PEEK A BOO

TOWARDS SMART SEARCH AND RESCUE

Presented by:

Mohamed Alshafai - 87311

Rana Elfakharany - 87725

Hiba Saleem - 87239

OVERVIEW

- Introduction
- Problem
- Literary Review
- Methodology
- Implementation
- Results
- Evaluation
- Discussion
- Work Division
- Demonstration

INTRODUCTION

- Unmanned Aerial Vehicles (UAVs) are used in SAR missions due to their speed, versatility, and ability to provide an overview of the scene

- During SAR missions, wide areas of water need to be quickly assessed using UAVs

- Focuses on detection, localization, and tracking people in open water

INTRODUCTION

Challenge 1

Lack of large-scale marine related datasets that do not focus on ships

Challenge 2

Marine datasets are often satellite images, which contain inference (ex: clouds)

Challenge 3

Satellite images only provide a top-down view of the scene, which is inefficient for SAR

Challenge 4

Current approaches make use of classical ML algorithms, which cannot handle a lot of factors

PROBLEM STATEMENT

Problem: Satellite imagery and marine datasets used by SAR missions to identify people in marine environments are limited and inefficient.

First Problem

The inability to accurately identify missing humans result in poor surveillance and security in maritime operations.

Second Problem

The low visibility and environmental variations in satellite imagery affects detection results.

LITERARY REVIEW

Model	Dataset	Strengths	Weaknesses
Faster R-CNN [1]	SeaDronesSee	AP of 30.4 and mAP@0.5 of 54.66	Confuses life jackets for swimmers
YOLOv4 [2]	Custom dataset captured from satellites & drones	F1-Score of 0.89	Low resolution of images; no variety of classes/labels
Efficientdet [3]	Custom dataset	mAP@0.75 of 74%	Model is slow with inference time of 48.72 ms
DoubleFPN [4]	M^2SODAI	mAP of 42.2	Performance can be improved by using pretrained backbones
ABT-YOLOv7 [5]	SeaDronesSee, MOBDrone	Precision of 92.3%, recall of 91.7%, mAP of 91.6%	Datasets do not include comprehensive data

METHODOLOGY

Data Collection

The SeaDronesSee dataset is selected for its high number of samples, variety of objects, high complexity, and assurance of model robustness

Model Selection

The models selected include variants of YOLO such as YOLOv6N6, YOLOv6S6, YOLOv6M6, and YOLOv6L6.

Model Training

Preprocessing of the data as well as configuration of the model architecture is performed before training and monitoring the metrics.

METHODOLOGY

Model Evaluation

The models are evaluated using the validation set to assess their performance in terms of precision, recall, F1-score, and mAP.

Documentation

The entire implementation process and results are documented and shared, including recommendations for future work.

Comparison

The results of the model are compared to the state-of-the-art models on the dataset.

IMPLEMENTATION

ARCHITECTURE

- **YOLOv6-S6**
  - “S”: small compact version for real-time processing
  - Strikes a balance between speed and accuracy
  - Limited compared to high-end systems but requires more power than the nano version

- **YOLOv6-N6**
  - “N”: nano lightweight version designed for resource-constrained environments
  - Focuses on efficiency
  - Suitable for edge devices

IMPLEMENTATION

ARCHITECTURE

- **YOLOv6-M6**
  - “M”: medium model with a balance between size and accuracy
  - Offers a reasonable compromise between accuracy and speed
  - Suitable for applications where a balance between both is needed
- **YOLOv6-L6**
  - “L”: large and complex model with high accuracy but increased computational demands
  - Use for applications that require high accuracy
  - Not suitable for edge devices with limited resources

IMPLEMENTATION

TRAINING PROCESS

1. **Install YOLOv6 and its required files:**

```
git clone https://github.com/meituan/YOLOv6
cd YOLOv6
pip install -r requirements.txt
```

2. **Download the dataset and place the .yaml file inside the ‘data’ folder**

Note: The dataset format must be .yaml, which is needed by YOLO.

