

# Instance Segmentation

1<sup>st</sup> Maanas Gaur  
Roll No: 2021537  
IIIT - Okhla, New Delhi  
maanas21537@iiitd.ac.in

2<sup>nd</sup> Avish Dahiya  
Roll No: 2021522  
IIIT - Okhla, New Delhi  
avish21522@iiitd.ac.in

3<sup>rd</sup> Sagar Keim  
Roll No: 2019196  
IIIT - Okhla, New Delhi  
sagar19196@iiitd.ac.in

**Abstract**—Evaluating the Transferability of COCO-Trained Models to Cityscapes for Instance Segmentation

**Index Terms**—component, formatting, style, styling, insert

## I. INTRODUCTION

State-of-the-art instance segmentation models often achieve impressive results on benchmark datasets like COCO (Common Objects in Context) [1]. However, these models are trained on datasets that focus on general object categories, which may not translate in similar manner to real-world scenarios with different visual characteristics and object distributions. This phenomenon, known as domain shift, can lead to performance change when applying a model trained on one domain (COCO) directly to another domain (e.g., urban street scenes in Cityscapes) [2].

### A. Motivation

This project aims to evaluate the transferability of COCO-trained models for instance segmentation on the Cityscapes dataset. By investigating the impact of domain shift on model performance, we can gain valuable insights into the limitations of current models and the importance of considering domain-specific training data. Furthermore, we will compare the performance of two popular instance segmentation models, Mask R-CNN and PANet, to assess how their ability to handle domain shift might differ

### B. Importance and Usefulness

This project will contribute to the field by:

- Quantifying the performance change caused by domain shift when using COCO-trained models for urban scene segmentation.
- Highlighting the need for domain adaptation techniques when applying instance segmentation models to specific real-world scenarios.
- Providing valuable insights for researchers developing more robust and transferable instance segmentation models.

The findings of this project can be used to guide the development of future models that are more adaptable to different visual domains and can achieve high accuracy in real-world applications, such as autonomous vehicles and robotics.

## II. LITERATURE REVIEW

### A. Mask RCNN (He et al., 2017)

Mask R-CNN builds upon Faster R-CNN for pixel-level image segmentation [3]. A key innovation is decoupling classification and mask prediction. It adds a branch to the Faster R-CNN architecture specifically for predicting an object mask alongside existing branches for classification and bounding box localization. This mask branch acts as a small fully-connected network applied to each Region of Interest (RoI), generating a segmentation mask in a pixel-to-pixel manner. Recognizing the importance of precise alignment for pixel-

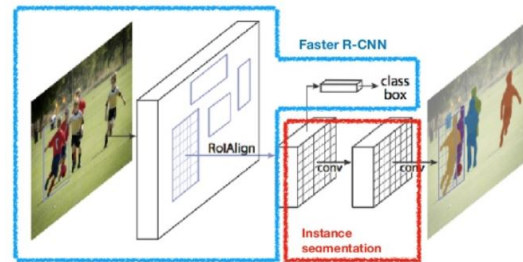


Fig. 1. Mask R-CNN Architecture

level tasks, Mask R-CNN introduces the RoIAlign layer. This refines the RoI pooling layer from Faster R-CNN by addressing quantization issues that could lead to misalignment between extracted features and the original image. RoIAlign achieves this by using bilinear interpolation for floating-point location values within the RoI, ensuring proper feature-to-pixel correspondence.

In summary, Mask R-CNN leverages Faster R-CNN's core architecture while introducing a dedicated mask prediction branch and the RoIAlign layer for precise mask generation. This combination enables Mask R-CNN to excel at instance segmentation tasks.

### B. PointRend (Kirillov et al.2019)

PointRend is a novel module designed to improve the efficiency and accuracy of image segmentation tasks, including instance and semantic segmentation. It approaches segmentation as an "image rendering" problem, aiming to create high-quality label maps with fewer computational resources.

PointRend offers flexibility, seamlessly integrating into existing segmentation architectures like Mask R-CNN. It replaces the standard mask head with its own point-based prediction approach. This strategy significantly reduces computational cost compared to directly predicting labels for every pixel in the image, while potentially achieving higher segmentation accuracy.

