Marine Creatures

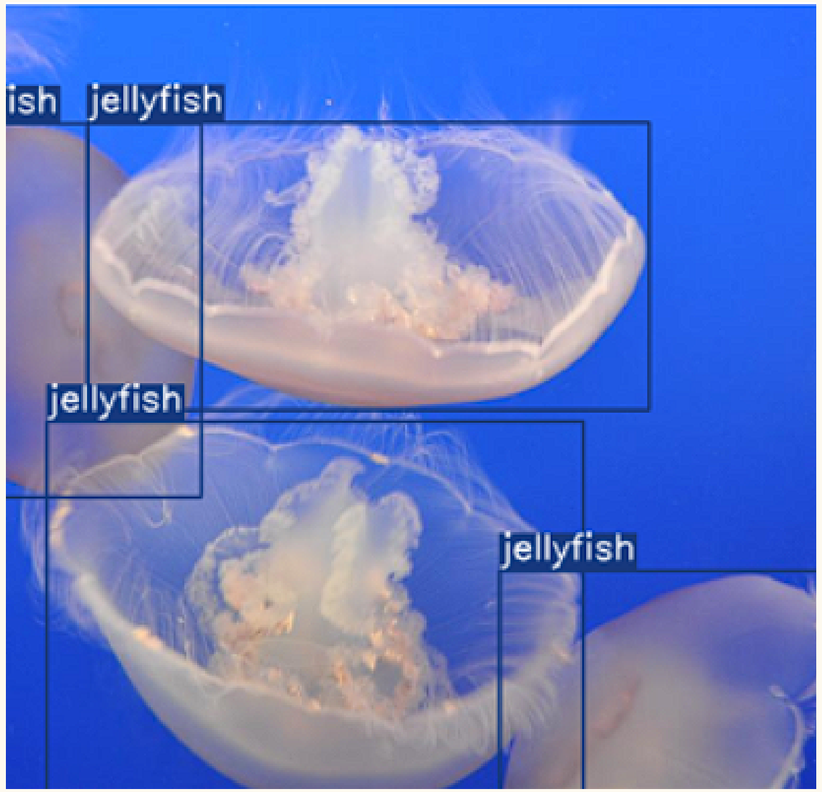
Project

Computer Vision:
Object detection
Dataset: Aquarium
Model: YOLOv5 (by Ultralytics)

Detection

Prepared by:
Zhanetta Koba

Computer Vision: Object Detection



- a localization task - outputs the bounding box (x, y coordinate). In essence, the localization task is a regression problem that outputs continuous numbers representing the bounding box coordinates

- a classification task - classifies an object (person vs car etc.)

As of today, COCO mAP is the most popular metric for evaluating object detection models.

A high mAP -> a model has both a low FN and a low FP rate.

Dataset: Aquarium

… is collected from the sources:

1.Aquarium Dataset https://public.roboflow.com/object-detection/aquarium
This dataset consists of images collected from two aquariums in the US:

- The Henry Doorly Zoo in Omaha (October 16, 2020);
- National Aquarium in Baltimore (November 14, 2020).

2. Flickr https://www.flickr.com

… encompasses 7 classes:

- fish
- jellyfish
- penguin
- puffin
- shark
- starfish
- stingray

… consists of:

1054 images (each image is fit within 1024x1024)

