Title: Instance Segmentation on a COCO-2017 Subset using Mask R-CNN

Introduction & Literature Review

Instance segmentation aims to detect objects in an image and delineate each object with a unique mask. A widely adopted baseline for this task is Mask R-CNN (He et al., 2017), which extends Faster R-CNN by adding a dedicated mask prediction branch on top of region proposals. Earlier segmentation architectures, such as U-Net (Ronneberger et al., 2015), helped popularise encoder-decoder designs for dense pixel prediction. DeepLabv3+ (Chen et al., 2018) introduced atrous spatial pyramid pooling for better boundary refinement.

For this project, Mask R-CNN is a natural choice. It directly outputs instance-level masks and bounding boxes, integrates seamlessly with COCO-style annotations, and is conveniently implemented in torchvision, making it suitable for efficient experimentation in a limited timeframe.

Dataset & Exploratory Analysis

The dataset is a curated subset of **COCO-2017**, consisting of 300 training images and 300 validation images, plus an unlabelled set of 30 test images. From the original COCO annotations, only four classes are retained: **person**, **cat**, **sports ball**, and **book**.

Basic EDA revealed a clear class imbalance, with *person* dominating the dataset and *book* and *sports ball* appearing far less frequently (as shown in the accompanying bar plot). Visual inspection showed significant variation in object scale and many partially occluded objects. This further motivated the use of a **feature pyramid network (FPN)** backbone, since it helps handle multi-scale features and improves detection of small objects.
To aid generalization, standard augmentations were applied: horizontal flips, mild rotations, and brightness/contrast changes.

Methodology

The model is built using **Mask R-CNN with a ResNet-50 FPN** backbone from torchvision. The classification and mask heads were modified to predict **five categories** (background + the four target classes). A custom dataset loader parses the COCO-format JSON files and constructs per-image targets containing bounding boxes, masks, and class labels.

The training setup used **SGD** with a learning rate of 0.005, momentum of 0.9, and weight decay of 5e-4. The model was trained for six epochs are sufficient to observe clear learning trends, although additional epochs would likely yield further improvements. Checkpoints were saved after each epoch.

Evaluation involved two complementary metrics:

- **Dice coefficient**, to assess pixel-level mask quality
- **Greedy IoU matching**, where each ground truth mask is paired with the prediction that overlaps with it the most

COCO-style mean Average Precision (mAP) was also computed using pycocotools by converting predicted masks to RLE format.

Results

Training curves show a consistent decrease in total loss over the six epochs. Qualitatively, the model performs well on **person** and **cat**, producing coherent masks with reasonably clean boundaries. Performance drops for **sports ball**, particularly for very small objects, and **book**, which appears in diverse orientations and under occlusion. The initial COCO mAP (AP50:95) is modest, expected given the small dataset size and class imbalance and the notebook includes a full COCOeval breakdown.

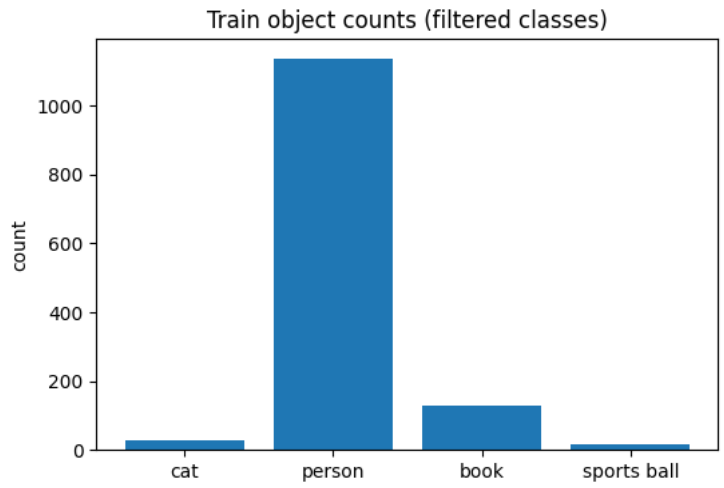

Figure 1 Train class counts

Figure 2 Sample images with masks

Discussion & Critical Analysis

The results reflect two main limitations: **dataset size** and **class imbalance**. Mask R-CNN is known to benefit greatly from large, varied datasets, so the scarcity of minority classes directly impacts recall and mask quality. Small objects also pose a challenge, partly due to input resolution and default anchor scales. Larger training resolutions or multi-scale training strategies could address this.

While basic augmentations helped reduce overfitting, stronger or more diverse augmentations (e.g., random crops, scale jittering) may further improve robustness. The chosen backbone (ResNet-50) offers a good balance of speed and capacity, but deeper backbones such as ResNet-101 or more modern architectures could yield higher accuracy at the cost of increased compute time.

Conclusion & Future Work

Mask R-CNN provided a solid and interpretable baseline for instance segmentation on this COCO subset. Despite the dataset limitations, the model achieved reasonable performance on common classes and delivered usable instance masks.

Future work could explore:

1. **Longer training schedules** and learning-rate decay.
2. **Stronger and multi-scale augmentations** to improve robustness.
3. **Alternative architectures**, such as Cascade Mask R-CNN or DeepLabv3+ to enhance segmentation detail.
4. **Hyperparameter tuning** targeting small-object detection, including modified anchor sizes and higher-resolution inputs.

The accompanying Colab notebook contains all source code, training logs, evaluation metrics, and visualized predictions. Final test predictions are saved separately for inspection.

References

- He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 22-29 October 2017, 2980-298. https://doi.org/10.1109/ICCV.2017.322 .
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*. https://arxiv.org/abs/1505.04597
- Chen, L., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *ArXiv*. https://arxiv.org/abs/1802.02611 .

Appendix

Colab link:
https://colab.research.google.com/drive/1_zEceLjSXIvkWEyFjgJnURa5H-WbKHqY#scrollTo=G51rEQzJ4ghE