**Adaptive Point Selection:** Instead of processing every pixel in the high-resolution output grid, it selectively chooses a small number of real-value points for making predictions. It is done for both inference and training. During inference, it iteratively refines a coarse segmentation by upsampling and then identifying the most uncertain points (often on object boundaries) on the denser grid. These points are chosen for further analysis using their local features extracted from the original feature map. For training, a different random sampling approach is used, prioritizing points with uncertain coarse predictions while also maintaining some even coverage across the image.

**Point-wise Feature Representation:** Once the points are chosen, PointRend extracts features specifically for those points. It achieves this by performing a process called bilinear interpolation. It creates a new feature vector by blending nearby grid point values which helps in gathering a unique feature description for every selected point.

**Point Head Subnetwork:** It's a lightweight neural network specifically designed for assigning labels to each selected point. It uses the feature vector from the previous step to predict the label for each point. .

### III. DATASETS

#### A. Microsoft COCO: Common Objects in Context

The COCO (Common Objects in Context) dataset [1] consists of 115k images for training and 5k images for validation (new split of 2017). The dataset includes 80 classes with pixel-wise instance mask annotation.

'person'	'bicycle'	'car'	'motorcycle'
'airplane'	'bus'	'train'	'truck'
'boat'	'traffic light'	'fire hydrant'	'stop sign'
'parking meter'	'bench'	'bird'	'cat'
'dog'	'horse'	'sheep'	'cow'
'elephant'	'bear'	'zebra'	'giraffe'
'backpack'	'umbrella'	'handbag'	'tie'
'suitcase'	'frisbee'	'skis'	'snowboard'
'sports ball'	'kite'	'baseball bat'	'baseball glove'
'skateboard'	'surfboard'	'tennis racket'	'bottle'
'wine glass'	'cup'	'fork'	'knife'
'spoon'	'bowl'	'banana'	'apple'
'sandwich'	'orange'	'broccoli'	'carrot'
'hot dog'	'pizza'	'donut'	'cake'
'chair'	'couch'	'potted plant'	'bed'
'dining table'	'toilet'	'tv'	'laptop'
'mouse'	'remote'	'keyboard'	'cell phone'
'microwave'	'oven'	'toaster'	'sink'
'refrigerator'	'book'	'clock'	'vase'
'scissors'	'teddy bear'	'hair drier'	'toothbrush'

TABLE I  
OBJECT CATEGORIES IN THE COCO DATASET

#### B. Cityscapes Dataset Description

The Cityscapes Dataset [2] is a large collection of images in street scenes from 50 different cities. It's designed to help researchers develop computer vision algorithms that can understand the content of urban environments. It has fine annotations for 2975 train, 500 val, and 1525 test images. It includes 20k coarse training images without instance annotations, which are not used in this project. All images in the dataset are 2048×1024 pixels.

The instance segmentation task in the Cityscapes dataset involves 8 object categories, whose numbers of instances on the fine training set are:

person	rider	car	truck	bus	train	motorcycle	bicycle
--------	-------	-----	-------	-----	-------	------------	---------

TABLE II  
OBJECT CATEGORIES IN THE CITYSCAPES DATASET

### IV. MODELS

The backbone used for our model is mask\_rcnn\_R\_50\_FPN\_3x. Here's the breakdown of each term

- **mask\_rcnn:** This indicates that the model is based on the Mask R-CNN architecture, which combines region-based convolutional neural networks (R-CNN) with instance segmentation.
- **R\_50:** This refers to the ResNet-50 backbone network used in the model. ResNet-50 is a variant of the ResNet architecture, which is known for its depth and skip connections that help alleviate the vanishing gradient problem.
- **FPN:** This stands for Feature Pyramid Network, which is a multi-scale feature extraction method that aggregates features from different levels of the network hierarchy to improve object detection and segmentation performance.
- **3x:** This indicates the training schedule, specifically the number of iterations or epochs. In this case, "3x" suggests that the model was trained for three times longer than the default training schedule.