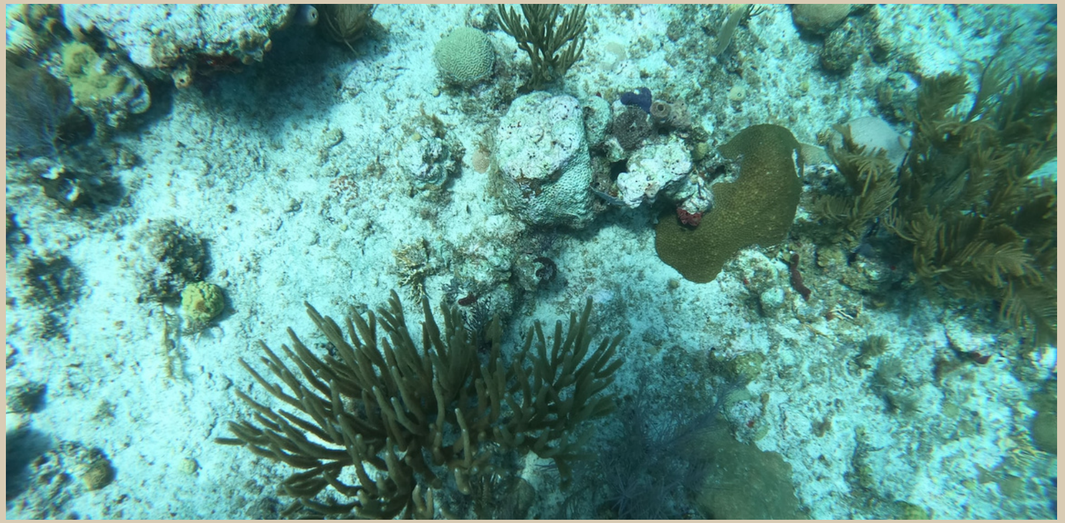
6, 284 annotations (most images contain multiple bboxes)

… is split into:

training (70%)
validation (20%)
test (10%)

… could be used for:

- coral reef conservation
- environmental health monitoring
- swimmer safety
- pet analytics
- automated feeding



Background images

Background images are images with no objects that are added to a dataset to reduce False Positives (FP).

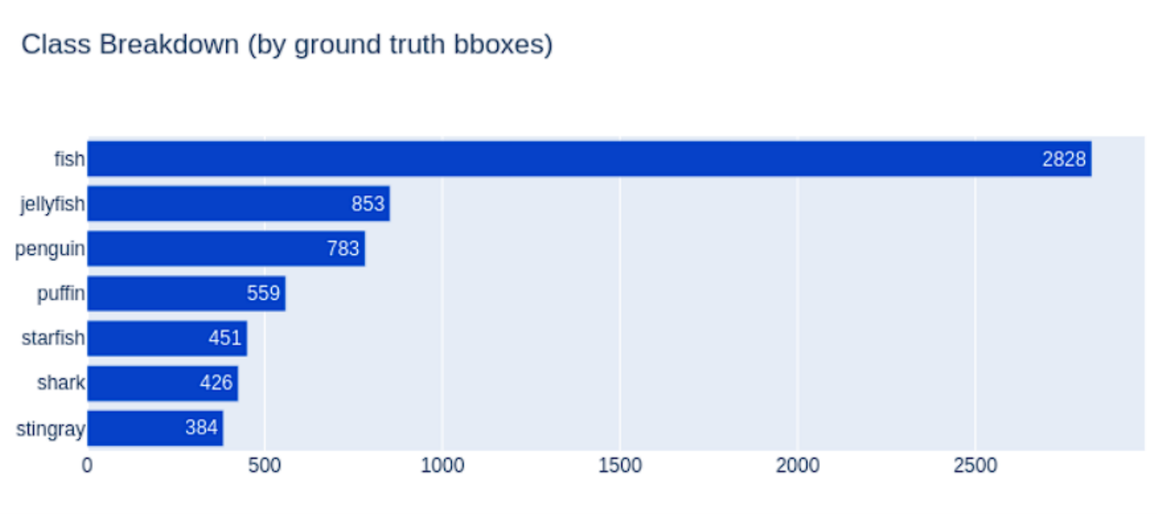
It is recommended to include about 0-10% background images in a dataset to help reduce FPs (e.g. COCO dataset has 1000 background images for reference, 1% of the total). No labels are required for them.

12 background images with no objects in the Aquarium dataset (which is 1% of the total).

