# **Project: Final Report**

# **Submarine for object detection**

Group 3: Gagandeep Singh, Wai Leung Ma, Pik Yu Ng

Red River College Polytechnic-Data Science and Machine Learning

COMP-3704: Neural Networks and Deep Learning

Instructor Baha Rababah

# Project Contribution

Gagandeep Singh – Finding dataset from Roboflow Human, Building and train model (CNN, YOLOv11), Project documents (Proposal, Project Mid Progress, Final report, Presentation)

Wai Leung (Cyrus) Ma – Finding underwater dataset from Kaggle, Building and train model (VGG16), Project documents (Proposal, Project Mid Progress, Final report, Presentation)

Pik Yu (Allie) Ng – Building and train model (ResNet50), Project documents (Proposal, Project Mid Progress, Final report, Presentation)

# Introduction and Problem Statement

Rescue missions in underwater environments, particularly those involving submarines, face significant challenges due to the unique and visually complex nature of underwater imagery. Identifying human bodies accurately and efficiently in such conditions is crucial for successful rescue operations. However, existing datasets and models often fail to account for the distinct visual characteristics of underwater environments, such as low visibility, distortion, and the presence of debris.

This project aimed to develop a robust object detection system using bounding boxes to locate humans in underwater environments. The goal was to improve the accuracy and reliability of human detection in real-time scenarios, facilitating faster response times and increasing the likelihood of successful rescues. By leveraging state-of-the-art deep learning techniques and innovative data preprocessing strategies, this project addressed the limitations of existing models and datasets, creating a tailored solution for submarine rescue missions.

# Dataset Description

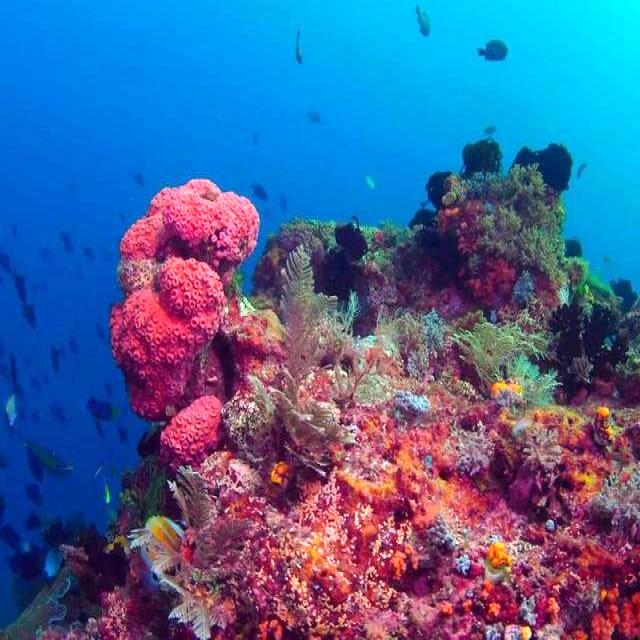
1. **Disaster Dataset:** This dataset has images of humans under different types of objects that could simulate a disaster or a wreckage.



1. **Range Dataset:** This dataset has images of human in varying distance.



1. **Kaggle Underwater Dataset**: Includes labeled underwater images of multiple classes, such as humans and debris, useful for training in underwater conditions.



1. **Roboflow Human Dataset**: Contains annotated images of humans in various settings, aiding in human detection tasks. (This is the one that we finally ended up using.)



# Model Description

The final deep learning model used for this project was **YOLOv11**, a state-of-the-art object detection model. YOLOv11 builds upon the YOLO (You Only Look Once) series, which is well-known for its speed and accuracy in real-time object detection tasks. It was selected for its ability to detect and localize objects in challenging environments like underwater settings, where visibility and image quality can vary significantly.

## Architecture Overview:

### Backbone:

* YOLOv11 utilizes a **CNN-based backbone** for feature extraction, pre-trained on large-scale datasets. This ensures the model can capture both low-level (edges, textures) and high-level (object parts) features.
* The backbone leverages an improved version of CSPNet (Cross Stage Partial Networks), which reduces computation while maintaining feature learning efficiency.

### Neck:

* The model employs a **Feature Pyramid Network (FPN)** with PANet (Path Aggregation Network) to handle multi-scale feature aggregation. This enables the model to detect objects at varying sizes, essential for underwater images with diverse object scales.
* This component also helps in retaining contextual information, which is critical for distinguishing humans from underwater debris or other objects.

### Head:

* The detection head predicts bounding boxes, class probabilities, and objectness scores using anchor-free detection mechanisms. The anchor-free approach enhances detection speed and reduces the computation needed to handle numerous anchor boxes.

### Activation Functions:

* Leaky ReLU is used in hidden layers for non-linearity.
* Sigmoid activation is applied to the output layers for bounding box coordinates and class confidence scores.

### Layers:

1. **Convolutional Layers**: Used for feature extraction with kernels of varying sizes, capable of detecting objects of multiple scales.
2. **Batch Normalization**: Ensures stable and faster training by normalizing inputs to each layer.
3. **Skip Connections**: Introduced to prevent vanishing gradient problems and improve gradient flow.
4. **SPP (Spatial Pyramid Pooling)**: Adds a spatial pyramid to improve the detection of smaller objects in cluttered environments.