3. **Adjust the data path and other parameters in the ‘train.py’ file**

Things to change include:

- Data path
- Image size
- Batch size
- Number of epochs

IMPLEMENTATION

TRAINING PROCESS

4. Choose the model architecture that you would like to run and download the .pt file into the ‘weights’ folder

   Note: If you are running the model from the start without any pre-trained weights, this step can be skipped

5. If you are using .py instead of .ipynb, make sure to save your results into files to be able to view them later

6. Start training by running the following:

   python tools/train.py --batch 32 --conf configs/yolov6m6_finetune.py --data /localHome/cloudies/PeekABoo/SeaDronesSee-Yolov8/data.yaml --img 1280 --device 3 --epochs 50

IMPLEMENTATION

PERFORMANCE METRICS

$$mAP = \frac{1}{k} \sum_{i}^{k} AP_i$$

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

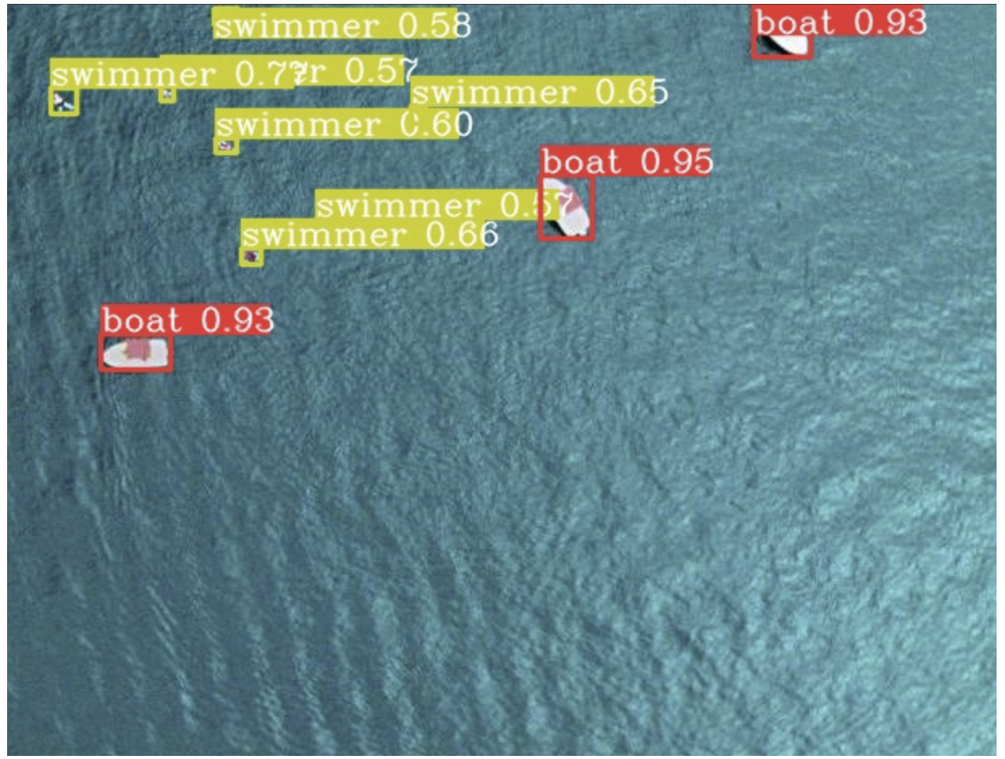
$$Precision = \frac{TP}{TP + FP}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

