Enhanced Object Detection and Instance Segmentation: A Project on Mask R-CNN

1st Aksheetha Muthunooru
Department of CSEE
University of Maryland, Baltimore County
me52139@umbc.edu

2nd Surjodeep Sarkar
Department of CSEE
University of Maryland, Baltimore County
ssarkar1@umbc.edu

***Abstract*—Introducing our framework, we propose a streamlined and versatile solution for object instance segmentation. Our method efficiently identifies objects within an image while simultaneously generating high-quality segmentation masks for each instance. Named Mask R-CNN, our approach builds upon the foundation of Faster R-CNN by incorporating an additional branch dedicated to predicting object masks alongside the existing branch for bounding box recognition. Remarkably, Mask R-CNN is easy to train and introduces minimal overhead to Faster R-CNN. Furthermore, Mask R-CNN readily adapts to various tasks, allowing for the estimation of human poses within the same framework. Using COCO dataset, our results are consistent across all three tracks, including instance segmentation, bounding box object detection, and person key-point detection with our sample images.**

I. Introduction

Mask R-CNN [1] is an algorithm crafted to address the challenging task of instance segmentation, a process that encompasses object detection and meticulous pixel-level segmentation. It expands upon the groundwork established by the Faster R-CNN [2] framework, incorporating a mask prediction function that fabricates exact segmentation masks for each unique object.

The algorithm operates in two crucial stages. The initial stage utilizes a Region Proposal Network (RPN)[1] to generate plausible regions of interest (RoIs) in the image, proposing candidate bounding boxes that might contain objects. These RoIs act as inputs for the ensuing stage.

In the secondary stage, the RoIs [3] are processed in parallel to achieve three primary goals: object class prediction, refinement of bounding box coordinates, and generation of a binary mask for each RoI. This procedure takes advantage of convolutional networks [4] and fully connected layers. The mask technique emphasizes predicting pixel-level masks that precisely demarcate the boundaries of each object.

During the training phase, annotated data is indispensable, offering ground truth information such as object classes, bounding box coordinates, and segmentation masks. The network is trained employing a blend of loss functions [5], including classification loss [6], bounding box regression loss, and mask loss. This training method fosters accurate object classification, precise localization, and detailed segmentation.

One of the remarkable attributes of Mask R-CNN is its versatility and flexibility. Besides its application in instance segmentation, the algorithm can be leveraged for various other tasks, like human pose estimation. For this function, the algorithm treats each keypoint as an independent object and represents it using a distinct one-hot binary mask.

Mask R-CNN has delivered extraordinary performance on widely accepted benchmark datasets such as COCO [7], surpassing preceding methods regarding both accuracy and computational efficiency. Its intuitive architecture, straightforward training procedure, and capability to generate high-quality segmentation masks have positioned it as a leading algorithm in the field of computer vision.

Instance segmentation [8] carries great significance in the field of computer vision, offering numerous benefits and applications. It offers a comprehensive understanding of images by accurately segmenting and differentiating individual object instances, thereby allowing for precise localization and boundary delineation. This is pivotal for object recognition, tracking, and scene comprehension. Moreover, instance segmentation plays an essential role in augmented reality and virtual reality, facilitating realistic virtual object placement and seamless amalgamation with the real world. In the context of image editing and manipulation, it enables users to modify or remove specific objects while maintaining the overall scene. Furthermore, instance segmentation supports object-level analysis, measurements, and dataset annotation, serving as a fundamental instrument for training and evaluating computer vision models. Its capacity to extract detailed information and enable sophisticated tasks makes instance segmentation highly valuable and applicable across a wide range of domains and applications.

The execution of Mask R-CNN involves several vital steps. Initially, the data is preprocessed by annotating the ground truth masks and bounding boxes for each instance and augmenting the dataset to enhance its diversity. Following this, the network architecture is structured, which incorporates a backbone network for feature extraction, a Region Proposal Network (RPN) [9] to generate candidate object proposals, and a mask branch for pixel-wise segmentation. The model is subsequently trained by initializing the network with pretrained weights, fine-tuning it on the instance segmentation dataset, and optimizing the loss function. During the inference

[1] https://medium.com/egen/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9

phase, the trained model is deployed to process new images, generating object proposals and subsequently enhancing their accuracy by predicting class labels, bounding boxes, and segmentation masks. Post-processing techniques such as filtering, non-maximum suppression, and result visualization are utilized to enhance the final segmentation output. Throughout the implementation, considerations are given to hyperparameter tuning [10], memory efficiency, and computational performance to achieve precise and efficient instance segmentation using Mask R-CNN.