### Hyperparameters:

1. **Learning Rate**: Initial learning rate: 0.001 with a cosine decay scheduler to reduce the learning rate during training for better convergence.
2. **Batch Size**: Set to 16, which balanced memory constraints and training efficiency.
3. **Optimizer**: AdamW Optimizer: Chosen for its adaptive learning rate properties and weight decay for regularization.
4. **Loss Function**: Multi-task loss combining:
   * **CIoU Loss**: For accurate bounding box regression.
   * **Binary Cross-Entropy Loss**: For object classification.
   * **Focal Loss**: To handle class imbalance effectively.
5. **Epochs**: The model was trained for around 100 epochs with early stopping to prevent overfitting.

### Optimization Process:

1. **Pretrained Weights**: YOLOv11 was initialized with weights pretrained on the COCO dataset, providing a strong baseline for feature extraction.
2. **Data Augmentation**: Extensive augmentation techniques, such as flipping, scaling, rotation, and color jittering, were used to make the model robust to underwater conditions.
3. **Fine-Tuning**: The model was fine-tuned on the final labeled dataset using a reduced learning rate to ensure it adapted well to the specific underwater object detection task.

YOLOv11's advanced architecture and fine-tuned hyperparameters made it the ideal choice for achieving accurate and reliable human detection in submarine rescue missions, ensuring it performed well even in challenging underwater environments.

## Experiments and Results

Model Architectures and Training:

1. Data Modification:
   * We tried using a CycleGAN model to modify the data that we had to underwater images.

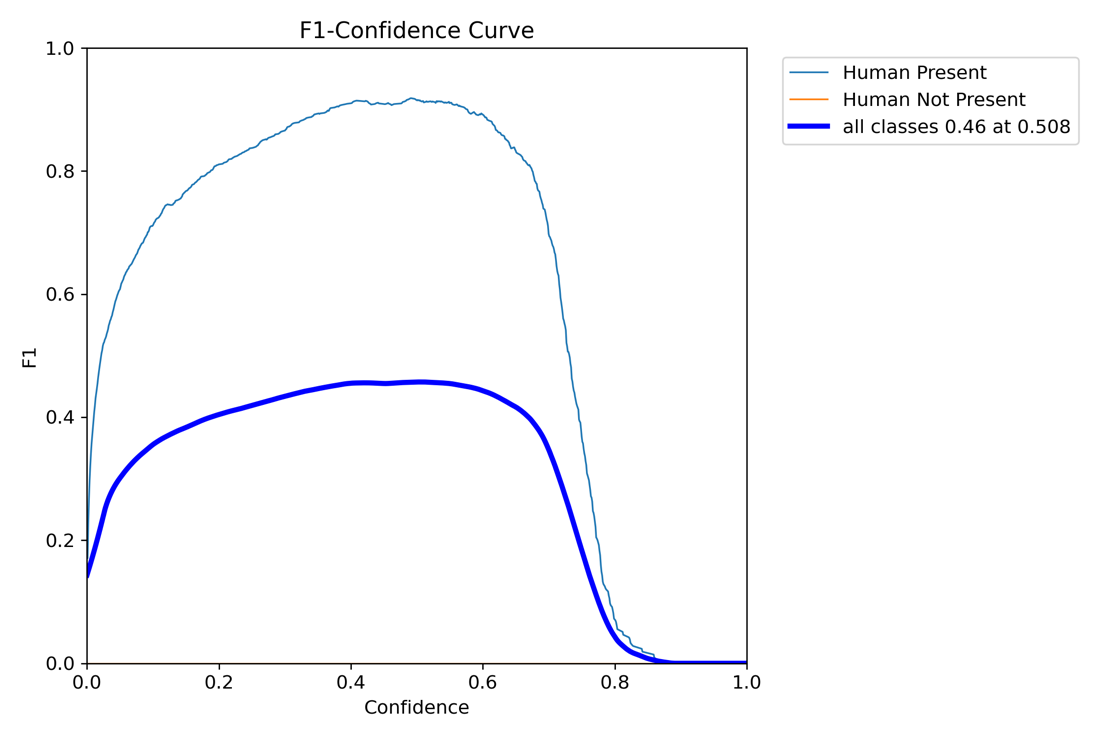
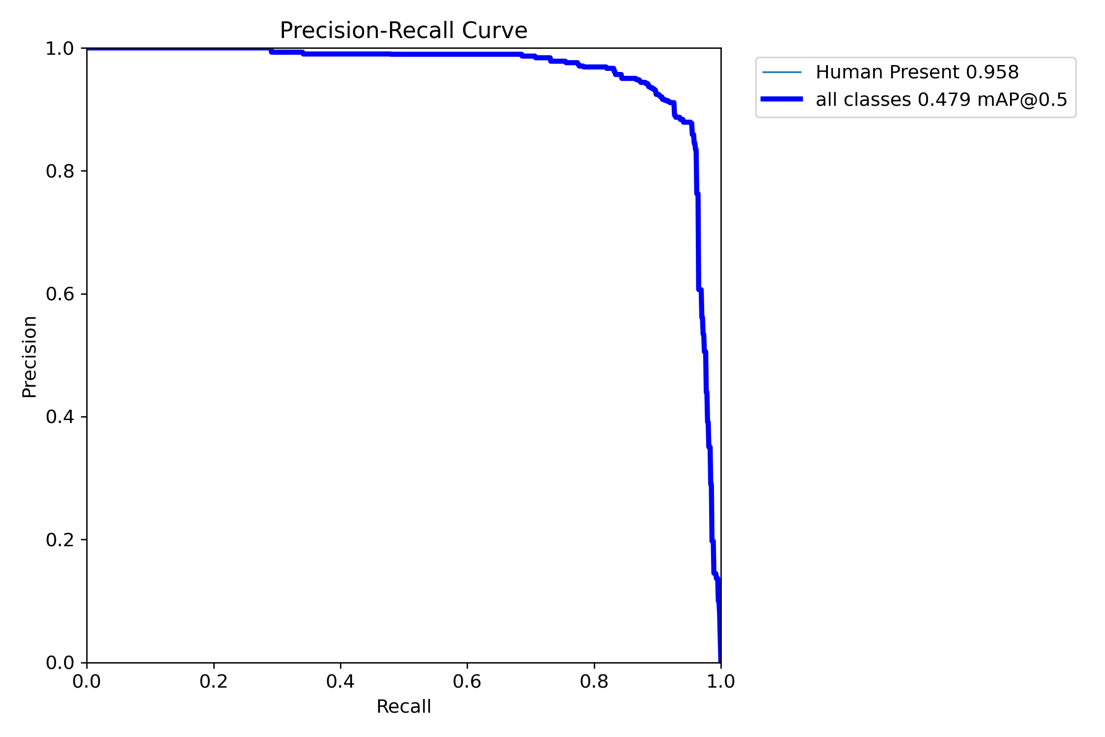