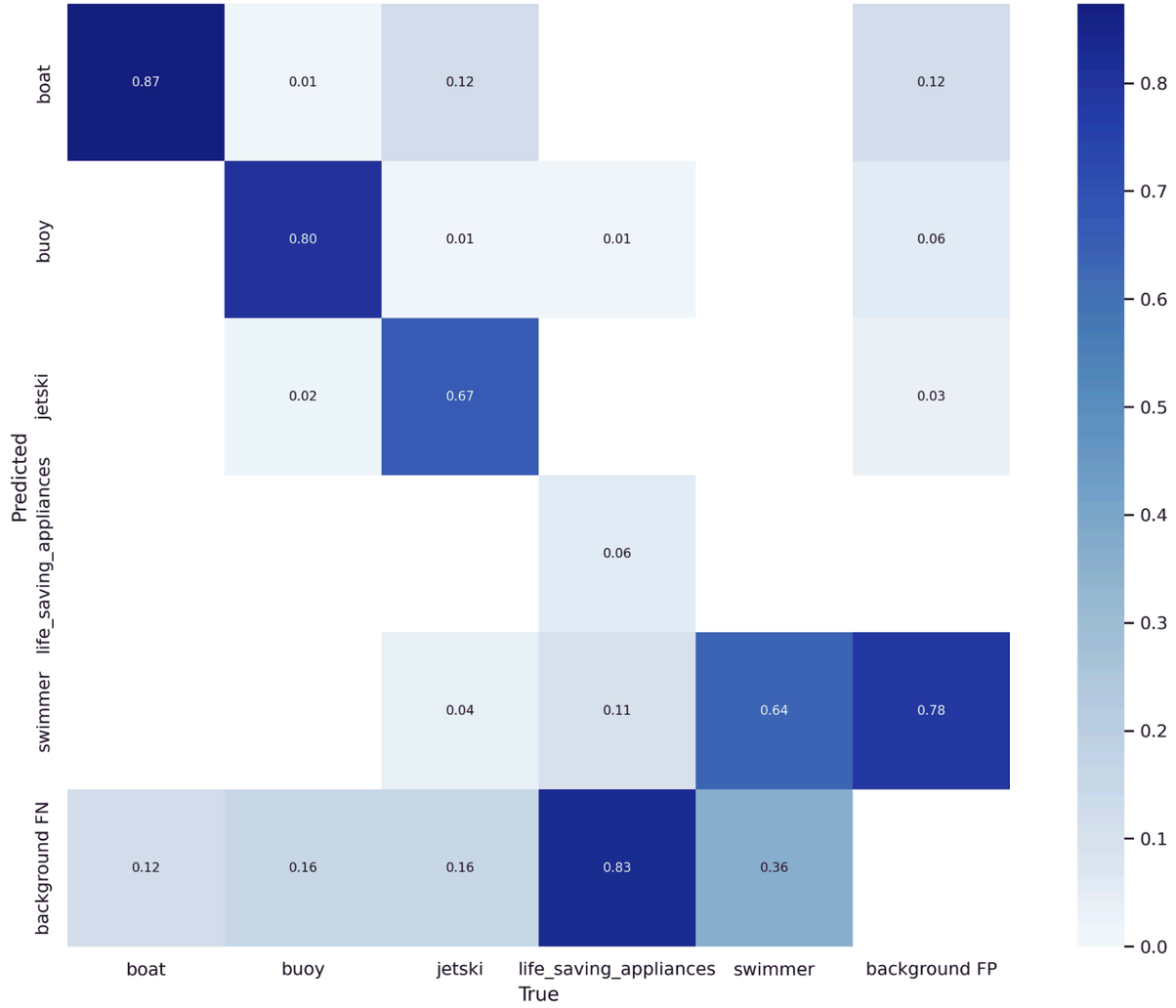
RESULTS

Model	Params (M)	mAP@0.5:0.95	mAP@0.5	Precision	Recall	F1-Score	Inference Time (ms)
YOLOv6N6	10.34	0.346	0.602	0.663	0.61	0.635	1.78
YOLOv6S6	41.32	0.375	0.636	0.66	0.68	0.67	4.99
YOLOv6M6	79.53	0.380	0.642	0.675	0.69	0.683	8.03
YOLOv6L6	140.21	0.392	0.658	0.69	0.69	0.69	16.32

RESULTS


swimmer 0.58
boat 0.93
swimmer 0.57
swimmer 0.65
swimmer 0.60
boat 0.95
swimmer 0.66
boat 0.93

EVALUATION


boat
buoy
jetski
life_saving_appliances
swimmer
background FN
Predicted
0.87
0.01
0.12
0.12
0.80
0.01
0.01
0.06
0.02
0.67
0.03
0.06
0.04
0.11
0.64
0.78
0.12
0.16
0.16
0.83
0.36
boat
buoy
jetski
life_saving_appliances
swimmer
background FP
True
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