II. Related Works

A. Semantic Segmentation

Semantic segmentation, a computer vision task, involves partitioning an image into meaningful regions and assigning class labels to individual pixels. By categorizing each pixel into pre-defined classes such as person, car, road, or tree, semantic segmentation offers a comprehensive comprehension of the image's layout. This pixel-level analysis finds diverse applications, including autonomous driving, image editing, augmented reality, and object recognition, among others.

Deep learning architectures, particularly Convolutional Neural Networks (CNNs) [11], are widely employed for semantic segmentation. These architectures leverage the power of deep learning to learn hierarchical features and capture both local and global contextual information.

Popular CNN-based models for semantic segmentation include Fully Convolutional Networks (FCNs) [12], U-Net [9], DeepLab [13], and SegNet [14]. These models typically consist of an encoder network that extracts features from the input image and a decoder network that upsamples the features to the original image resolution while producing pixel-level class predictions.

During training, these models are trained on annotated datasets, where each pixel in the training images is labeled with its corresponding class. The training process involves optimizing the model's parameters to minimize the disparity between predicted class labels and ground truth labels using loss functions like cross-entropy loss or Dice loss.

The notable progress in semantic segmentation owes its credit to the existence of extensive annotated datasets such as Cityscapes[2], PASCAL VOC [14], and ADE20K [15], alongside the evolution of advanced network architectures. These advancements have substantially enhanced the precision of segmenting intricate scenes, thereby unlocking a multitude of applications across diverse domains.

B. R-CNN

The bounding-box object detection approach known as Region-based CNN (R-CNN) [5] focuses on identifying a limited number of candidate object regions for analysis. By evaluating convolutional networks independently on each Region of Interest (RoI), R-CNN achieved notable progress in object detection. Building upon this, RoIPool [15] was introduced, allowing for the examination of RoIs on feature maps. This enhancement led to faster processing speed and improved accuracy.

The introduction of Faster R-CNN [16] further advanced this line of research by incorporating a Region Proposal Network to learn the attention mechanism. This addition enhanced the flexibility and robustness of the framework, accommodating subsequent improvements and positioning Faster R-CNN as the leading framework in various benchmark evaluations.

C. Instance Segmentation

Inspired by the effectiveness of R-CNN, several instance segmentation approaches have been developed, primarily focusing on segment proposals. Earlier methods relied on bottom-up segments, while subsequent works such as DeepMask and others learned to generate segment candidates, which were then classified using Fast R-CNN. However, these methods suffered from slower speed and lower accuracy as segmentation preceded recognition.

In contrast, our method takes a different approach by simultaneously predicting masks and class labels in parallel, offering a simpler and more flexible solution. Another recent approach called "fully convolutional instance segmentation" (FCIS) [17] combined segment proposal and object detection systems. By predicting position-sensitive output channels, FCIS addressed object classes, boxes, and masks in a fully convolutional manner, resulting in faster processing. However, FCIS encountered challenges with overlapping instances and produced spurious edges, highlighting the inherent difficulties of instance segmentation.

Based on the success of semantic segmentation, another family of instance segmentation techniques has been developed. These techniques start with the per-pixel classification outcomes from models like FCN and make an effort to differentiate between instances of pixels belonging to the same category. Mask R-CNN adopts an instance-first strategy as opposed to these approaches' segmentation-first approach. Future studies are anticipated to investigate a deeper integration of both approaches in order to advance instance segmentation.

III. Mask R-CNN

As depicted in figure 2, the execution of Mask R-CNN involves a sequence of critical components and steps. Initially, a deep convolutional neural network, often pre-trained on extensive image classification tasks such as ImageNet[3], serves as the backbone, facilitating the extraction of informative features from the input image. Subsequently, a Region Proposal Network (RPN) is utilized to produce prospective object areas by predicting bounding box proposals and objectness scores, derived from the feature maps of the backbone network.

