Faster R-CNN Assignment

1. Architecture of Faster R-CNN & Component Roles

Faster R-CNN is a two-stage object detection framework consisting of:

1. **Backbone CNN (e.g., ResNet, VGG)**: Extracts hierarchical features from the input image.
2. **Region Proposal Network (RPN)**: Generates potential object bounding boxes (region proposals) using anchor boxes.
3. **RoI (Region of Interest) Pooling**: Resizes variable-sized proposals to fixed dimensions for the detector.
4. **Fast R-CNN Detector**: Classifies proposals into object classes and refines their bounding box coordinates.

Pipeline Workflow:

- The backbone processes the image to produce feature maps.
- The RPN scans these maps with anchor boxes to propose regions likely to contain objects.
- RoI pooling standardizes proposal sizes.
- The Fast R-CNN head performs final classification and bounding box regression.

---

2. Advantages of RPN Over Traditional Methods

Traditional methods like Selective Search rely on handcrafted algorithms to propose regions, which are:

- **Slow**: Run on CPU and lack parallelization.
- **Inflexible**: Use fixed heuristics unsuitable for diverse objects.

The RPN improves this by:

- **End-to-end learning**: Integrates proposal generation with detection, optimizing both jointly.
- **Speed**: Runs fully on GPU, enabling near real-time performance.
- **Adaptability**: Learns to propose regions tailored to the dataset via anchor boxes.

---

3. Training Process: Joint RPN & Fast R-CNN Training

Faster R-CNN uses a **4-step alternating training strategy**:

1. **Train RPN**: Initialize with a pre-trained backbone; optimize for generating high-quality proposals.
2. **Train Fast R-CNN**: Use proposals from the RPN to train the detector.
3. **Fine-tune RPN**: Fix the detector and refine the RPN to improve proposals.

4. **Fine-tune Fast R-CNN**: Fix the RPN and further optimize the detector.

Key Mechanisms:

- **Multi-task loss**: The RPN simultaneously predicts objectness (foreground/background) and bounding box adjustments.
- **Weight sharing**: The backbone CNN's features are shared between the RPN and detector, ensuring consistency.

---

4. Anchor Boxes in the RPN

Anchor boxes are predefined boxes of multiple scales (e.g., 64×64, 128×128) and aspect ratios (e.g., 1:1, 1:2, 2:1). They serve as references to detect objects of varying shapes.

How They Work:

- Anchors are placed at each sliding window location on the feature map (e.g., 9 anchors per position).
- The RPN predicts:
  - An **objectness score** (probability the anchor contains an object).
  - **Offset values** to adjust the anchor's coordinates for better fit.
- The top-N anchors (e.g., 300) with the highest scores become region proposals.

Example: For a feature map position, anchors might include a tall box (1:2) for pedestrians and a wide box (2:1) for vehicles.

---

5. Performance Evaluation on COCO & Pascal VOC

Strengths:

- **Accuracy**: Achieves ~76% mAP on Pascal VOC and ~42% mAP on COCO, outperforming earlier models.
- **Precision**: Two-stage design reduces false positives compared to one-stage detectors.
- **Versatility**: Handles multi-class detection effectively.

Limitations:

- **Speed**: Processes 5-7 FPS (Pascal VOC) and 3-5 FPS (COCO), slower than YOLO or SSD.
- **Complexity**: Requires tuning anchor box hyperparameters.
- **Memory**: High resource usage due to RoI pooling and two-stage processing.

Improvement Areas:

- **Efficiency**: Replace RoI pooling with RoI Align (used in Mask R-CNN) for better accuracy.
- **Anchor-free designs**: Reduce dependency on anchor boxes (e.g., FCOS, CenterNet).

- **Lightweight backbones**: Use MobileNet or EfficientNet for faster inference.