Object Detection with RetinaNet: Theory and Implementation

Introduction

This project focuses on implementing the **RetinaNet** model, a state-of-the-art object detection architecture designed to excel at detecting objects, especially small ones. RetinaNet is a **single-stage** model that outperforms many two-stage models like Faster R-CNN in both speed and accuracy, making it suitable for real-time applications. This repository (`https://github.com/Coluding/RetinaNet`) provides a comprehensive implementation, including training and evaluation code, pre-trained models, and a detailed README.

The primary objective was to gain a theoretical understanding of the RetinaNet model and its applications in object detection.

RetinaNet Architecture

Single-Stage Object Detection

Unlike two-stage models that first generate region proposals (e.g., Faster R-CNN), RetinaNet directly predicts object bounding boxes and class scores in a single pass. This makes it faster and computationally more efficient.

Key Components

RetinaNet combines several key ideas for efficient and accurate detection:

- **Feature Pyramid Network (FPN)**: Provides multi-scale feature maps to detect objects of different sizes.
- **Anchor Boxes**: Predefined boxes of various scales and aspect ratios centered at each location of the feature maps.
- **Focal Loss**: Addresses the class imbalance problem inherent in single-stage detectors.

Mathematical Formulation

Anchor Boxes

Given an image, RetinaNet generates a set of **anchor boxes** A with different scales s and aspect ratios r for each location in the feature map:

$$A = \left\{ (x, y, w, h) \mid w = s\sqrt{r}, h = \frac{s}{\sqrt{r}} \right\}$$

where (x, y) is the anchor center, and (w, h) are the width and height of the anchor.

Focal Loss

The **Focal Loss** is a modified version of the Cross Entropy Loss designed to address the class imbalance between foreground and background objects. It reduces the loss contribution from easy-to-classify examples, focusing on hard examples:

$$\text{FL}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

where:

- p_t is the predicted probability of the ground-truth class.
- α_t is a weighting factor for class balance.
- γ (typically 2) is a focusing parameter that reduces the loss contribution from easy examples.

Bounding Box Regression

The model predicts bounding box offsets relative to the anchor boxes. Let (x_a, y_a, w_a, h_a) be the anchor box parameters and (x, y, w, h) the ground-truth box. The network predicts offsets (t_x, t_y, t_w, t_h) such that:

$$t_x = \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a}, \quad t_w = \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right)$$

The predicted bounding box (x', y', w', h') is then computed as:

$$x' = t_x w_a + x_a, \quad y' = t_y h_a + y_a, \quad w' = w_a e^{t_w}, \quad h' = h_a e^{t_h}$$

Training and Implementation

The repository includes code for:

- **Data Preprocessing**: Loading and augmenting datasets (e.g., COCO).
- **Model Training**: Using the Focal Loss and backpropagation to optimize the network.
- **Model Evaluation**: Measuring performance using metrics like **Mean Average Precision (mAP)**.

Optimization

- **Optimizer**: Adam or SGD
- **Learning Rate**: $\eta = 0.001$
- **Batch Size**: 16
- **Epochs**: 50

Evaluation Metrics

RetinaNet's performance is evaluated using:

- **Mean Average Precision (mAP)**: The average precision across all classes and IoU thresholds.
- **Intersection over Union (IoU)**: The ratio of the intersection and union of predicted and ground-truth boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Theoretical Insights

This project offered a deeper understanding of:

- How **single-stage** object detectors like RetinaNet achieve high efficiency compared to two-stage models.
- The role of **Focal Loss** in addressing class imbalance.
- The importance of **multi-scale feature maps** (via FPN) in detecting objects of different sizes.

Conclusion

RetinaNet's combination of speed and accuracy makes it ideal for real-time object detection tasks. The project provided a hands-on exploration of RetinaNet's architecture and its theoretical foundations, making it a valuable resource for understanding modern object detection.