A notable advancement in Mask R-CNN is the incorporation of the RoI Align operation. This resolves the inherent misalignment problem found in previous RoI pooling techniques, ensuring precise pixel-level alignment between input feature

[2] https://www.cityscapes-dataset.com/

[3] https://www.image-net.org/

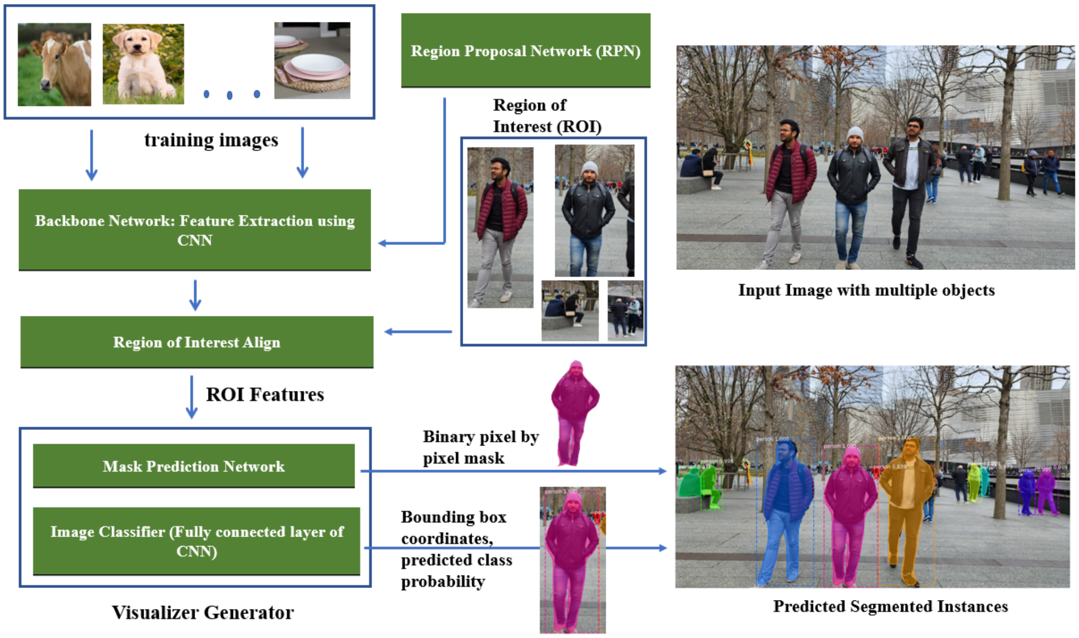
training images
Region Proposal Network (RPN)
Region of Interest (ROI)
Input Image with multiple objects
Backbone Network: Feature Extraction using CNN
Region of Interest Align
ROI Features
Binary pixel by pixel mask
Mask Prediction Network
Image Classifier (Fully connected layer of CNN)
Bounding box coordinates, predicted class probability
Visualizer Generator
Predicted Segmented Instances

Fig. 1. Overview of Instance Segmentation

maps and output regions of interest. The essence of Mask R-CNN resides in its mask branch—a fully convolutional network—that takes each proposed region of interest and predicts a binary mask for every class, facilitating accurate instance segmentation. The model's training follows a supervised learning approach, leveraging annotated datasets wherein each instance comes labeled with its class, bounding box, and segmentation mask.

The training procedure involves refining a combined loss function that includes classification loss, bounding box regression loss, and mask loss. During the inference stage, the trained model is applied to classify suggested regions and fabricate masks for each instance within unseen images. Further considerations for implementation encompass anchor boxes for the Region Proposal Network (RPN), multi-scale training and testing, and the deployment of data augmentation techniques. Collectively, these elements significantly influence the comprehensive performance and robustness of Mask R-CNN.

A. *Two-stage procedure in Mask R-CNN*

The implementation of our Mask R-CNN involves a two-step task comprising the Region Proposal Network (RPN) and the mask prediction branch. In the first stage, the RPN generates candidate object proposals by analyzing feature maps from the backbone network. It slides a small window over the feature maps, predicting the presence of objects and proposing bounding boxes. These proposals serve as potential regions of interest (RoIs) for further processing.

In the second stage, the mask prediction branch is introduced to produce precise instance segmentation masks for each RoI. Operating in parallel with the class prediction and bounding box regression branches, the mask prediction branch utilizes a fully convolutional network (FCN) to predict a binary mask for each class. By taking the RoI as input, the FCN generates a mask output that matches the spatial resolution of the RoI. This enables the mask prediction branch to accurately outline the boundaries of each instance.

