Report: Training YOLOv3 on the Aquarium Dataset with MMDetection

Approach

The task involved training the YOLOv3 object detection model on the Aquarium Dataset, which consists of 638 images labeled for seven classes: fish, jellyfish, penguins, sharks, puffins, stingrays, and starfish. This dataset presents class imbalance, as classes like puffins, stingrays, and starfish are underrepresented, reflecting common real-world data challenges.

The steps followed for training were:

1. **Dataset Preparation**: The dataset, split into training, validation, and test sets, was downloaded in Pascal VOC format. Using the `prepare_voc_format.py` script, it was organized to fit the expected directory structure for MMDetection, including annotations in XML files and images in JPEG format.
2. **Model Configuration**: The YOLOv3 configuration file was customized to accommodate our dataset, setting the number of classes to seven, updating batch size, learning rate, and paths. Pretrained YOLOv3 weights, fine-tuned on the COCO dataset, were downloaded and used as initialization to speed up training.
3. **Training Setup**: The training was conducted with MMDetection's training API, with checkpointing and evaluation intervals set for validation on each epoch. The model was trained over 100 epochs using an RTX 3080 GPU.

Key Findings

1. **Model Performance**: The trained model achieved a mean Average Precision (mAP) of 0.651 on the validation set, which is a satisfactory result given the relatively small dataset and lower input resolution (320x320). Class-specific AP scores showed variability: common classes like jellyfish and fish had higher APs (0.849 and 0.652, respectively), while rarer classes like puffin had lower scores (0.482).
2. **Class Imbalance Impact**: Performance disparity among classes highlights the impact of class imbalance. Classes with fewer instances, such as puffins and starfish, had comparatively lower APs, indicating that limited training samples affect detection accuracy.

Recommendations for Future Work

1. **Data Augmentation**: Implementing advanced data augmentation techniques could help address class imbalance by synthetically expanding underrepresented classes, potentially improving the model's robustness and overall AP scores.
2. **Higher Resolution Images**: Using higher resolution images (e.g., 416x416 or 608x608) could improve the model’s ability to detect small or occluded objects, particularly for minor classes where finer details might help with recognition.
3. **Hyperparameter Tuning**: Experimenting with different hyperparameters, such as learning rates, batch sizes, and optimizers, may help in achieving better convergence and potentially higher mAP scores.
4. **Class Weighting**: Incorporating class weights in the loss function could improve learning for underrepresented classes, balancing model focus across all categories.

This experiment demonstrates that YOLOv3, even with limited data, can achieve reasonable detection accuracy in a multi-class aquarium environment. Further improvements could yield a more robust model suitable for real-time applications in similar aquatic or zoological monitoring contexts.