Roboflow platform pros:

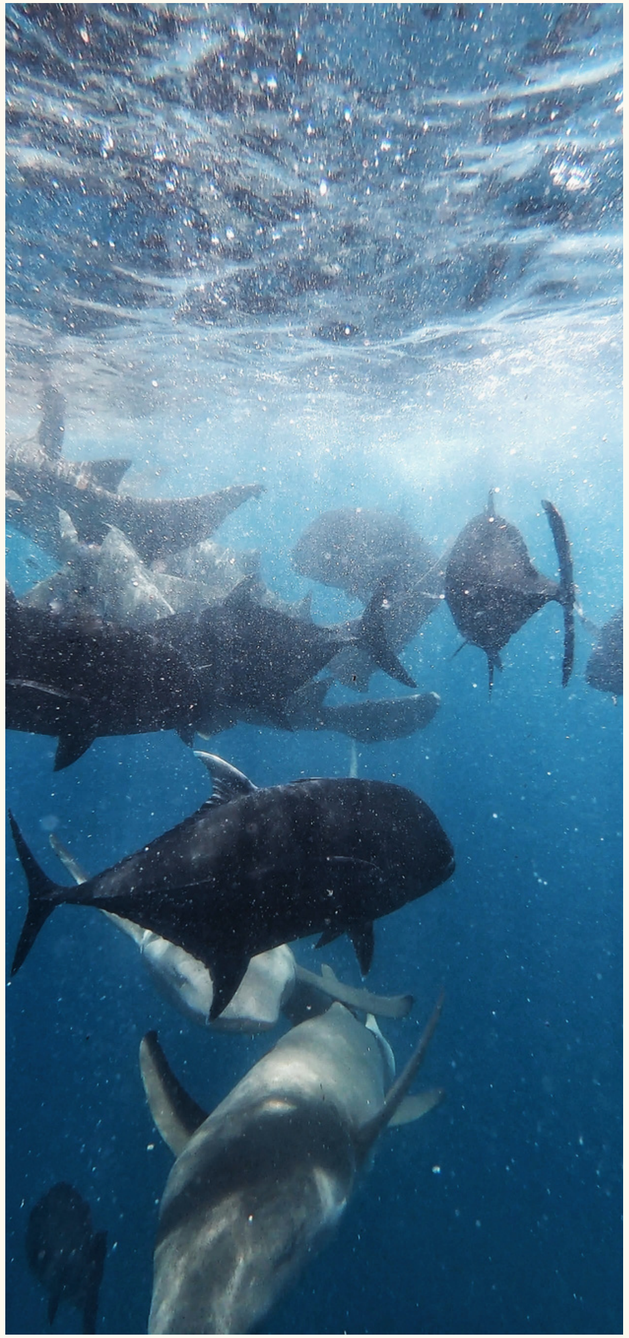
- allows to create a project and store collected images in one place at hand;
- images can be further annotated with bounding boxes;
- while generating dataset, additional configurations are available such as train/test split, preprocessing, and augmentation;
- dataset health check page provides some analytics;

… and there's a version control (for datasets).



Difficulties in data collection and preparation

- class imbalance;
- annotation errors in the Aquarium dataset (from Roboflow), e.g. not all objects labelled in the images, some annotations misclassify species: shark vs. fish, stingray vs. jellyfish).



YOLOv5

In YOLO family, inference speed and small model size are of utmost importance.

The 'You only look once' model is a single-stage object detector (the task of localization and classification is done in one pass)

YOLOv5 uses a Convolutional Neural Network (CNN) backbone to form image features. These features are combined in the model neck and sent to the head. The model head then interprets the combined features to predict the class of an image.

With each training batch, YOLOv5 passes training data through a data loader, which augments data online.

Data Augmentation

YOLOv5 applies online imagespace and colorspace augmentations in the trainloader. Images are never presented twice in the same way.

Techniques applied:

- HSV (Hue-Saturation-Value)
- Image scale
- Image flip (left-right)
- Image Mosaic

Additionally, Albumentations techniques:

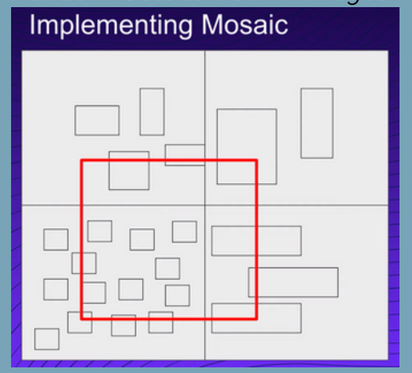
- Blur
- MedianBlur
- To Gray
- CLAHE (Apply Contrast Limited Adaptive Histogram Equalization)