The combination of the RPN and the mask prediction branch in Mask R-CNN enables efficient object detection and high-quality instance segmentation simultaneously. The RPN identifies potential object regions, while the mask prediction branch refines the segmentation masks, resulting in precise instance boundaries. This two-stages, contributes to the high performance of Mask R-CNN in our tasks.

The network architecture of Mask R-CNN, as represented in figures 1 and 2, incorporates several vital components. below is a brief introduction of the architectural overview:

Backbone Network: Mask R-CNN utilizes a backbone network, such as ResNet [18] or VGG [19], to derive features from the input image. This backbone network is typically pre-trained on large-scale image classification datasets like ImageNet.

Region Proposal Network (RPN): The RPN takes the features extracted by the backbone network and generates region proposals for potential objects. It predicts the probability of an object's presence and proposes bounding boxes that closely envelop these objects.

Region of Interest (RoI) Align: The RoI Align layer harvests fixed-size feature maps from the backbone network for each region proposal generated by the RPN. This layer ensures precise pixel-level alignment between the input image and the region proposals, a critical aspect for exact mask prediction.

Mask Head: The mask head is tasked with predicting the

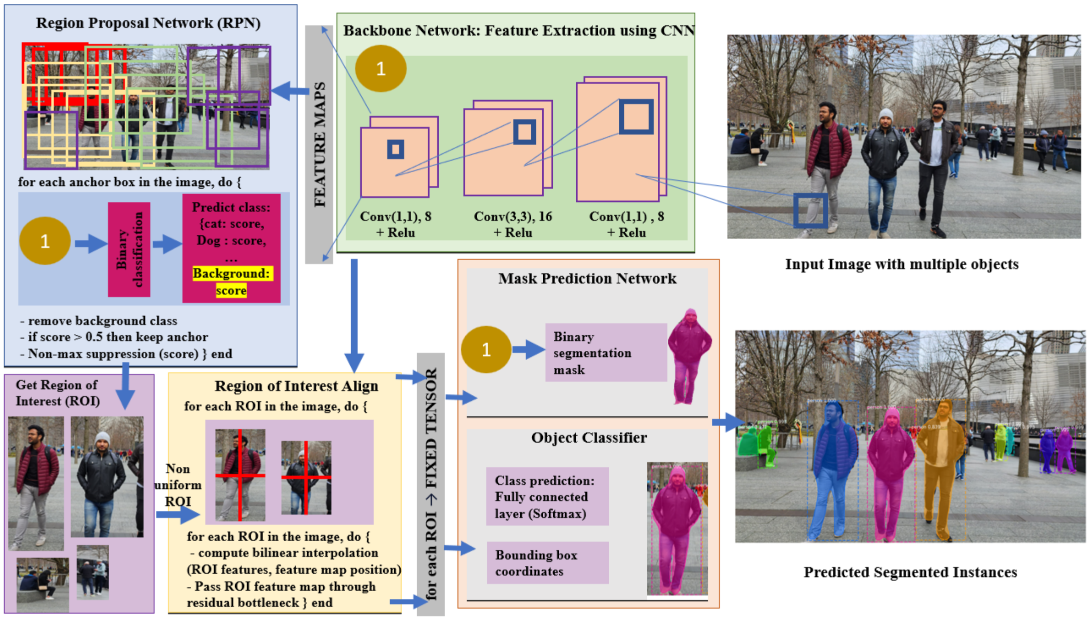
Region Proposal Network (RPN)
for each anchor box in the image, do {
1
Binary classification
Predict class:
{cat: score,
Dog : score,
...
Background: score
- remove background class
- if score > 0.5 then keep anchor
- Non-max suppression (score) } end
FEATURE MAPS
Backbone Network: Feature Extraction using CNN
1
Conv(1,1), 8 + Relu
Conv(3,3), 16 + Relu
Conv(1,1) , 8 + Relu
Input Image with multiple objects
Mask Prediction Network
1
Binary segmentation mask
Object Classifier
Class prediction: Fully connected layer (Softmax)
Bounding box coordinates
Predicted Segmented Instances
Get Region of Interest (ROI)
Non uniform ROI
Region of Interest Align
for each ROI in the image, do {
for each ROI in the image, do {
- compute bilinear interpolation
(ROI features, feature map position)
- Pass ROI feature map through
residual bottleneck } end
for each ROI → FIXED TENSOR

Fig. 2. Illustration of Mask R-CNN

instance segmentation masks for each region proposal. It utilizes the region proposal feature maps from the RoI Align layer and applies a compact fully convolutional network (FCN) [20], [21] to produce a binary mask for each object class. The output masks maintain the same spatial resolution as the region proposals.

