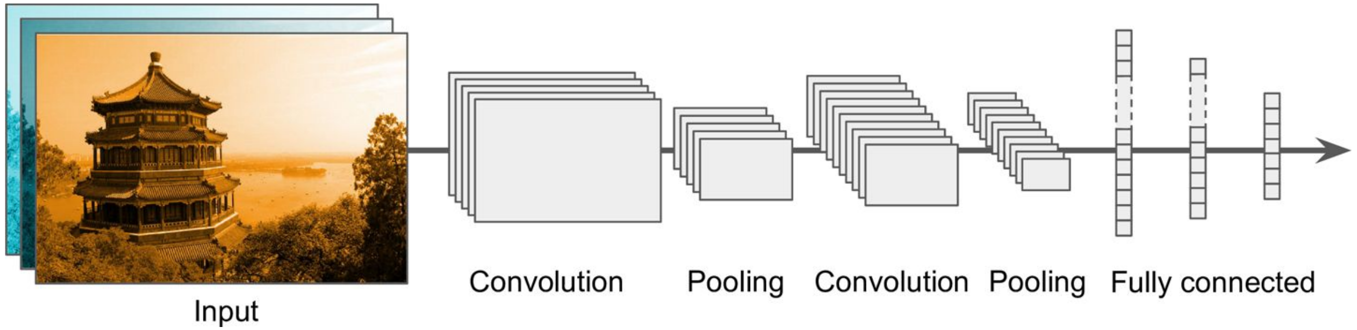
- The name amphora is derived from the Greek word amphoreus, which means ”two-handled”.
- Amphoras (or amphorae) were commercially used to ship products throughout the Mediterranean by the ancient Greek and Roman empires.
- They have great significance in archaeology. Archaeologists have been using them to recreate the trade patterns and date ruins.

Introduction - Motivation

- Computer vision has lower costs compared to sonar imagery and laser scanning. Plus, the increasingly abundant visual data obtained through autonomous underwater vehicles (AUVs), unmanned underwater vehicles (UUVs), and seafloor cabled observatories motivate us to utilize deep learning.
- The study of deep-water shipwrecks is in high demand, as the threats to these sites are increasing. Many shipwrecks are likely to be damaged before they can be studied.

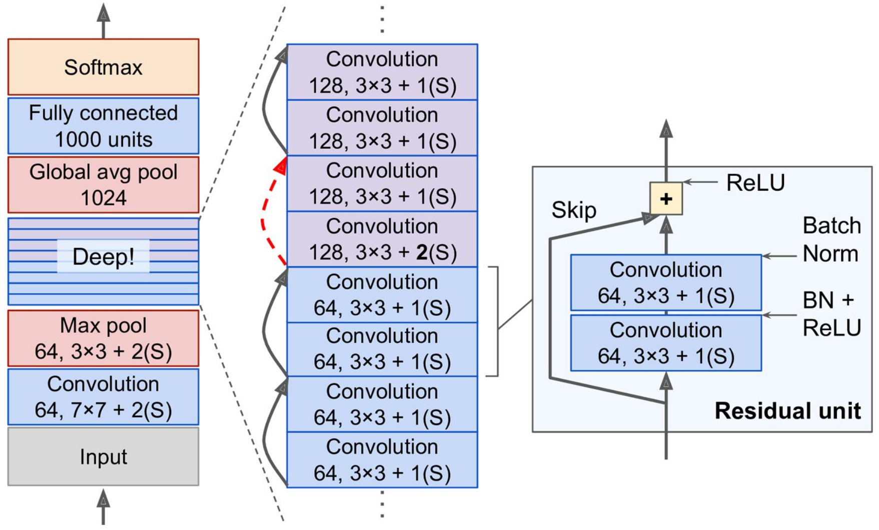
Introduction - Convolutional Neural Networks (CNNs)

- Convolutional Neural Networks (CNNs) were inspired by the brain's visual cortex, and they have been used in computer vision since the 1980s.
- The filters or convolution kernels are learned during training.
- The pooling layers subsample (i.e. shrink) the input image to reduce the computational load.