DISCUSSION

1. The best performing model was YOLOv6L6 in terms of mAP, precision, recall and F1-score.

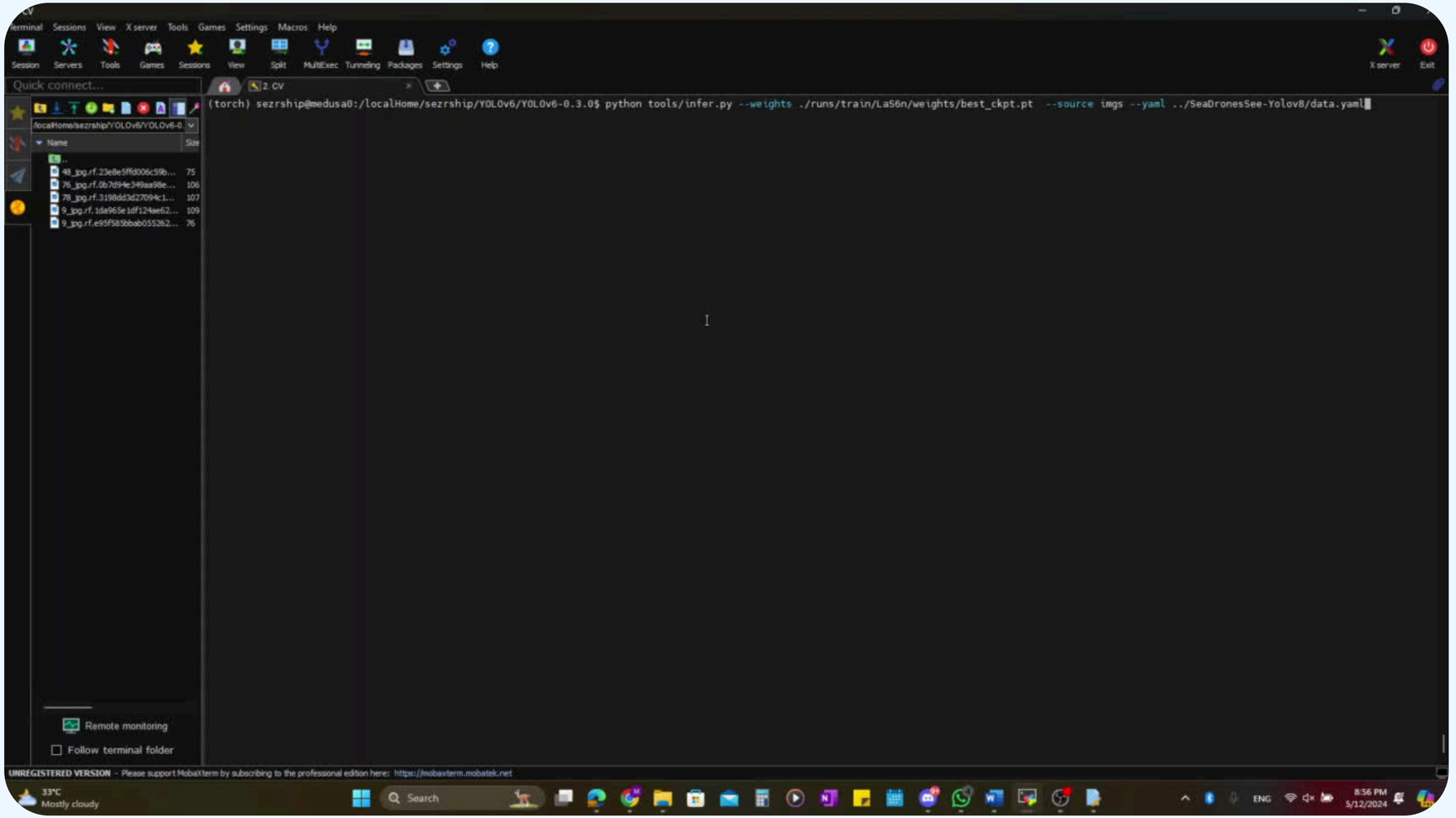
2. However, UAVs are edge devices with limited computational power, meaning we need to be careful with efficiency, inference time, etc.

3. Keeping edge devices in mind, the best model to use would be YOLOv6N6:
   - Smallest inference time
   - Smallest model
   - Comparable results

WORK DIVISION

Mohamed Alshafai 87311	Rana Elfakharany 87725	Hiba Saleem 87239
Ran models Recorded the video Evaluated models Prepared Github	Ran models Evaluated models Prepared Github Prepared presentation	Ran models Evaluated models Prepared Github Prepared presentation

DEMONSTRATION



If the video does not open, please click this link: Demo Video

THANK YOU!

REFERENCES

[1] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, “SeaDronesSee: A Maritime Benchmark for Detecting Humans in Open Water,” presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2260–2270. Accessed: Mar. 22, 2024. [Online]. Available: https://openaccess.thecvf.com/content/WACV2022/html/Varga_SeaDronesSee_A_Maritime_Benchmark_for_Detecting_Humans_in_Open_Water_WACV_2022_paper.html

[2] J. Sharafaldeen, M. Rizk, D. Heller, A. Baghdadi, and J.-Ph. Diguet, “Marine Object Detection Based on Top-View Scenes Using Deep Learning on Edge Devices,” in 2022 International Conference on Smart Systems and Power Management (IC2SPM), Nov. 2022, pp. 35–40. doi: 10.1109/IC2SPM56638.2022.9988928.

