Problem1

1. Faster R-CNN for Object Detection:
    - New Addition: Region Proposal Network (RPN).
    - Components: CNN backbone, RPN, and region-based classifier.
    - Training: Pre-trained CNN backbone, RPN trained for region proposals, and classifier fine-tuned for object classification and bounding box regression.
    - Inference: RPN generates region proposals, passed to the classifier for object detection.
2. Mask R-CNN for Instance Segmentation:
    - New Addition: Mask branch.
    - Components: Faster R-CNN components + mask branch.
    - Training: Object detection, bounding box regression, and mask branch trained for pixel-level segmentation masks.
    - Inference: Object detection + instance-level segmentation masks.
3. Mask R-CNN for Human Pose Estimation (Keypoint R-CNN):
    - New Addition: Keypoint branch.
    - Components: Mask R-CNN components + keypoint branch.
    - Training: Object detection, bounding box regression, mask generation, and keypoint branch trained for human joint keypoints.
    - Inference: Object detection, instance segmentation, and human pose estimation with keypoints.

In summary, the evolution involves adding new branches and components to the original Faster R-CNN framework. Mask R-CNN introduces instance segmentation, while Keypoint R-CNN extends it to human pose estimation. The training and inference strategies are adjusted accordingly to incorporate the additional tasks. These advancements have significantly improved the accuracy and versatility of RCNN models in various computer vision applications.

Problem 2

PointNet and PointNet++ utilize a simple symmetric function for point cloud classification to ensure the model's predictions are invariant to point permutations. This function treats different point orders as equivalent, enabling classification regardless of point order. In contrast, point cloud segmentation focuses on labeling individual points, so a symmetric function is not needed. The introduction of T-Net in PointNet and PointNet++ allows the model to learn spatial transformations for aligning and normalizing input point clouds. This improves performance by capturing geometric patterns and achieving better accuracy and robustness in tasks like object classification and part segmentation.