Introduction - Convolutional Neural Networks (CNNs)

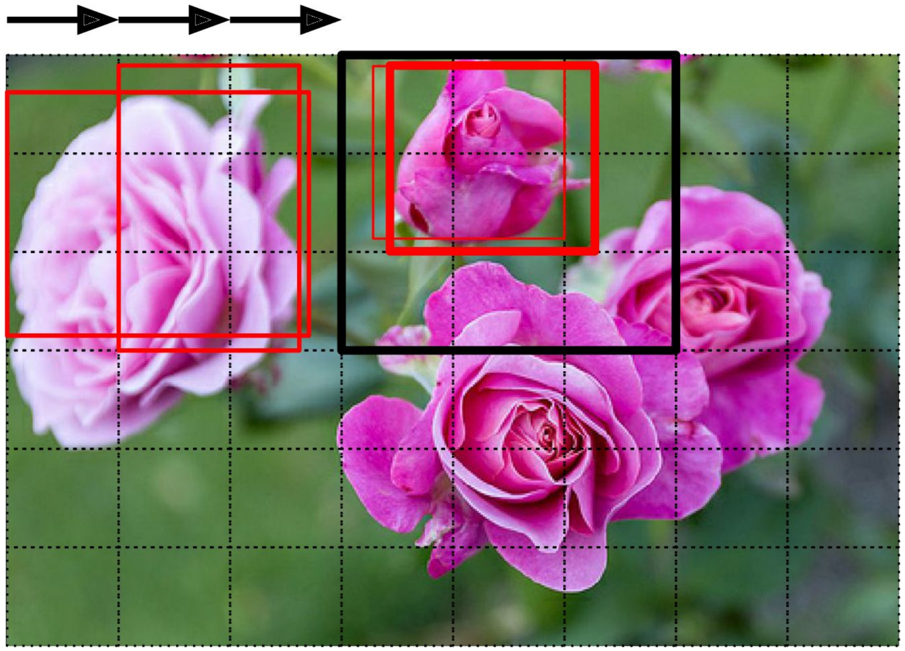
- One CNN architecture is called ResNet (Residual Network), which is the backbone network used in this study. ResNet introduced skip connections (or shortcut connections), which help speed up the training.



Softmax
Fully connected 1000 units
Global avg pool 1024
Deep!
Max pool 64, 3×3 + 2(S)
Convolution 64, 7×7 + 2(S)
Input
Convolution 128, 3×3 + 1(S)
Convolution 128, 3×3 + 1(S)
Convolution 128, 3×3 + 1(S)
Convolution 128, 3×3 + 2(S)
Convolution 64, 3×3 + 1(S)
Convolution 64, 3×3 + 1(S)
Convolution 64, 3×3 + 1(S)
Convolution 64, 3×3 + 1(S)
Skip
+
ReLU
Batch Norm
BN + ReLU
Convolution 64, 3×3 + 1(S)
Convolution 64, 3×3 + 1(S)
Residual unit