Mosaic Augmentation

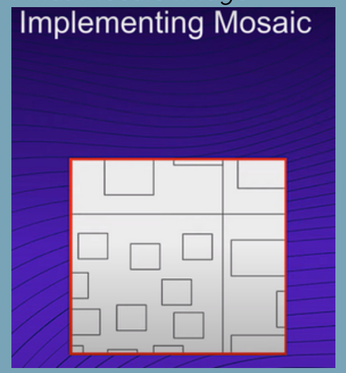
Mosaic augmentation helps the model learn to address the well known "small object problem" - where small objects are not as accurately detected as larger objects.

- simulates four random crops (which can help the model perform better in cases of occlusion);
- combines classes that may not be seen together in the training set;
- varies the number of objects in the images;
- images are never presented twice in the same way.

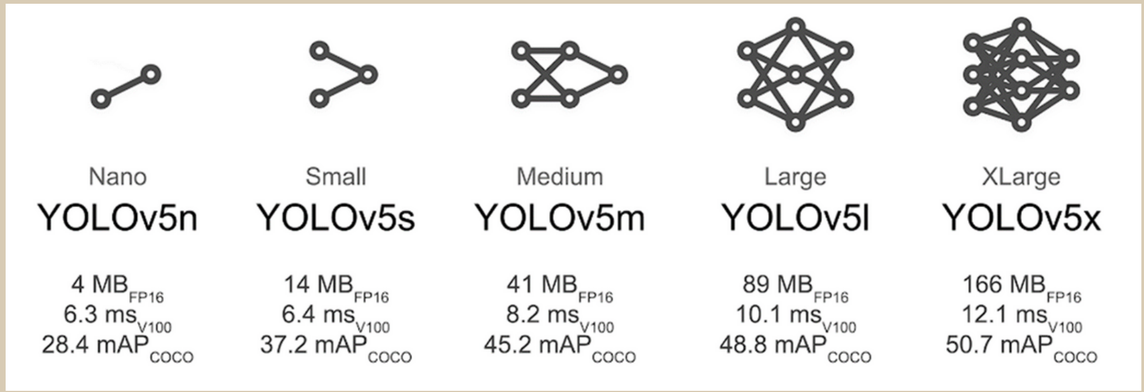
Random Cutout from 4 Images



Final Mosaic Image



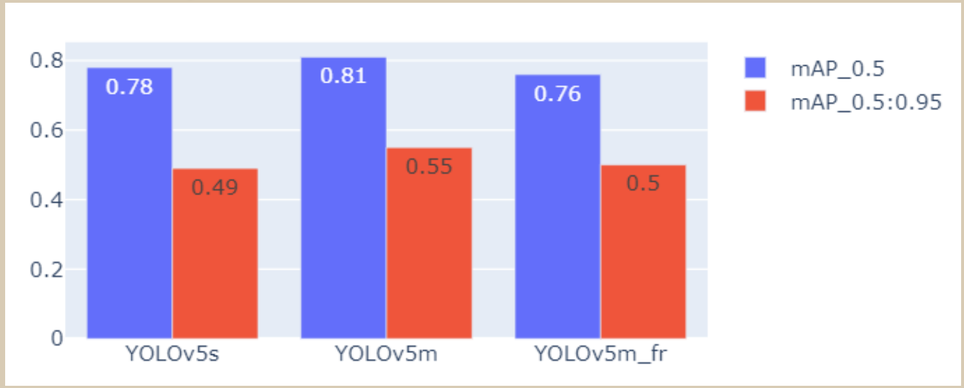
YOLOv5



Models YOLOv5s and YOLOv5m are selected for transfer learning in the project.