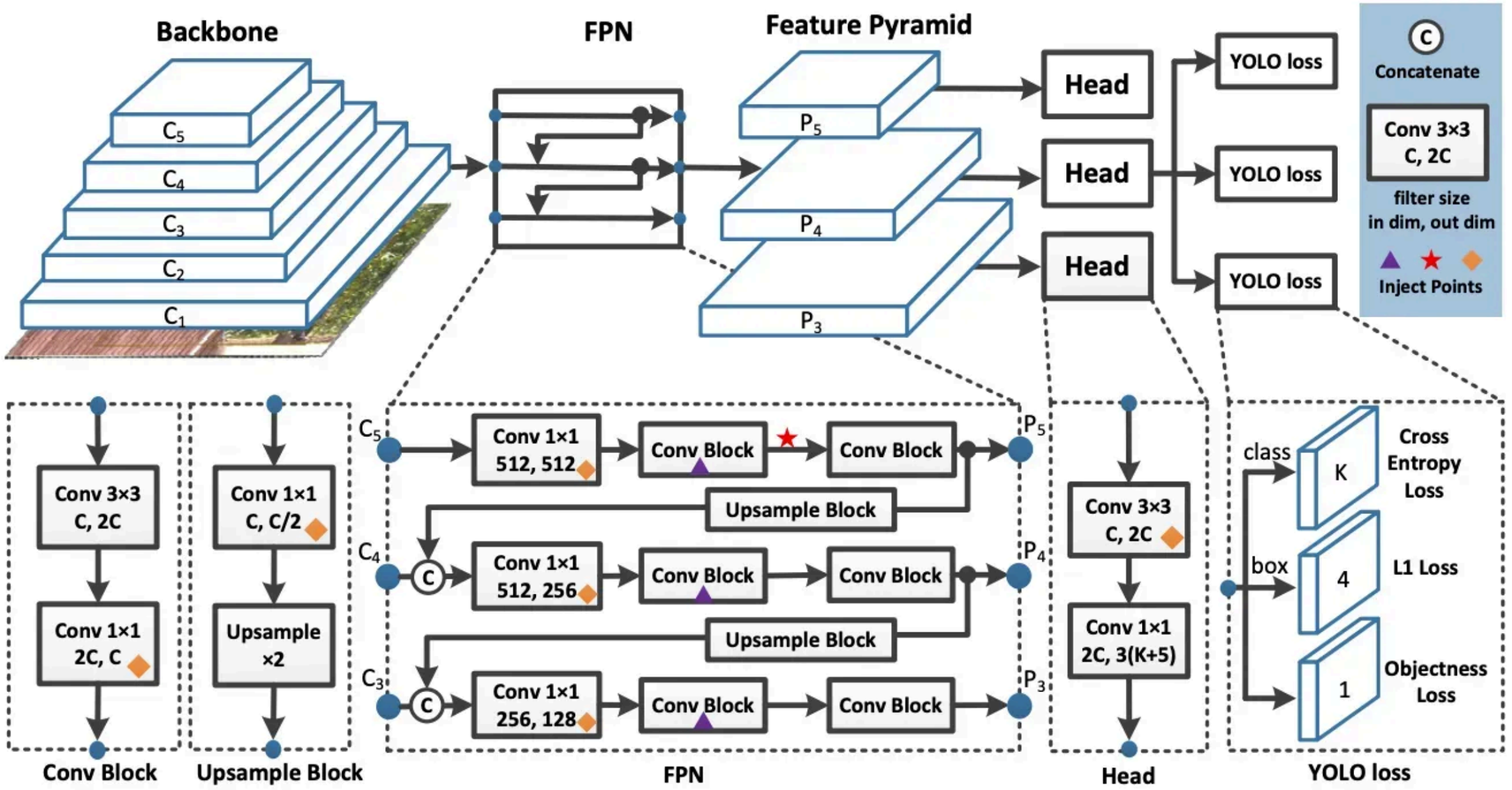
[3] E. Mulcahy, P. Van de Ven, and J. Nelson, “Aerial Object Detection for Water-Based Search & Rescue,” in Artificial Intelligence and Cognitive Science, L. Longo and R. O’Reilly, Eds., Cham: Springer Nature Switzerland, 2023, pp. 344–354. doi: 10.1007/978-3-031-26438-2_27.

