系統 112 F14081046 周呈陽

1. (15%) Can you illustratively describe the evolutions of three widely used RCNN families, from Faster RCNN for object detections, mask RCNN for instance segmentation and mask RCNN for human pose estimation (i.e., Keypoint RCNN). In each evolution, what are the new additions of components and their corresponding training and inference strategies.

A. Faster RCNN for Object Detection:
   - Addition: Introduction of Region Proposal Network (RPN) for generating region proposals.
   - Training Strategy: Pretrain shared backbone network, fine-tune RPN and Fast R-CNN.
   - Inference Strategy: Extract features, generate region proposals with RPN, perform object classification and bounding box regression with Fast R-CNN.

B. Mask RCNN for Instance Segmentation:
   - Addition: Additional Mask branch for pixel-level segmentation of proposed regions.
   - Mask Prediction: Generating binary masks for each proposed region.
   - Training Strategy: Introduce mask loss to measure accuracy of predicted masks.
   - Inference Strategy: Same as Faster RCNN, with the addition of generating segmentation masks for proposed regions.

C. Keypoint RCNN for Human Pose Estimation:
   - Addition: Additional Keypoint branch for detecting and localizing human keypoints.
   - Keypoint Prediction: Regressing coordinates of human keypoints.
   - Training Strategy: Introduce keypoint loss to measure accuracy of predicted keypoints.
   - Inference Strategy: Similar to Faster RCNN and Mask RCNN, while simultaneously predicting segmentation masks and keypoints.

2. (10%) Why do we always need a simple symmetric function in the PointNet and PointNet++ to perform the classification of point cloud data? On the other hand, why we do not need this simple symmetric function for segmentation of point cloud data? Can you also verbally describe what is the necessity of introducing T-Net in the PointNet and PointNet++, what is effect of using T-Nets and the evidence of performance improvement.

In PointNet and PointNet++, a simple symmetric function is used for point cloud classification to ensure that the model is invariant to the order of points. This allows the model to capture global information about the point cloud regardless of the point order. However, for point cloud segmentation, the order of points is important as it determines the spatial relationships between neighboring points. To address this, PointNet and PointNet++ introduce T-Net (Transformation Network), which learns an alignment transformation for each input point cloud. T-Net aligns the point cloud into a canonical coordinate system, allowing subsequent network layers to capture local structures and preserve spatial relationships. The introduction of T-Net improves segmentation performance by enabling the model to accurately assign semantic labels to individual points. Experimental results demonstrate that T-Net enhances the discriminative power of the network and leads to improved segmentation results on benchmark datasets.