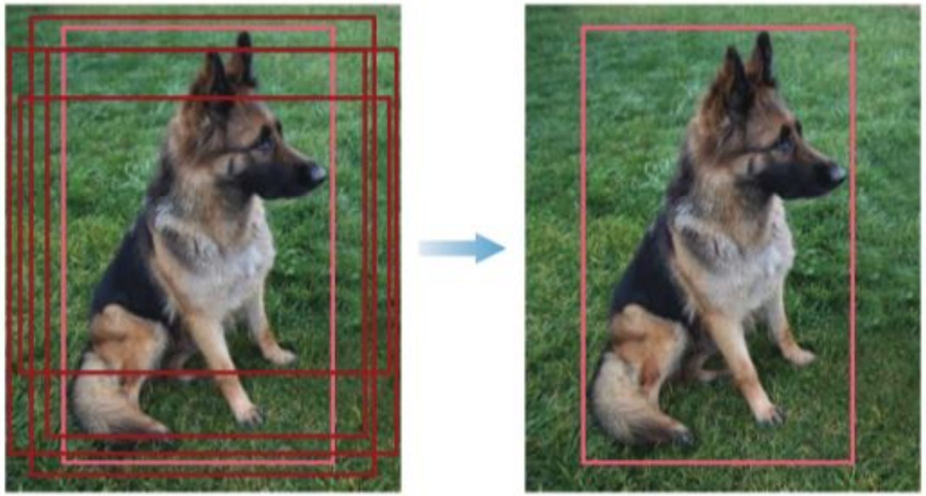
Introduction - Object Detection

- Object detection is the computer vision task that localizes and classifies objects in an image.
- The traditional sliding window approach is to slide a trained CNN multiple times with various window sizes to detect objects at different scales, which is quite slow.



Introduction - Object Detection

- There are four high-level components for modern general object detection frameworks.
- **Region Proposal.**
- **Backbone Network.**
- **Non-Maximum Suppression (NMS).**


Predictions before NMS

After applying non-maximum suppression

Introduction - Object Detection

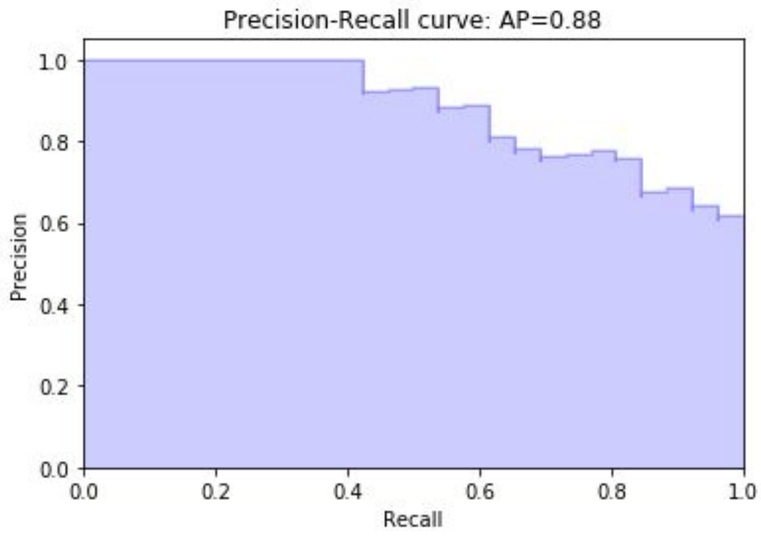
- **Metric** The main metric is mean average precision (mAP). To understand mAP, we need to understand first intersection over union (IoU) and the precision-recall curve (PR curve).

$$IoU = \frac{B_{ground\ truth} \cap B_{prediction}}{B_{ground\ truth} \cup B_{prediction}}$$

