IMPLEMENTATION AND ANALYSIS OF FASTER RCNN AND MASK RCNN ARCHITECTURES FOR OBJECT DETECTION AND INSTANCE SEGMENTATION

KARTHIKEYA JAYARAMA [JKARTHIK@SEAS]

ABSTRACT. Object detection and instance segmentation are fundamental tasks in computer vision, essential for a range of applications, including autonomous driving and robotics. While many approaches have been proposed, they face challenges in real-time application and multi-class detection, especially when balancing accuracy and computational efficiency. Faster RCNN [5] introduced a Region Proposal Network (RPN) to reduce processing time by efficiently proposing object regions. Mask RCNN extends Faster RCNN by adding a mask prediction branch, enabling per-pixel segmentation of objects. This project aims to implement both Faster RCNN [5] and Mask RCNN [3] with ResNet50 [4] as the backbone, visualizing feature extraction layers and experimenting with Fine-Tuned Feature Pyramid Networks (FPN). By doing so, we intend to analyze how various feature levels contribute to object and mask detection and assess the models' performance on diverse datasets.
Keywords: Object detection, Instance Segmentation, Region Proposal Network, Feature pyramid network.

1. INTRODUCTION

Object detection and instance segmentation have evolved significantly, with key breakthroughs rooted in improvements in deep learning architectures. Initial approaches like R-CNN (Regions with CNN features) relied on region proposals from selective search, followed by separate steps for classification and regression, which were slow and computationally intensive. Fast R-CNN addressed some of these issues by integrating feature extraction and classification in a single CNN, but the process of generating region proposals was still a bottleneck.

Ren et al. (2015) proposed Faster R-CNN, introducing a Region Proposal Network (RPN) that effectively eliminated the need for external region proposal algorithms. This RPN generates region proposals directly from the feature map of a convolutional network, drastically reducing computation time and allowing real-time application in object detection tasks.

To address the need for instance segmentation, He et al. (2017) [3] developed Mask R-CNN, an extension of Faster R-CNN that adds a mask prediction branch. This branch performs per-pixel segmentation on each detected object, enabling precise instance-level segmentation, which is essential for tasks where spatial accuracy is crucial (e.g., medical imaging, autonomous driving). Mask R-CNN not only builds on Faster R-CNN's framework but also introduces RoIAlign (Region of Interest Align) to better preserve spatial information by eliminating quantization errors in the pooling process.

The first milestone of this project is to implement region proposal network (RPN) using subset coco dataset we convert them into ground truths and validate it by visualizing it. Next, the model will be fine-tuned by unfreezing ResNet50 [4] layers and training on a target dataset, saving checkpoints to track feature evolution and model performance over epochs. Intermediate evaluations will be done using metrics like mAP and IoU to assess detection and segmentation accuracy. The project will conclude with a final model evaluation, code submission, and presentation, highlighting improvements achieved through visual diagnostics.

2. BACKGROUND

2.1. **R-CNN.** R-CNN [2] (Region-based Convolutional Neural Network), proposed by Girshick et al., introduced a significant shift in object detection by combining region proposals with deep convolutional neural networks (CNNs). The approach involves generating region proposals for an image, followed by a CNN to extract features from each region. These features are then classified with linear SVMs for object detection. However, R-CNN has limitations, including high computational costs and a slow multi-stage training process. Fast R-CNN builds on this by streamlining R-CNN into a single-stage training process, where a Region of Interest (RoI) pooling layer enables direct processing of entire images instead of individual region proposals.

2.2. **Fast R-CNN.** While Fast R-CNN [1] that improves efficiency by extracting a single feature map from the entire input image using a CNN backbone and applying Region of Interest (RoI) pooling to align region proposals with the feature map. It outputs both class probabilities and bounding box regression offsets for each proposal, enabling

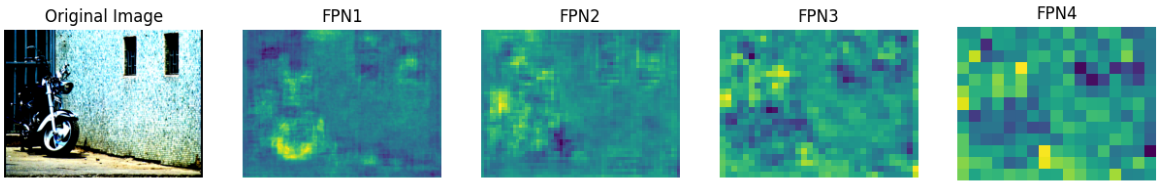


FIGURE 1. Pretrained Resnet50 FPN features

simultaneous object classification and localization. By sharing computation across proposals and avoiding redundant feature extraction, Fast R-CNN achieves faster training and inference compared to its predecessor, R-CNN.