#### A. Loss Function

Mask R-CNN employs a multi-task loss function combining classification loss, bounding box regression loss, and a mask binary cross-entropy loss. This encourages the model to simultaneously optimize these tasks for accurate object detection, localization, and fine-grained segmentation.

The loss function can be expressed as:

$$L = L_{cls} + L_{box} + L_{mask}$$

where:

- $L_{cls}$  is the classification loss,
- $L_{box}$  is the bounding box regression loss, and
- $L_{mask}$  is the mask binary cross-entropy loss.

In PointRend The point-wise loss represents the effectiveness in further refining the mask at selected points within the instance mask. Lower values of this loss ensure more accurate object boundaries, highlighting finer details at the selected points.

The point-wise loss for each selected point  $m$  from instance mask is defined using a loss function such as Smooth L1:

$$L_{\text{point}} = \frac{1}{N} \sum_m \text{smooth}_{L1}(p_m - p_{gt})$$

Where:

- $N$  is the total number of selected points.
- $p_m$  is the refined prediction at point  $m$ , and
- $p_{gt}$  is the ground truth label at point  $m$ .

Smooth L1 is a common choice for bounding box regression tasks because it reduces the influence of outliers while still providing smooth gradients during optimization.

Incorporating this additional term into the overall loss function, the new formulation would be:

$$L = L_{cls} + L_{box} + L_{mask} + L_{point}$$

Each component of the loss function still contributes to the overall objective of training the model, but now there's an added emphasis on refining the mask at selected points within the instance mask, aiming for more accurate object boundaries and finer details.

## V. EXPERIMENT

### Experiment 1: COCO Evaluation (Source Domain)

#### Quantitative:

We utilized the Detectron2 library to load pre-trained models from the model zoo, including Mask R-CNN and Mask R-CNN with PointRend. For quantitative analysis, we calculated various metrics including Average Precision (AP) across different IoU thresholds (AP50, AP75), as well as APs, APm, and API for small, medium, and large objects respectively. By comparing model performances against each other and analyzing individual AP variants, we identified potential struggles with specific object sizes, indicating existing biases within the models.

### Experiment 2: Cityscapes Evaluation (Target Domain)

#### Quantitative:

In this experiment, we evaluated the pre-trained models on the Cityscapes dataset, a different domain from COCO. We calculated AP and AP50 for each class and total AP, and compared these results with those obtained on the COCO dataset. Lower AP values for specific classes in Cityscapes compared to COCO indicated potential biases towards COCO's dominant categories, revealing performance drops due to domain shift and class imbalance.

#### Qualitative:

Visualizations of the segmentation results allowed us to directly observe how well the models localized and segmented objects, including bounding boxes and mask accuracy. This qualitative analysis complemented the quantitative results, helping us identify potential issues such as misidentified objects or inaccurate mask boundaries, and understand how the scale of objects influenced the model's performance.

## VI. RESULT AND ANALYSIS



Fig. 2. COCO Qualitative

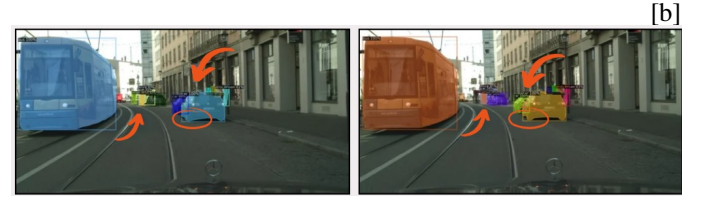


Fig. 3. Cityscapes Qualitative

Fig. 4. Comparison of qualitative results between COCO and Cityscapes datasets.

TABLE III  
COCO EVALUATION RESULTS

Model	Metric	AP	AP50	AP75	APs	APm	API
Mask RCNN	bbox	37.459	54.561	41.855	19.956	40.365	50.871
	segm	34.336	52.787	37.434	15.699	36.583	50.555
Mask RCNN+PointRend	bbox	37.318	54.534	41.471	19.979	40.255	49.893
	segm	35.170	53.001	38.271	16.138	37.441	51.249

#### Maanas Analysis

#### Avish Analysis

The project evaluated how well models trained on the COCO dataset perform when applied to the Cityscapes dataset, particularly investigating the impact of domain shifts on instance segmentation models like Mask R-CNN and PointRend. My analysis of the findings confirms that although these models achieve excellent results on COCO, their performance declines on Cityscapes without specific adaptations to account for different domain characteristics.