$$Precision = \frac{TP}{TP + FP}$$

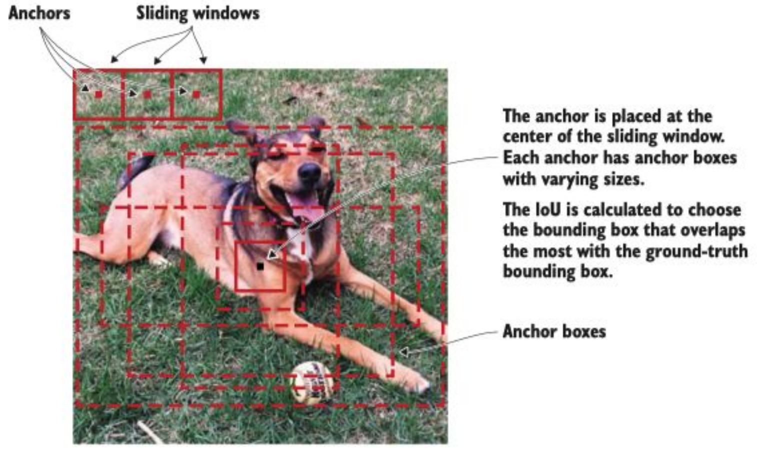
$$Recall = \frac{TP}{TP + FN}$$

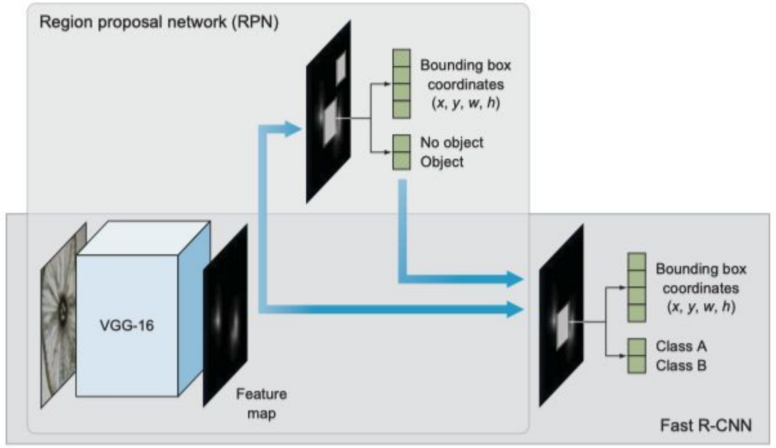
We can obtain the average precision (AP) by drawing the PR curve and computing the area under the curve (AUC). Then we get the mAP by averaging the AP over all the classes. The traditional Pascal VOC metric uses mAP@0.5, which means the IoU threshold in NMS is 0.5. The new COCO metric mAP_{coco} = mAP@[0.50 : 0.05 : 0.95] is averaged over different IOU thresholds from 0.5 to 0.95 in steps of 0.05, which rewards detectors with better localization.



Introduction - Object Detection

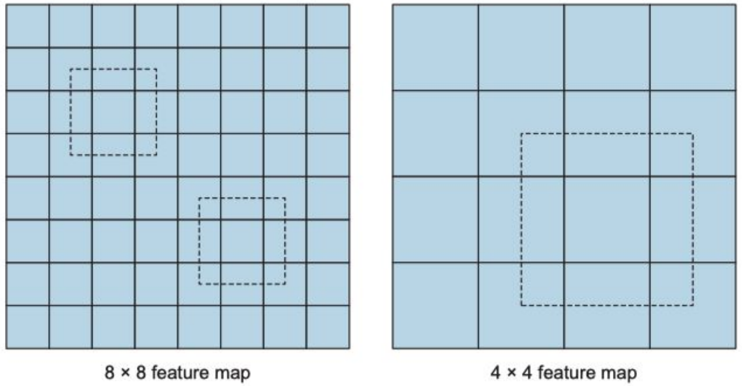
- The Region-Based Convolutional Neural Networks (R-CNN) family has the original R-CNN, Fast R-CNN, and then Faster R-CNN.
- They are two-stage detectors that separate region proposal and detection. They can not achieve real-time detection, and they are very computationally intensive.