These advances—RPN in Faster R-CNN and the mask branch in Mask R-CNN—enhanced both the speed and accuracy of object detection and segmentation, providing a comprehensive framework for complex vision tasks. Mask R-CNN has become the standard for instance segmentation across diverse applications, from autonomous driving to medical imaging, where precise object boundaries are crucial. Its success underscores the importance of seamless integration of region proposal generation and multi-task learning for complex visual tasks, setting a foundation for further research and applications in object detection and segmentation.

3. DATASET OVERVIEW

The dataset consists of RGB images resized to 800x1088 pixels, annotated with bounding boxes and masks for three categories: vehicles, people, and animals. Images are normalized and padded uniformly across all instances. The data is preprocessed to ensure consistency in scale and format for all images, applying rescaling and zero-padding. Distribution among the classes will be examined to establish a baseline for balanced model training and to assess class-specific performance.

4. CURRENT EXPERIMENTS AND RESULTS

4.1. **Region Proposal network.** A RPN [5] is a critical component in object detection frameworks like Faster R-CNN, designed to efficiently propose candidate regions that likely contain objects. The RPN operates by sliding a small network over a convolutional feature map extracted from an input image. At each position in the feature map, the RPN predicts a set of bounding boxes (proposals) with associated scores indicating the likelihood of containing objects.

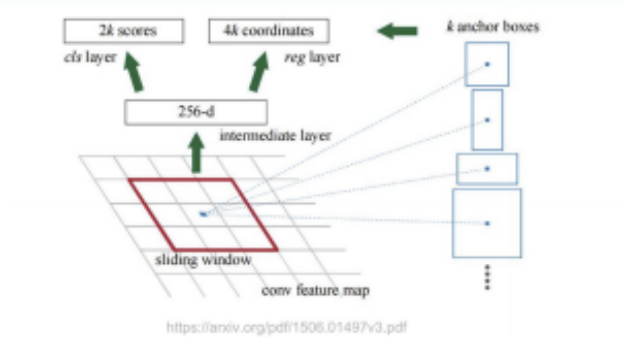


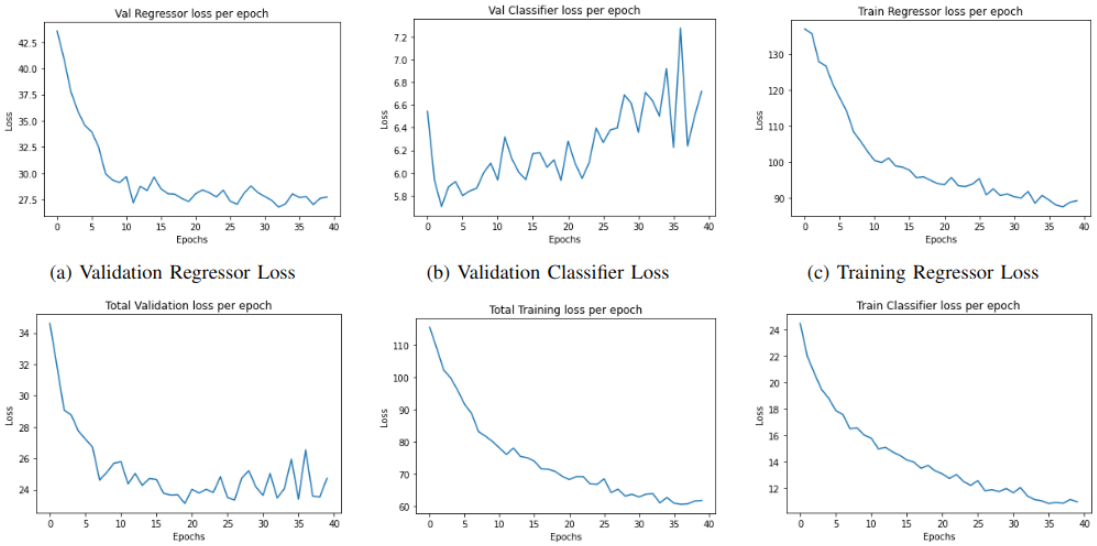
FIGURE 2. detailed RPN architecture [5]