Classification and Bounding Box Regression: Besides mask prediction, Mask R-CNN also conducts classification and bounding box regression [12]. The region proposal feature maps are channeled into separate branches for object category classification and bounding box coordinates refinement.

The comprehensive architecture of Mask R-CNN amalgamates these components to achieve precise instance segmentation. It proficiently detects objects, proposes region candidates, and fabricates high-quality segmentation masks for each instance. By extending the Faster R-CNN [2] framework with the mask prediction branch, Mask R-CNN attains superior performance in instance segmentation tasks.

B. Preprocessing and Hyperparameter Settings

Preprocessing steps, data augmentation techniques, and hyperparameter settings play crucial roles in the training and performance of Mask R-CNN. Here is a discussion of some common practices:

- Preprocessing steps, data augmentation techniques, and hyperparameter settings play crucial roles in the training and performance of Mask R-CNN. Here is a discussion of some common practices:
- Data Augmentation: Data augmentation is employed to increase the diversity of the training data and improve the model's generalization. Common augmentation techniques include random horizontal flipping, random scaling, random cropping, rotation, and color jittering. These transformations help the model learn to be robust to variations in object appearance, orientation, and position.
- Anchor Scales and Ratios: Anchor Scales and Ratios play a crucial role in Mask R-CNN, as they enable the model to capture objects with diverse sizes and shapes. By employing anchor boxes of different scales and aspect ratios, the algorithm can effectively handle objects of various dimensions. Selecting appropriate anchor scales and ratios is an essential hyperparameter configuration. Typically, the anchors are carefully designed to encompass a range of object sizes that exist within the dataset.
- Learning Rate and Optimizer: The learning rate determines the step size during model optimization. It is typically set initially to a relatively high value and then reduced over the course of training. Popular learning rate scheduling techniques include step decay, exponential decay, or cyclic learning rates. The choice of optimizer, such as stochastic gradient descent (SGD) or Adam [22], also impacts training dynamics and model performance.
- Loss Function: The loss function used in Mask R-CNN consists of several components, including the classification loss, bounding box regression loss, and mask segmentation loss. These losses are combined and weighted to form the overall loss. The weights assigned to each loss term can be adjusted to balance their contributions during training.
- Regularization: Regularization techniques like weight decay or L2 regularization [23]are often applied to prevent overfitting and encourage the model to learn more generalizable features.
- Hyperparameter Tuning: Hyperparameter Tuning is a crucial aspect of optimizing the performance of Mask R-CNN. Parameters such as the number of feature maps,

convolutional filter sizes, layer configurations, and RoI pooling sizes have a substantial impact on the model's effectiveness. Tuning these hyperparameters involves iterative experimentation and validation using a separate dataset. It's worth mentioning that the choice of preprocessing steps, data augmentation techniques, and hyperparameter settings can vary depending on the dataset, domain, and specific implementation of Mask R-CNN.
It's important to note that the specific preprocessing steps, data augmentation techniques, and hyperparameter settings may vary depending on the dataset, domain, and specific implementation of Mask R-CNN. Experimentation and fine-tuning are often required to find the optimal configuration for a given task.

IV. Experimental Setup

Our implementation of Mask R-CNN is based on Python3, utilizing Tensorflow and Keras, with ResNet101 serving as the backbone network. We've employed the MS COCO (Microsoft Common Objects in Context) dataset [7], a benchmark dataset broadly used in the domain of computer vision. It's structured to facilitate tasks like object detection and instance segmentation. Comprising an extensive array of 320,000 images spanning 80 distinct object categories, the COCO dataset includes a variety of common objects, from cars and people to animals and household items.

Every image within the COCO dataset comes annotated with details such as object bounding boxes, category labels for each object, and segmentation masks for every individual instance of the object.