Baseline: YOLOv5s with default settings (by Ultralytics).

mAP: training results



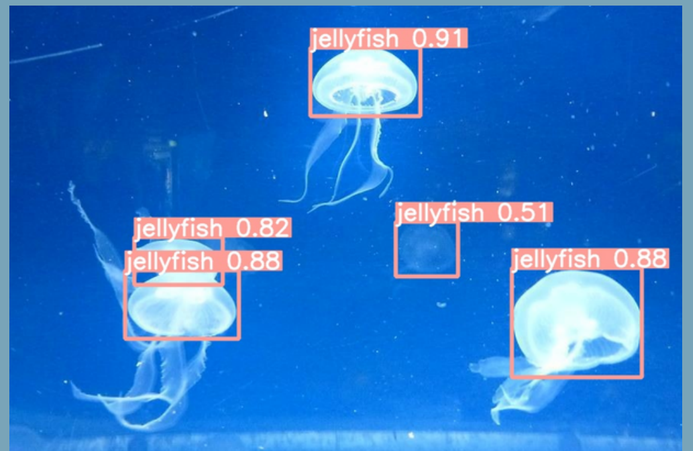
https://github.com/ultralytics/yolov5

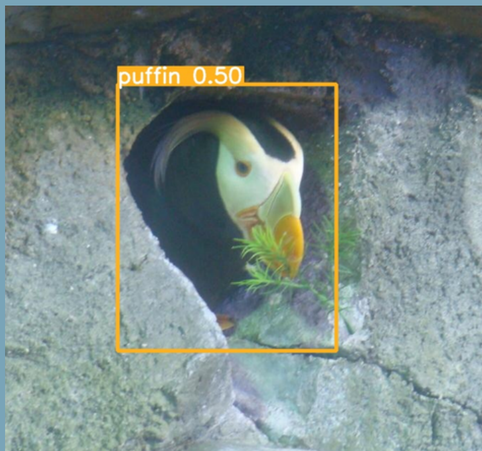
Project Folder Structure

```
├──── docs
├──── input
│   ├──── data
│         ├──── test
│         ├──── train
│         ├──── valid
│          ──── data.yaml
│   └──── inference
├──── models
│   ├──── best_models_storage
│   ├──── runs
│        ├──── train
│        └──── detect
├──── notebooks
│   └──── imgs
├──── src
├──── yolov5
 ──── app_utils.py
 ──── main.py
 ──── README.md
 ──── requirements.txt
```

*yolov5 folder is the external one from github with its own structure.

Predicted examples:





CLI

(examples)

src/dataset.py

python src/dataset.py python dataset.py --api_key jxxxxxxxxxxxx
--workspace aquariumdataset --project_name aquarium-dataset-78xxx
--version 10

src/train_infer.py

cd src
python train_infer.py --path_yaml ../input/data/

python src/train_infer.py --dir_move_from results_yolo5s

yolov5/train.py

python yolov5/train.py --data input/data/data.yaml --weights
yolov5/yolov5m.pt --img 640 --epochs 40 --batch-size 16
--name results_yolo5m --cache --project models/runs/train

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in input/data/ to yolov5pytorch: 100% [110720546 /
Extracting Dataset Version Zip to input/data/ in yolov5pytorch:: 100%|

Dataset downloaded.

The dataset consists of 1054 images.
-------------------
Train: 743 images.
Validation: 207 images.
Test: 104 images.

The dataset encompasses 7 classes:
['fish', 'jellyfish', 'penguin', 'puffin', 'shark', 'starfish', 'stingray']

There are 12 background images with no objects.

Number of annotations: 6284

Training Set Stats
-------------------
Class breakdown (by ground truth bboxes):
fish: 1970
jellyfish: 594
penguin: 593
puffin: 402
shark: 305
starfish: 268
stingray: 269

Validation Set Stats
-------------------
Class breakdown (by ground truth bboxes):
fish: 536
jellyfish: 176
penguin: 90
puffin: 106
shark: 88
starfish: 147
stingray: 72
(venv) ubuntu@ubuntu:~/Documents/project_yolov5/src$
```

```
(venv) ubuntu@ubuntu:~/Documents/project_yolov5$ python src/train_infer.py

The paths to image folders adjusted in the .yaml file:
Train: ../input/data/train/images
Val: ../input/data/valid/images
Test: ../input/data/test/images

Outputs are saved to directory: results_yolo_5s ...
Outputs will be saved to directory: results_yolo_5s
(venv) ubuntu@ubuntu:~/Documents/project_yolov5$
```

Thank You!