Input are the features from the backbone which are usually VGGNet-16 or ResNet. In our experiment we have used ResNet-50 [4]. During **training**, the RPN assigns ground truth labels to anchor boxes based on their **Intersection over Union (IoU)** with actual bounding boxes in the image. An anchor is classified as a positive sample if its IoU with any ground truth box exceeds a predefined threshold (e.g., 0.7) or if it has the highest IoU with a ground truth box, even if the IoU is below the threshold. Anchors with IoU below a lower threshold (e.g., 0.3) are labeled as negatives (background). A unique sampling strategy is employed to maintain a balanced ratio of positive to negative samples in

each training batch, typically 1:1. This balance is crucial as the majority of anchors may not overlap significantly with objects, leading to an overwhelming number of negative samples.

The **loss function** used in training combines two components: Binary Cross-Entropy Loss (BCE)for the classification head and Smooth L1 Loss for the regression head. The BCE loss measures the accuracy of the objectness predictions, while the Smooth L1 Loss penalizes deviations in the predicted coordinates (x, y, w, h) from the ground truth.

In the **post-processing** stage, the RPN outputs a large number of region proposals. To reduce redundancy and select the most relevant proposals, **Non-Maximum Suppression (NMS)** is applied. NMS removes overlapping proposals with lower objectness scores, retaining only the top k proposals (e.g., 200) based on confidence scores. This refined set of proposals is passed to subsequent stages, such as RoI pooling and classification, for final object detection and localization. This approach ensures computational efficiency and high-quality region proposals.



(a) Validation Regressor Loss (b) Validation Classifier Loss (c) Training Regressor Loss

(d) Total Validation Loss (e) Total Training Loss (f) Training Classifier Loss

FIGURE 3. Loss curves for training and validation phases over epochs (RPN).

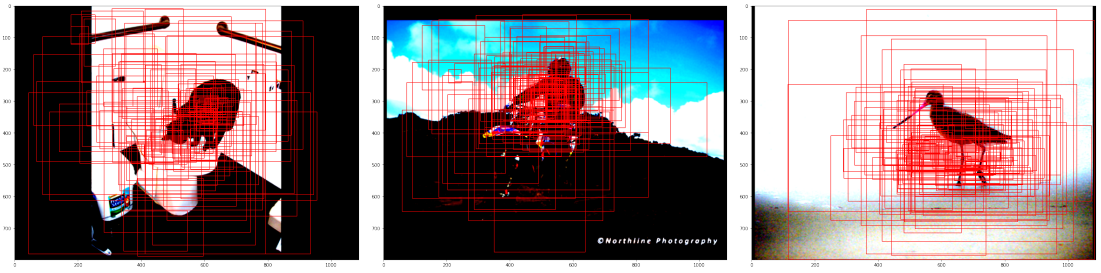


FIGURE 4. Region Proposal Network results visualized with anchor boxes.

4.2. **Faster-RCNN.** The object detection pipeline integrates outputs from a **backbone network**, a **RPN**, and a final **detection head**. Specifically, the backbone extracts features from the input image, which are then passed to the RPN to generate region proposals. These proposals, represented as bounding boxes, are refined and sent to the detection head for classification and bounding box regression. The final detection head outputs bounding boxes (x, y, w, h) and their associated class predictions, where the classes include $C + 1$ categories (the additional category being the background class).

To align the region proposals with features from the backbone, the pipeline uses **ROIAlign** instead of ROIPooling. The proposals generated by the RPN often do not have integer coordinates, and rounding them can introduce misalignments. **ROIAlign** addresses this issue by using bilinear interpolation and smart indexing to accurately locate the