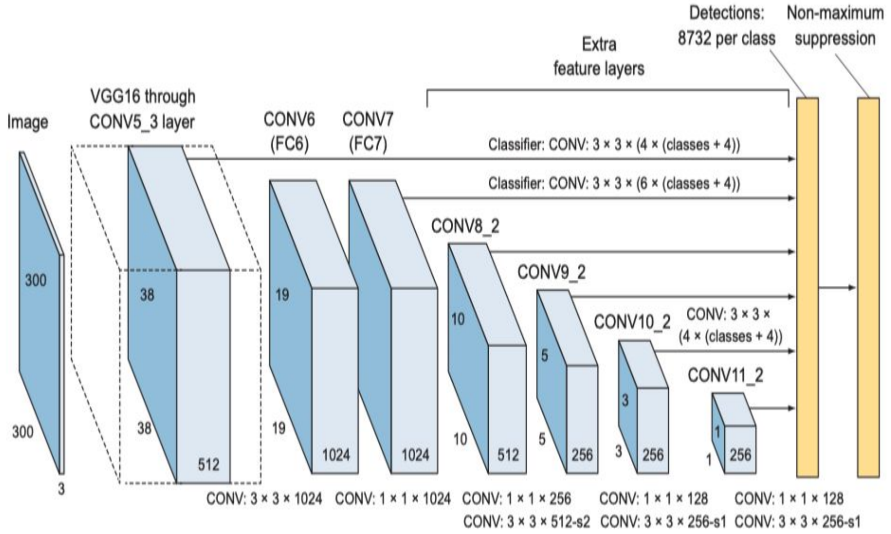

Anchors
Sliding windows
The anchor is placed at the center of the sliding window. Each anchor has anchor boxes with varying sizes.
The IoU is calculated to choose the bounding box that overlaps the most with the ground-truth bounding box.
Anchor boxes


Region proposal network (RPN)
Bounding box coordinates (x, y, w, h)
No object
Object
VGG-16
Feature map
Bounding box coordinates (x, y, w, h)
Class A
Class B
Fast R-CNN

Introduction - Object Detection

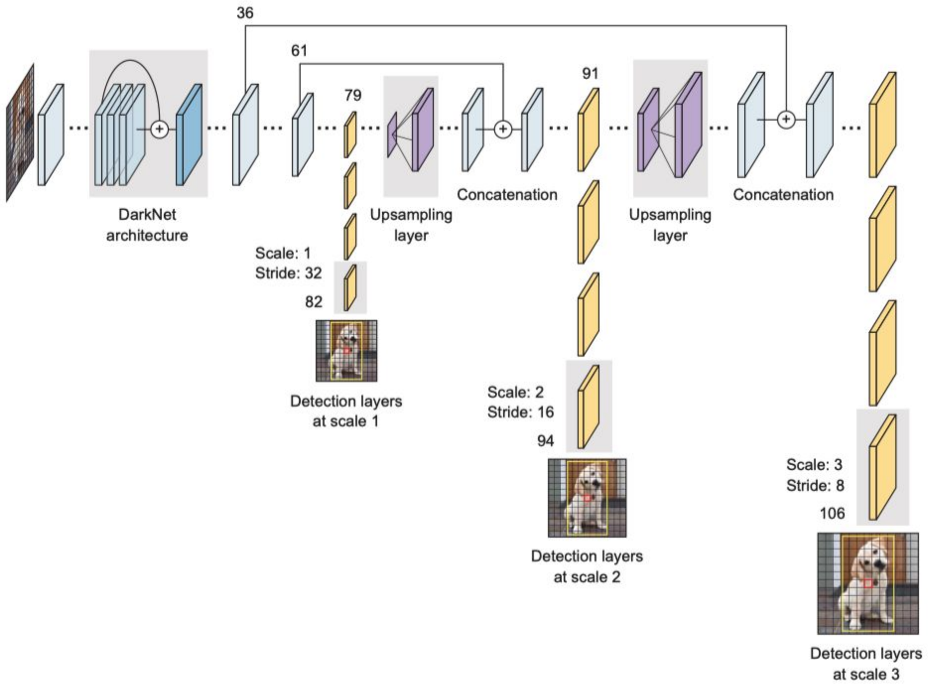
- Single Shot Detector (SSD) is a one-stage detector that makes both the objectness and class probability predictions directly in one shot.
- Multi-scale feature layers are convolutional layers that decrease in size progressively to detect objects at multiple scales.





Introduction - Object Detection

- You Only Look Once (YOLO) is a one-stage real-time detector family similar to SSD.
- YOLO divides the image into a grid, and a grid cell is responsible for detecting an object if the center of the object is inside the cell.
- YOLOv4, a bleeding-edge detector introduced in 2020, utilizes numerous new features to improve the performance from YOLOv3.
- YOLOv5 is under active development and the authors have yet to publish a paper.