V. Analysis of Algorithms

In this section, we are going to describe about the algorithms implemented in this project.

Algorithm 1 Mask R-CNN

```
Backbone network: backbone_network ← load_backbone_network()
Region Proposal Network (RPN): rpn_network ← load_rpn_network()
Classification and mask prediction network: classification_network ← load_classification_network()
mask_network ← load_mask_network()
for epoch ← 1 to num_epochs do
  for each images, annotations in training_data do
    Forward pass:
    features ← backbone_network(images)
    rpn_regions ← rpn_network(features)
    roi_features ← roi_align(features, rpn_regions)
    class_probs, bbox_deltas ← classification_network(roi_features)
    masks ← mask_network(roi_features)
    Calculate losses:
    class_loss ← calculate_classification_loss
(class_probs, annotations) bbox_loss ← calculate_bbox_regression_loss(bbox_deltas, annotations)
    mask_loss ← calculate_mask_loss(masks, annotations)
    total_loss ← class_loss + bbox_loss + mask_loss
    Backward pass and update weights:
    total_loss.backward()
    optimizer.step()
    optimizer.zero_grad()
function PREDICT_MASKS(image)
  features ← backbone_network(image)
  rpn_regions ← rpn_network(features)
  roi_features ← roi_align(features, rpn_regions)
  class_probs, bbox_deltas ← classification_network(roi_features)
  masks ← mask_network(roi_features)
  detections ← apply_nms(class_probs, bbox_deltas, masks)
  return detections
=0
```

- The methodology employed in Mask R-CNN is a fusion of various techniques. It harnesses the power of convolutional neural networks (CNNs) for extracting features, utilizes region proposal networks (RPNs) for creating potential object suggestions, and employs fully connected networks (FCNs) for tasks such as classification, bounding box regression, and mask prediction.
- Moreover, Mask R-CNN employs RoI Align, a more accurate region of interest pooling approach in comparison to conventional RoI pooling. This aligns features with a consistent grid, facilitating the prediction of precise pixel-level segmentation masks. Consequently, the algorithm harnessed in Mask R-CNN melds these techniques, amalgamating the strengths of CNNs, RPNs, FCNs, and RoI Align to deliver robust capabilities in object detection and instance segmentation.
- The loss function in Mask R-CNN comprises three elements: classification loss, bounding box regression loss, and mask segmentation loss. Each component plays a role in the cumulative loss, which is aimed to be minimized during the training process. Below is a detailed explanation of the loss functions utilized in Mask R-CNN:
    - Classification Loss: This gauges the divergence between the predicted class probabilities and the actual class labels for each proposed object. The softmax cross-entropy loss is typically the preferred loss function for classification tasks.
      Bounding Box Regression Loss: This loss quantifies the disparity between the predicted bounding box coordinates (for instance, the offsets for the top-left and bottom-right corners) and the true bounding box coordinates. The smooth L1 loss is generally employed as the loss function for regression tasks, given its lower sensitivity to outliers compared to the conventional L2 loss.
    - Mask Segmentation Loss: The mask segmentation loss measures the similarity between the predicted instance

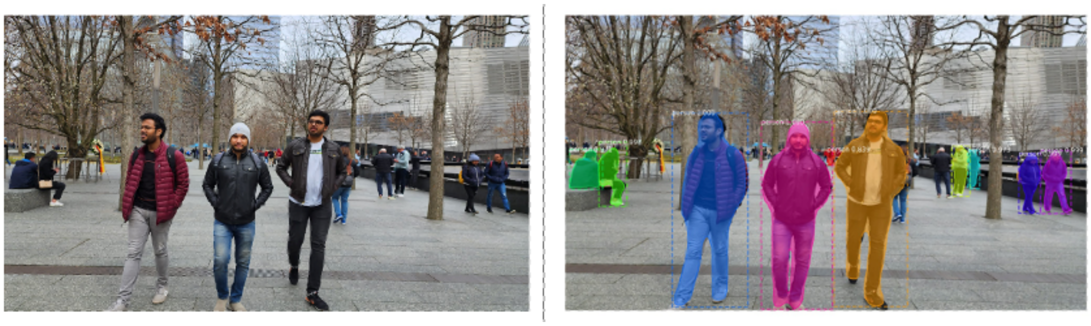

Fig. 3. Results of Mask R-CNN

masks and the ground truth masks. It is typically computed using the binary cross-entropy loss, treating the mask prediction as a binary pixel-wise classification problem (foreground vs. background) for each object.