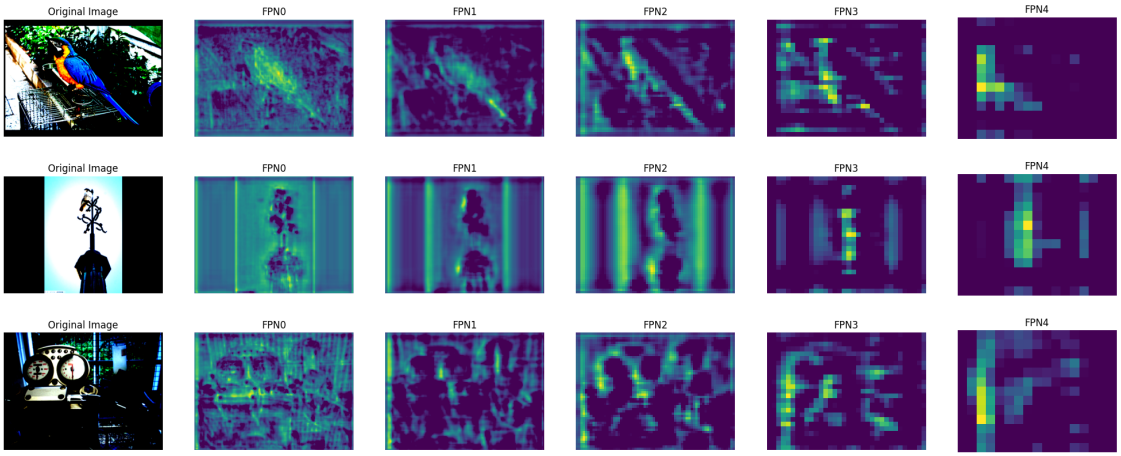


FIGURE 5. RPN intermediate feature visualization.

fractional bounding box positions on the feature pyramid. This ensures that the extracted features align perfectly with the proposed bounding boxes, resulting in improved accuracy in the classification and regression tasks.

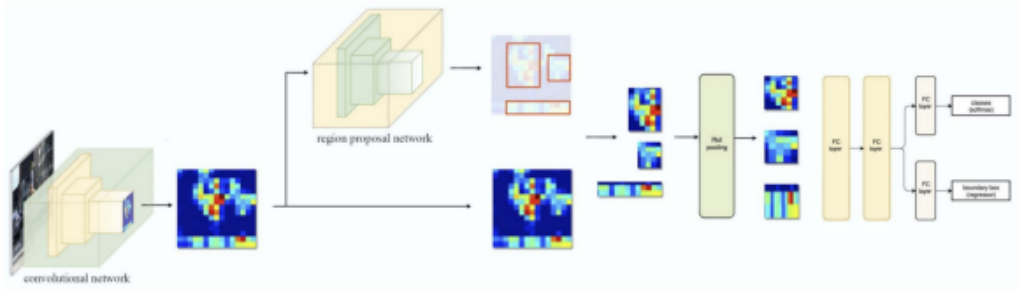


FIGURE 6. Faster-RCNN Architecture [5]

The **loss function** for training combines two components: **Cross-Entropy Loss (CE)** for the classification task and **Smooth L1 Loss** for bounding box regression. The CE loss ensures accurate class predictions for each proposal, while the Smooth L1 Loss minimizes the difference between the predicted and ground truth bounding box coordinates, making it robust to outliers. I have trained the network for 40 epochs with MultiStep learning rate scheduler with intervals of 22,28 and 35. The initial learning rate is 0.0005 with weight decay of 10^{-4}. The optimizer used is SGD with momentum of 0.9.

For further downstream tasks, such as instance segmentation, the top 100 proposal boxes are selected using NMS. These top proposals, ranked by confidence scores, are passed to the segmentation head (**ToMask**) to generate pixel-level masks for the detected objects. The average MAP achieved is 0.62 for IoU threshold of 0.5.

4.3. **Mask R-CNN.** It is an extension of Faster R-CNN, designed to perform instance segmentation by predicting pixel-level masks for detected objects in addition to bounding boxes and class labels. The process begins by selecting the top 100 proposal boxes from the detection head. These proposals, ranked by confidence scores, are refined further.

For training the segmentation head, **binary masks** are used as ground truth for each detected object. These masks represent whether each pixel within a bounding box belongs to the object or not. The segmentation head outputs masks of fixed size $(C, 28, 28)$, where C is the number of object classes. Post-processing involves resizing these predicted masks to match the size of the ground truth bounding boxes using interpolation. This ensures consistency between the predicted masks and the ground truth.