[4] J. Jang, S. Oh, D. Seo, Y. Choi, Y. Kim, and H. J. Yang, “M2SODAI: Multi-Modal Maritime Object Detection Dataset With RGB and Hyperspectral Image Sensors”.

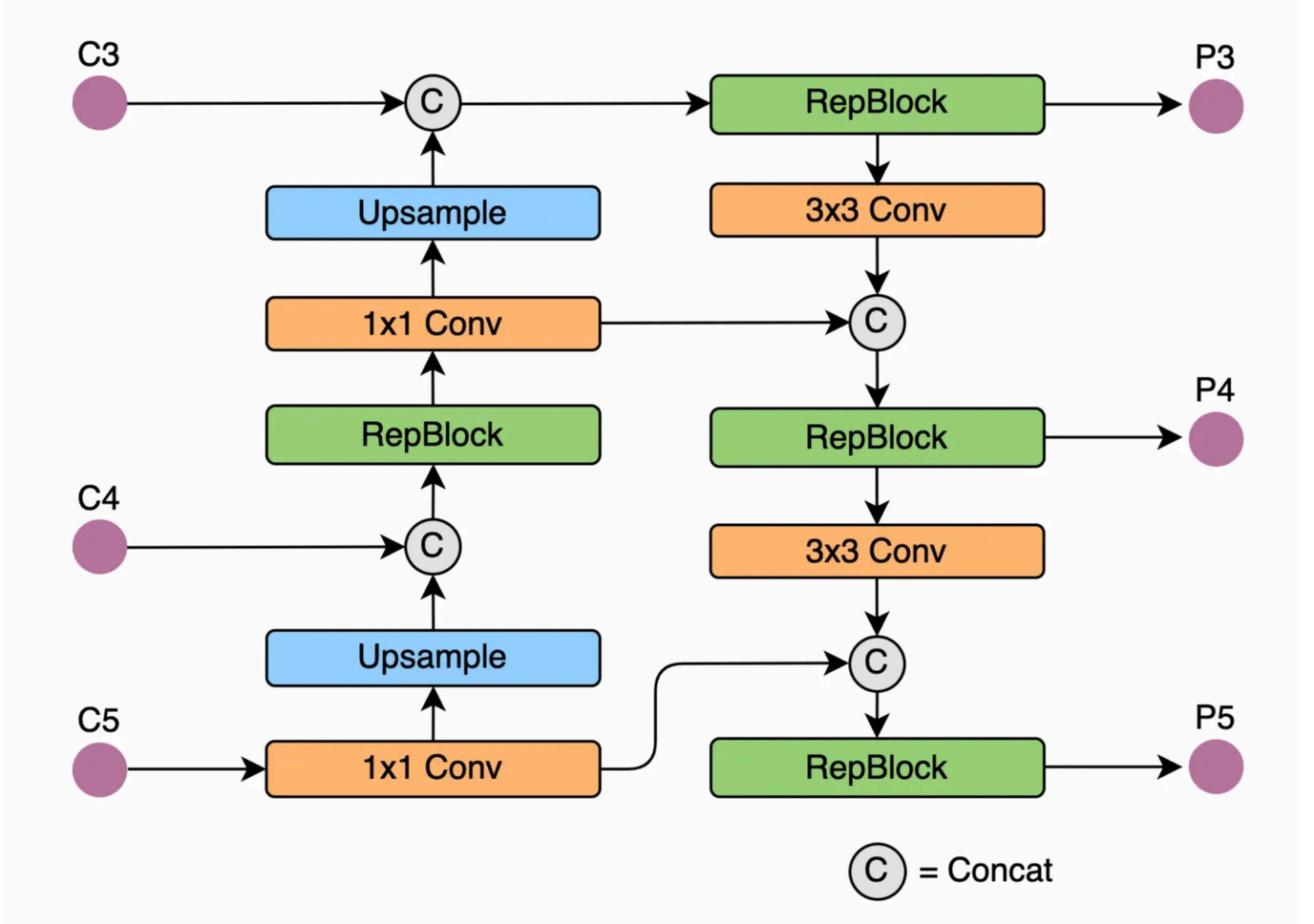
[5] “Remote Sensing | Free Full-Text | An Enhanced Target Detection Algorithm for Maritime Search and Rescue Based on Aerial Images.” Accessed: Mar. 24, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/15/19/4818

APPENDIX

MODEL ARCHITECTURE



MODEL ARCHITECTURE



MODEL ARCHITECTURE