1. Initial Models
   * ResNet50 and VGG16:
     + These pre-trained architectures were fine-tuned for object detection tasks.
   * Results: The models performed poorly in detecting underwater humans due to their design for image classification rather than object detection.
2. Custom Lightweight Models:
   * Simple architectures were tested for faster inference.
   * Results: The models were too simplistic, leading to significant underfitting.
3. YOLOv11 Model
   * Description: A state-of-the-art object detection model known for real-time performance.
   * Hyperparameters:
     + Input size: Resized images to 640x640
     + Confidence threshold: 0.25
     + IoU threshold: 0.5
   * Results: Achieved the highest mAP and the most reliable detection across underwater images.

## Performance Metrics:

Training and validation losses decreased steadily, as shown in the provided graphs**A group of graphs showing results

Description automatically generated with medium confidence** Precision-recall and F1-confidence curves highlighted the model's robust performance.



## A screenshot of a computer Description automatically generatedA collage of images of a person under water Description automatically generatedResult:

## Challenges

## The image dataset's accuracy is often affected by the challenging conditions present underwater, which typically include low visibility, poor lighting, color distortion, and high levels of noise.

## The training process was time-consuming, with a single run of 100 epochs taking approximately 15-20 hours. Unfortunately, the results obtained were not satisfactory, necessitating the need to rerun the training process.

## Conclusion

## While ResNet50, VGG16, and basic CNN models are generally effective for image classification, YOLOv11 stands out for superior performance in object detection tasks, particularly excelling in detecting humans underwater in this project. This model demonstrates high confidence levels and overall superior performance. However, due to time constraints and limited computational power, we only trained it for around 100 epochs. Enhancing this model's performance could involve running additional epochs or incorporating extra hidden layers if deemed necessary.

## Appendix

1. **Figures and Charts**: Include the provided graphs of loss curves, precision-recall, and F1-confidence metrics.
2. **Code Snippets**: Add critical code blocks related to data preprocessing, model architecture, and training.
3. **Additional Results**: Showcase extended visualizations and additional evaluation metrics

.

**References**

1. Roboflow. (n.d.). Human Dataset. Retrieved from <https://universe.roboflow.com/school-lozls/human-ighgb-rl2ck/browse>
2. Akshat S. (2024). Underwater Dataset for 8 Classes with Labels [Dataset]. Kaggle. Retrieved from <https://www.kaggle.com/datasets/akshatsng/underwater-dataset-for-8-classes-with-label>
3. Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2242–2251. https://doi.org/10.1109/ICCV.2017.244
4. Human Dataset. (2024, September). Roboflow Universe. Roboflow. Retrieved December 5, 2024, from <https://universe.roboflow.com/school-lozls/human-ighgb-rl2ck>
5. Drowning Detection Dataset. (2024, November). *Roboflow Universe*. Roboflow. Retrieved December 5, 2024, from <https://universe.roboflow.com/ardian-yyb0l/drowning-detection-b3lye>