The **loss function** for the segmentation head is computed using **Binary Cross Entropy (BCE) loss** applied pixel-wise. This loss measures the similarity between the predicted mask and the ground truth mask for each pixel. It is

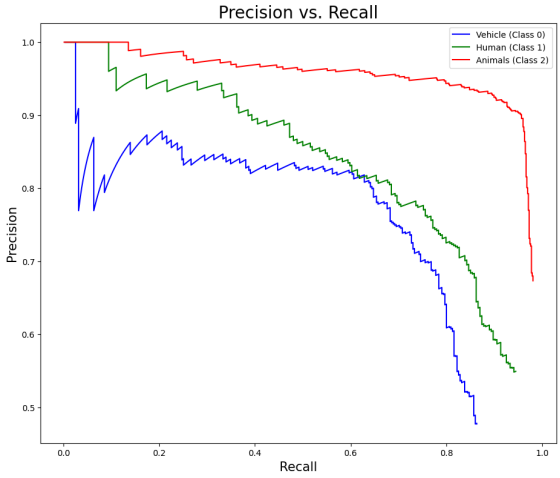
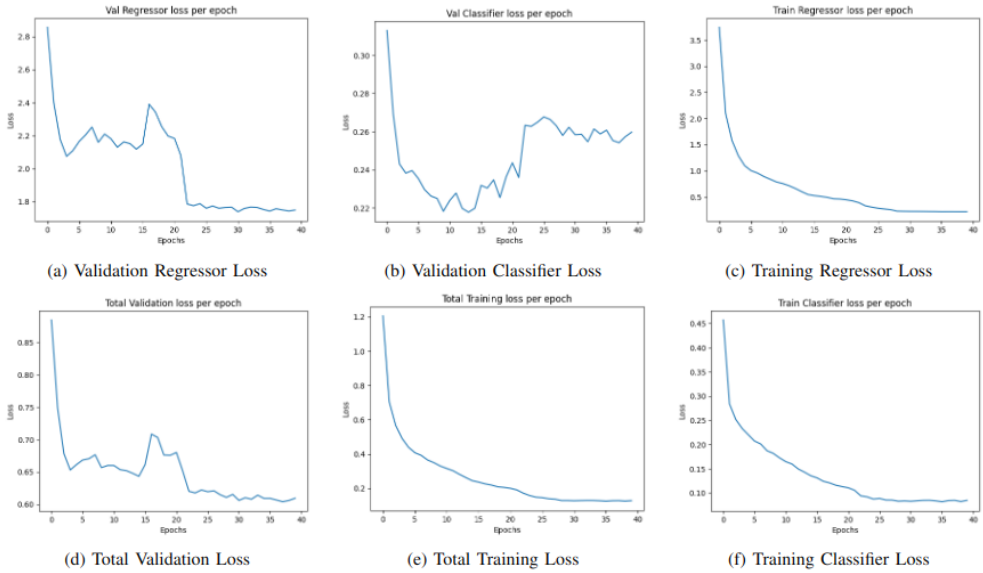


FIGURE 7. Percision vs Recall curve for 3 classes (Faster RCNN).



(a) Validation Regressor Loss (b) Validation Classifier Loss (c) Training Regressor Loss

(d) Total Validation Loss (e) Total Training Loss (f) Training Classifier Loss

FIGURE 8. Loss curves for training and validation phases over epochs (Faster RCNN).

computed independently for each mask and summed over all pixels, ensuring the model learns to produce accurate pixel-level predictions for each object instance.

During **training**, the backbone network, the Region Proposal Network (RPN), and the detection head (box head) are kept **frozen** and run for 45 epochs. This allows the segmentation head to learn without affecting the pre-trained features used for proposal generation and classification. The training process uses a learning rate of 0.001 and MultiStep learning rate rescheduler as discussed above.

5. CONCLUSION

In this project, we successfully implemented and analyzed Faster R-CNN and Mask R-CNN architectures for object detection and instance segmentation tasks. Our experiments demonstrated that Faster R-CNN provides robust object detection capabilities with high accuracy, as evidenced by consistently low validation loss and effective bounding box predictions on the target dataset. The addition of the segmentation branch in Mask R-CNN further enhanced the functionality, enabling precise instance-level segmentation.