- The individual losses are computed for each object proposal, and the total loss is the sum of these component losses, possibly with different weighting factors for balancing their contributions. It's important to note that the specific implementations of these loss functions may vary depending on the framework or library used for Mask R-CNN. Different variations or optimizations may also be employed, such as focal loss for handling class imbalance or pixel-wise loss functions for mask segmentation.
- Within the framework of the Mask R-CNN algorithm, "apply NMS" signifies the implementation of the non-maximum suppression (NMS)[4] technique. NMS is a post-processing method employed to discard redundant or repetitive object detections, keeping only those that are the most accurate and non-overlapping.

  Upon completion of the classification, bounding box regression, and mask prediction phases, Mask R-CNN yields a collection of object detections, each associated with bounding boxes, class probabilities, and segmentation masks. Nonetheless, it's possible that multiple detections pertain to the same object or that overlapping detections account for identical regions.
- To address this, NMS is applied to filter out redundant detections and select the most reliable ones. The process typically involves the following steps:
- Arrange the detections in order based on their respective class probabilities. Begin with the detection that possesses the maximum class probability, and regard it as a chosen detection. Evaluate the remaining detections in comparison to the chosen one, and compute their intersection-over-union (IoU)[5] values relative to this selected detection.
- Discard any detections that have a high IoU (above a threshold) with the selected detection, as they are considered redundant. Repeat steps 3 and 4 until all detections have been considered. Retain the selected detections that survived the NMS process.
- By applying NMS, redundant detections that overlap significantly are removed, and only the most confident and non-overlapping detections are kept. This helps ensure that each object is represented by a single detection and reduces the number of false positives. In the context of Mask R-CNN, this step is the implementation of this NMS process to filter the object detections based on their class probabilities, bounding box coordinates, and segmentation masks. The resulting detections are the final output of the Mask R-CNN algorithm.

VI. Results

The figure 3 above shows our outcome of successful implementation of Mask R-CNN on MS COCO dataset.

VII. Conclusion

In conclusion, implementing Mask R-CNN involves a comprehensive set of steps and considerations. By combining region proposal networks, feature extraction networks, and 7instance segmentation heads, Mask R-CNN enables accurate 1object detection and pixel-level segmentation. The training process involves classification, bounding box regression, and loss functions. During inference, the trained model is applied to unseen images to classify proposed regions and generate masks for each instance. Additionally, factors such as anchor scales and ratios, as well as hyperparameter tuning, significantly impact the performance of Mask R-CNN.

VIII. Future Work

Future work for Mask R-CNN involves several key areas for improvement. Firstly, enhancing its efficiency and speed for real-time applications by optimizing the network architecture and proposing more efficient region proposals. Secondly, addressing occlusion and partial visibility to improve its performance in challenging scenarios.

Another direction is expanding its generalization capabilities to handle unseen or novel classes using techniques like few-shot learning, transfer learning, and domain adaptation. Incorporating higher-level semantic understanding and contextual information can further improve accuracy and consistency.

[4] https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c

[5] https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Exploring weakly supervised and self-supervised learning approaches can reduce the reliance on pixel-level annotations, making training more scalable. Integrating Mask R-CNN with 3D perception and exploring applications in robotics, healthcare, surveillance, and industrial automation are also promising avenues.

Lastly, addressing deployment challenges such as model compression, optimization for edge devices, and privacy considerations will facilitate practical adoption. Focusing on these areas will contribute to the advancement of Mask R-CNN, making it more efficient, robust, and applicable in various real-world scenarios.

References

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[2] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[3] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *CVPR*, 2017.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[5] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[8] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multi-cut," in *CVPR*, 2017.

[9] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *CVPR*, 2017.

[10] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, 2015.

[11] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.

[12] S. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *CVPR*, 2017.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[14] P. Arbelaez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.

[16] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *CVPR*, 2015.

[17] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *ECCV*, 2016.

[18] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016.

[21] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *CVPR*, 2015.

[22] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What makes for effective detection proposals?," *PAMI*, 2015.

[23] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *ICCV*, 2017.