Related Work

- For fish detection, Qin et al., Zhang et al., Villon et al., and Xu et al. respectively used Fast R-CNN, a model similar to R-CNN, sliding window with a CNN, and YOLOv3.
- For crab detection, Cao et al. proposed a detector based on SSD .
- For amphora detection, Pasquet et al. used sliding window with a CNN on a high-resolution orthophoto (i.e. aerial photo). This is the only study we found on amphora detection with deep learning.

Data and Methods - Data

- The dataset consists of 50 images (294 objects) in the training set and 7 images (31 objects) in the validation set, which maintains a 90%: 10% ratio for the object count. Two additional images were used as the test set. All the images were obtained through various sources and were labeled with labelImg.

Data and Methods - Model

- Out of the three families of object detection frameworks we have discussed, SSD was chosen due to the following reasons:
    - Faster R-CNN is too computationally intensive and too slow for AUVs and UUVs.
    - YOLOv5, the state-of-art model of the YOLO family, is under active development and the authors have yet to publish a paper.

Data and Methods - Model Training

- Transfer learning was used as the model was pre-trained on the COCO dataset.
- Data augmentation was also used to artificially enrich the dataset.
- The model was trained on Google Colaboratory with a GPU runtime, and the training took 2 hours and 40 minutes to reach the peak mAP result.

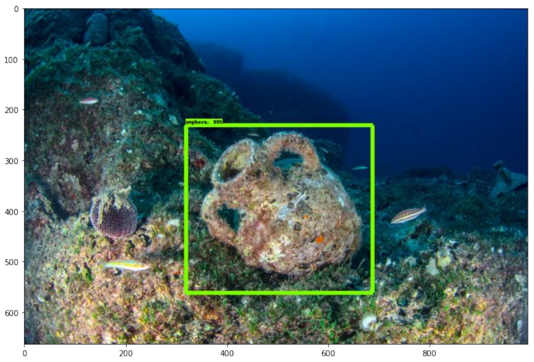
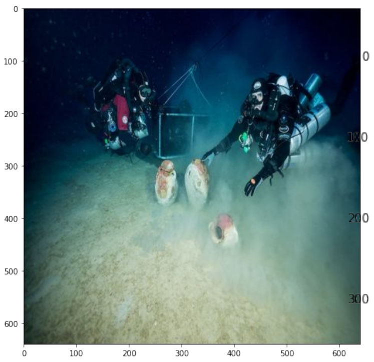
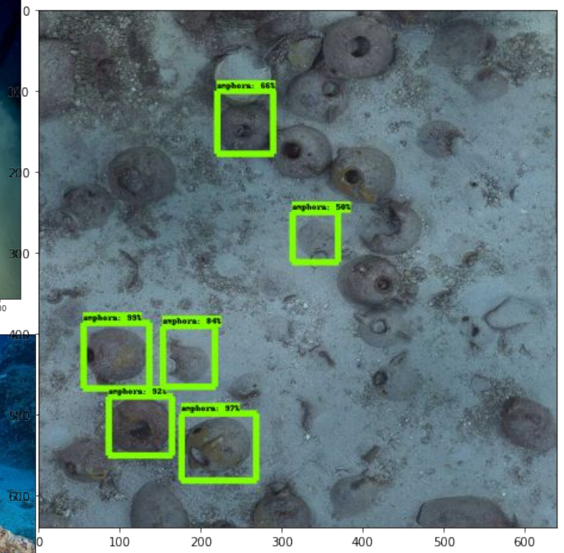
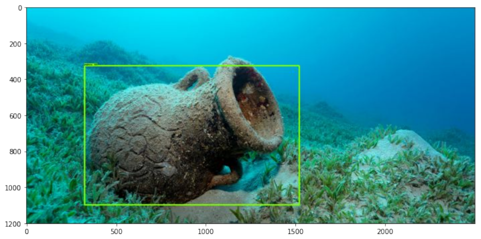
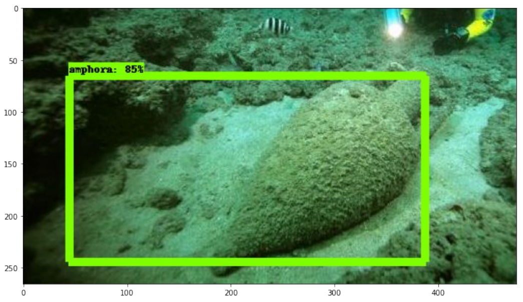
Evaluation