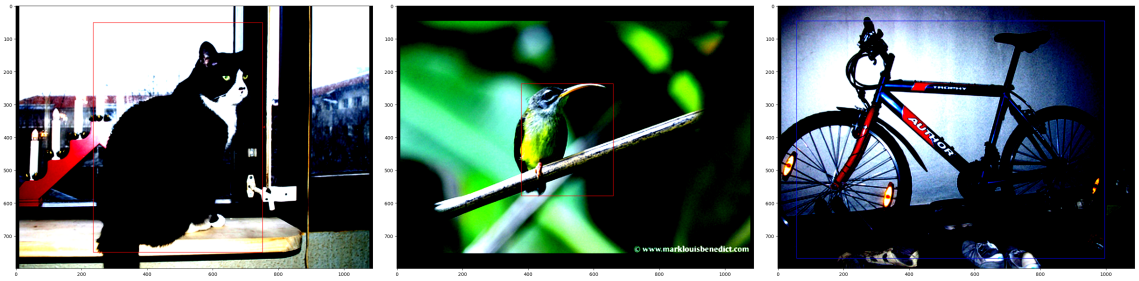
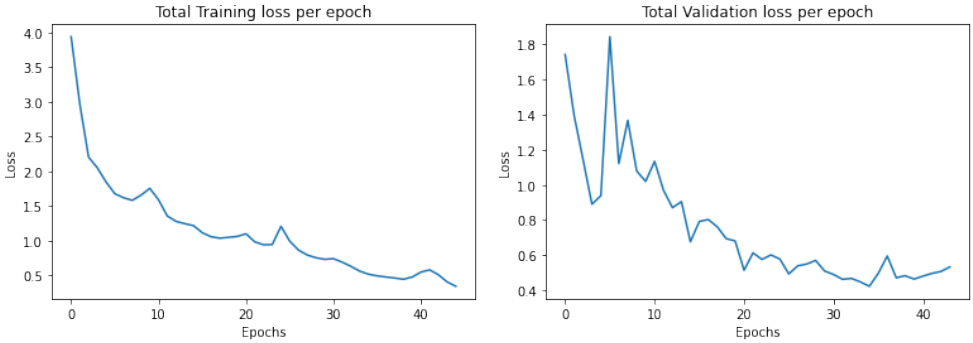


FIGURE 9. Visualizations of predicted bounding boxes using Faster-RCNN.



(A) Training Loss per Epoch (B) Validation Loss per Epoch

FIGURE 10. Training and Validation Loss Curves for Mask R-CNN.

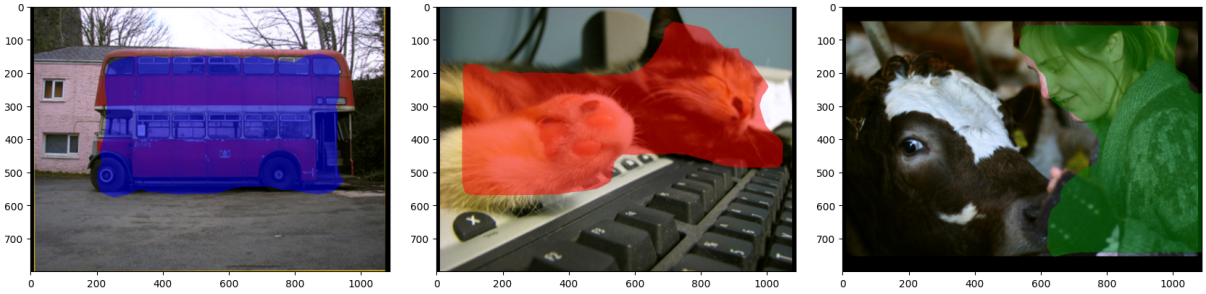


FIGURE 11. Instance segmentation masks generated for different objects using Mask RCNN.

The results show that both models achieve satisfactory performance on a diverse dataset with varying object categories, even under challenging conditions such as overlapping objects and occlusions. By fine-tuning the ResNet50 backbone with Feature Pyramid Networks (FPN), we observed improved detection and segmentation accuracy, highlighting the importance of multi-scale feature extraction. The evaluation metrics, including mAP and IoU, confirmed the effectiveness of the models in balancing accuracy and computational efficiency.

Future Work: Despite the promising results, there are several areas for potential improvement. First, exploring alternative backbones such as ResNet101 or EfficientNet could further enhance the performance, especially for larger and more complex datasets. Second, incorporating techniques like data augmentation or semi-supervised learning could improve generalization across unseen data.

In conclusion, my approach demonstrated the efficacy of combining Faster R-CNN and Mask R-CNN for high-accuracy object detection and instance segmentation. These findings pave the way for deploying such architectures in practical applications like autonomous driving, medical imaging, and robotics, where precise and reliable object detection and segmentation are critical.

5.1. **Code:** https://github.com/jayaramakarthikeya/MaskRCNN

REFERENCES

[1] Ross Girshick. Fast r-cnn, 2015.
[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.
[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.
[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.