*Trends and learnings from model performances::* The comparative analysis, as illustrated in the evaluation results, reveals that while the standard Mask R-CNN and its PointRend-enhanced version perform robustly, PointRend generally improves performance across various metrics. This

TABLE IV  
PERFORMANCE COMPARISON OF MASK R-CNN AND POINTREND

Class	AP	AP <sub>50%</sub>
<b>Mask R-CNN</b>		
Person	25.500	31.874
Rider	1.270	1.587
Car	34.868	43.585
Truck	1.799	2.248
Bus	2.270	2.838
Train	9.869	12.336
Motorcycle	1.193	1.491
Bicycle	3.232	4.040
<b>Average</b>	10.000	12.500
<b>PointRend</b>		
Person	39.219	49.022
Rider	1.953	2.441
Car	53.627	67.034
Truck	2.767	3.457
Bus	3.491	4.365
Train	15.179	18.973
Motorcycle	1.835	2.293
Bicycle	4.971	6.214
<b>Average</b>	15.380	19.225

highlights a significant trend: models enhanced with specific features for addressing segmentation challenges, such as finer detail delineation, can improve outcomes in diverse application scenarios.

*Key insights and overall learning::* This study’s crucial insight is the importance of domain adaptation when deploying pre-trained models across different environments. The research underlines that enhancements in model architecture, such as incorporating PointRend, can alleviate some performance degradation due to domain shifts. This emphasizes the need for adaptive model designs to effectively boost transferability and accuracy.

*Future research directions::* Looking forward, several research pathways appear promising: Enhancing Domain Adaptation Techniques: Developing advanced methods for better-adapting models trained on datasets like COCO for use with other datasets like Cityscapes. Refining Model Architectures: Further improving architectures to enhance their efficiency and effectiveness, particularly for processing high-resolution details. Broadening Model Applications: Extending these models to more varied real-world settings to comprehensively evaluate their limitations and capabilities. Automated Model Adjustments: Innovating within the training process to automatically fine-tune key model parameters to fit specific domain characteristics optimally.

These reflections solidify my understanding of current segmentation model capabilities and limitations and guide potential advancements in adapting deep learning models for varied application domains.

## Sagar Analysis

The investigation delved into how effectively models, originally trained on the COCO dataset, operate when applied to Cityscapes, specifically probing the ramifications of domain shifts on instance segmentation models like Mask R-CNN and PointRend. My analysis confirms that while these models yield exceptional results on COCO, their efficacy diminishes on Cityscapes in the absence of tailored adaptations to accommodate differing domain characteristics.

Insights gleaned from model performances underscore a notable trend: the incorporation of specific features, such as PointRend’s enhancements for finer detail delineation, tends to bolster performance across a spectrum of metrics. The comparative analysis elucidates that while the standard Mask R-CNN and its PointRend-augmented counterpart exhibit robust performance, PointRend consistently enhances outcomes. This underscores the significance of augmenting models with tailored features to tackle segmentation challenges, ultimately fostering improved results across diverse application scenarios.

## REFERENCES

- [1] Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014, May 1). Microsoft COCO: Common Objects in Context. arXiv.org. <https://arxiv.org/abs/1405.0312>
- [2] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016, April 6). The Cityscapes Dataset for Semantic Urban Scene Understanding. arXiv.org. <https://arxiv.org/abs/1604.01685>
- [3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, March 20). Mask R-CNN. arXiv.org. <https://arxiv.org/abs/1703.06870>
- [4] Kirillov, A., Wu, Y., He, K., Girshick, R. (2019, December 17). PointRend: Image segmentation as Rendering. arXiv.org. <https://arxiv.org/abs/1912.08193v2>