- The main metric mAP_{coco} from our model is 0.238, while the traditional mAP@0.5 is 0.503.
- The documented mAP_{coco} in the TensorFlow 1 Detection Model Zoo of our selected model is 0.35. Our model's performance is indeed lower than that in the model zoo.
- Pasquet et al. did not compute the mAP and only mentioned that they detected around 90.3 percent of amphoras. However, we can conclude that our model did not detect more than 90.3 percent based on the visual inspection.

Evaluation

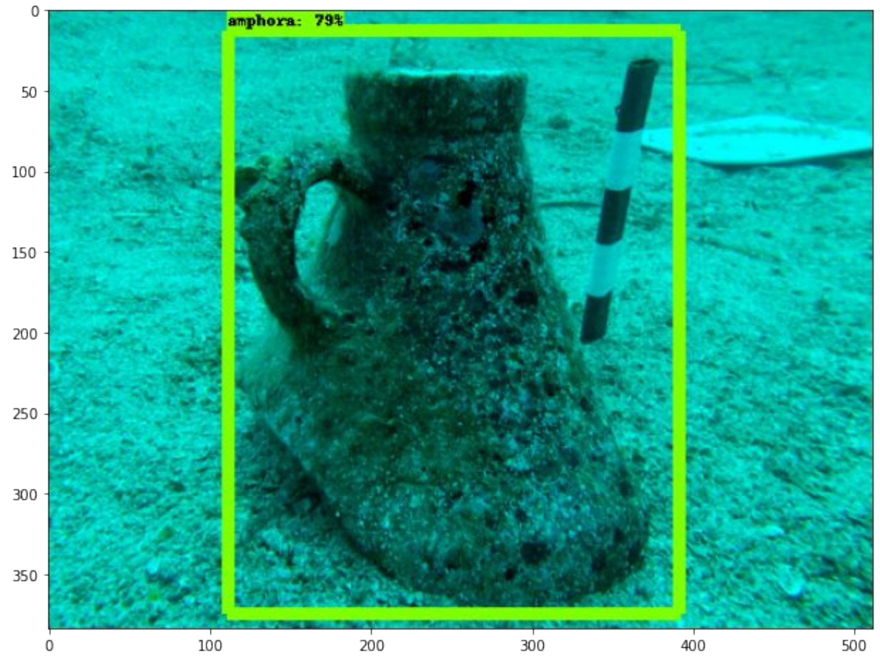
Nevertheless, our model's performance is still impressive due to the following reasons:

