A brief report on Object Detection Model

By
Md. Rasel Uddin.

There are several models to detect objects, such as YOLOV8, Faster R-CNN, and SSD. YOLO is common and popular for object detection. Considering performance such as accuracy, Faster R-CNN is better than others. But the SSD model is faster and also compatible with the PASCAL VOC data format. As I annotated my dataset into this format, I used this model to implement a basic object detection model using the TensorFlow framework.

Model architecture: The model leverages pretrained MobileNetV2 as the backbone for feature extraction, providing a balance between computational efficiency and accuracy as I have a low computational experimental setup. The SSD architecture extends MobileNetV2 with a set of fully connected layers for bounding box regression. Key layers are:

Backbone: MobileNetV2 (pretrained on ImageNet, without top layers)
Feature extraction: Flatten, Dense (1024 units, ReLU activation), Dropout (0.3)
Bounding box regression: Dense layer predicting (NUM_CLASSES * 4) values, reshaped to (NUM_CLASSES, 4) to represent bounding box coordinates.
Loss function: Mean Squared Error (MSE) for bounding box regression

Training Methodology: I have implemented the dataset, which I annotated in this assessment task. The dataset consists of 150 images, which are split into 80:20 segments for training and validation. The model is trained on this Pascal VOC dataset, which includes labeled images with bounding box annotations in XML format. The dataset is preprocessed as follows:
Images are resized to 300×300 pixels. Bounding boxes are normalized relative to image dimensions. A maximum of 5 objects per image is considered (NUM_CLASSES=5). A custom data generator loads images and their corresponding bounding boxes in batches. The bounding boxes are represented as (xmin, ymin, xmax, ymax) coordinates normalized between 0 and 1. Training Configurations are as below:

Optimizer: Adam (learning rate = 0.0001)
Batch size: 8
Training epochs: 30
Train-validation split: 80%-20%
Augmentations: Data is shuffled during training for generalization

Evaluation Metrics and Results: The model is evaluated on the validation set using the Mean Squared Error (MSE) loss. The evaluation pipeline follows the same data generator approach as training. After training for 10 epochs, the model achieved the final validation loss 0.07690971344709396 at the end of training. The model successfully detects multiple objects per image.

The model effectively learns bounding box regression but may require additional tuning to enhance detection accuracy. Performance could be further improved by incorporating intersection over union as an evaluation metric and leveraging focal loss instead of MSE.
Additional augmentations and a larger dataset could enhance robustness.