MMDetection

1. Dataset Overview

- The Aquarium dataset, sourced from Roboflow, includes 638 images captured from two U.S. aquariums, annotated with seven classes: fish, jellyfish, penguins, sharks, puffins, stingrays, and starfish. The distribution of these classes is unbalanced, with puffins, stingrays, and starfish underrepresented, presenting a common challenge in real-world datasets where class imbalance can impact model performance. The dataset is split into 448 training images, 127 validation images, and 63 test images.

2. Directory and Dataset Preparation

- The data is structured in the Pascal VOC format to align with MMDetection’s requirements. The setup includes organizing JPEG images and XML annotation files under directories for training, validation, and testing. The `prepare_voc_format.py` script automates the restructuring, copying images, and generating train/val text files for correct model data processing.

3. Model and Configuration Setup

- The training utilizes a pretrained YOLOv3 model with a Darknet-53 backbone, optimized on the COCO dataset. This model is chosen for its balance between speed and real-time processing. The configuration file `cfg.py` is modified to:
  - Point to the dataset path.
  - Set batch size, learning rate, and other hyperparameters.
  - Define specific parameters like checkpointing intervals and evaluation metrics.
- The training configuration specifies seven classes and a Pascal VOC directory setup, leveraging MMDetection's XMLCustomDataset class.

4. Training Process

- The model is trained over 100 epochs with checkpoints saved every 15 epochs. Performance metrics, specifically mean Average Precision (mAP), are computed per class and overall. Notably, smaller class sizes (e.g., puffin and starfish) result in comparatively lower AP scores due to limited data.

5. Results

- Final evaluation reports a mean Average Precision (mAP) of 65.1%, with fish and jellyfish classes scoring the highest AP due to a greater number of training examples. Model performance remains robust despite limited data, indicating the suitability of the YOLOv3 architecture for diverse aquatic classes.
- TensorBoard logs reveal training and validation trends, allowing further analysis of the model’s performance over the epochs.

6. Inference and Utilization

- Inference can be performed on new data using the trained model, enabling real-time object detection across images and videos. The outputs directory retains all inference results for further evaluation.