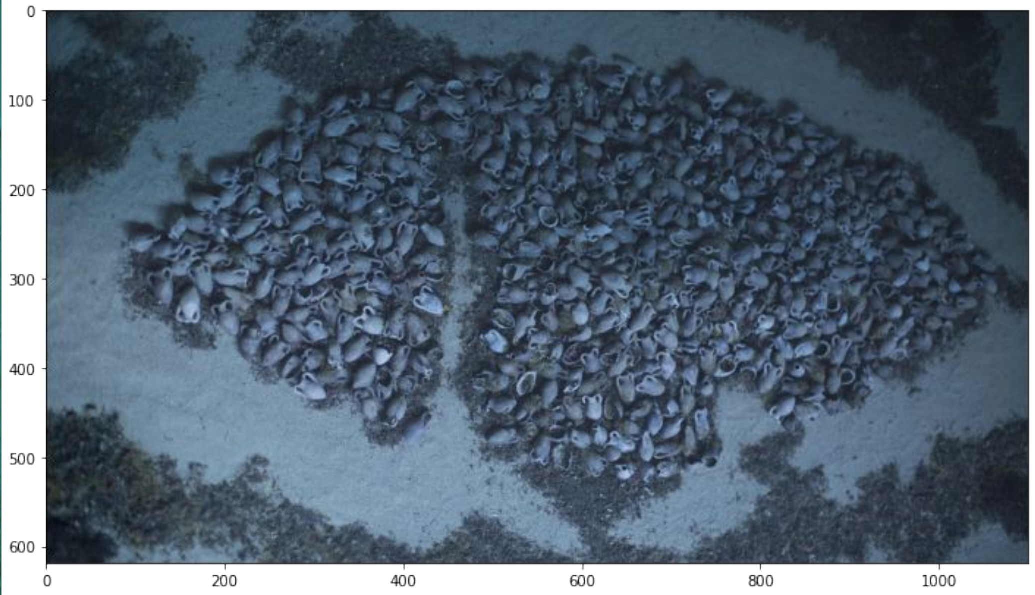
- Our dataset is very small. The COCO dataset has 328 thousand images. For Pasquet et al., they split one orthophoto into around 890 images for training.
- Our dataset is more diverse than that of Pasquet et al., since our images are from numerous sources instead of from a single orthophoto.
- Underwater object detection is more challenging than general object detection. In fact, the undetected amphoras in the validation set are almost all either broken, or partially buried in sand, or blocked by suspended particles.
- Our mAP@0.5 score is comparable with that from Xu et al. for fish detection. They achieved a 0.5392 mAP@0.5 using YOLOv3, although they did not compute the mAP_{coco}.


amphora: 99%


amphora: 66%
amphora: 50%
amphora: 99%
amphora: 84%
amphora: 92%
amphora: 97%


amphora: 85%

Evaluation

- We also ran the model on 2 test images. To our surprise, the model managed to detect the amphora in the first image even though it is so broken that only half of it is present.
- As for the second image, none of the amphoras was detected. The size of the image is only 709 x 411, while it has hundreds of amphora instances.
    - This means that they are mostly small instances, which our training set does not include.
    - The image can be described as a crowded and densely packed scene, which is also notoriously challenging for object detectors.
    - We tried to split the image and add parts of it to the training set. However, the inter-occlusion and the low resolution caused the training to diverge.


amphora: 79%
0
50
100
150
200
250
300
350
0
100
200
300
400
500


0
100
200
300
400
500
600
0
200
400
600
800
1000

Conclusion and Future Work

- Detecting underwater amphoras is a challenging task, as the object detectors have to overcome the same difficulties posed by general underwater object detection, the considerable variations of the amphoras' shapes, and the crowdedness of the scenes in some shipwrecks.
- For future studies, there are many aspects to improve upon:
  - The small number and the low resolution of the images in our dataset are the bottlenecks.
  - Many bleeding-edge object detectors with better performance are emerging. It will be worth experimenting with YOLOv5, EfficientDet, CenterNet, etc.
  - Following the approach from Pasquet et al., we could try to define 2 separate classes for the head and the body to better detect broken instances.

Key References

- Diana Twede. “Commercial amphoras: the earliest consumer packages?” In: Journal of Macromarketing 22.1 (2002), pp. 98–108.
- Mohamed Elgendy. Deep Learning for Vision Systems. Manning Publications, 2020.
- Aurelien Geron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O’Reilly Media, 2019.
- Kaiming He et al. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.
- Jerome Pasquet et al. “Amphora detection based on a gradient weighted error in a convolution neuronal network”. In: (2017).
- Wei Liu et al. “Ssd: Single shot multibox detector”. In: European conference on computer vision. Springer. 2016, pp. 21–37.
- Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: arXiv preprint arXiv:1506.01497 (2015).
- Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: arXiv preprint arXiv:1804.02767 (2018).

Thank you!