Instance Segmentation on the BDD100K Image Dataset

404 Not Found):

Malde Dharmi (MDS202343)

Esha Bhattacharya (MDS202324)

Narendra C (MDS202336)

Siddhesh Maheshwari (MDS202347)

INTRODUCTION

This project addresses the challenges of **object detection and instance segmentation** in urban environments, a critical task for applications such as autonomous driving and traffic management. Using the **BDD100K Image dataset**, which features annotated images across nine driving-related classes, the project aims to develop a model capable of accurately identifying and segmenting objects in complex, real-world driving scenarios.

KEY OBJECTIVES:

- Inspecting how the model performs and interacts with these images to improve its ability to detect and segment relevant objects.
- Fine-tune a **mask r-cnn** architecture and evaluate its interaction with the images for accurate object detection and segmentation.

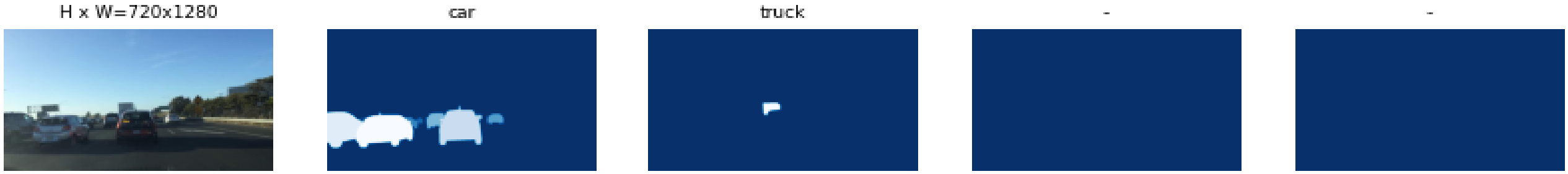
DATASET OVERVIEW

Data Format:

The BDD100K Image dataset is formatted using the json, a widely-used structure for object detection and segmentation tasks.
the original BDD100K dataset contains 100,000 high-definition videos, each video is about 40 seconds. The key frames are sampled at the 10th second of each video to obtain 100,000 images.

Key JSON Fields:

- **Images**: Contains metadata (file name, dimensions) for each image.
- **Categories**: Defines object classes, including id and name.
- **Annotations**: Stores object instance data (bounding boxes, category IDs, attributes etc.).



ANNOTATION EXAMPLE:

Below is the complete JSON annotation from the dataset. The JSON annotation for each image in the dataset contains detailed information about the image's attributes like weather, scene description, time of the day, geometric annotations like box2d(co=ordinates of 4 corners for rectangular objects), poly2D for polygonal areas(a list of co-ordinates, types of segments, whether it's closed or not) etc.


```
{'name': '00091078-84635cf2.jpg',
 'attributes': {'weather': 'snowy',
  'scene': 'city street',
  'timeofday': 'dawn/dusk'},
 'timestamp': 10000,
 'labels': [{'category': 'truck',
   'attributes': {'occluded': True,
    'truncated': True,
    'trafficLightColor': 'none'},
   'manualShape': True,
   'manualAttributes': True,
   'box2d': {'x1': 0, 'y1': 1.468279,
'x2': 356.05738, 'y2': 701.295692},
   'id': 344},
```

```
{'category': 'car',
   'attributes': {'occluded':
False,
    'truncated': True,
    'trafficLightColor': 'none'},
   'manualShape': True,
   'manualAttributes': True,
   'box2d': {'x1': 196.015093,
    'y1': 352.386687,
    'x2': 706.975787,
    'y2': 690.090594},
   'id': 345},
```

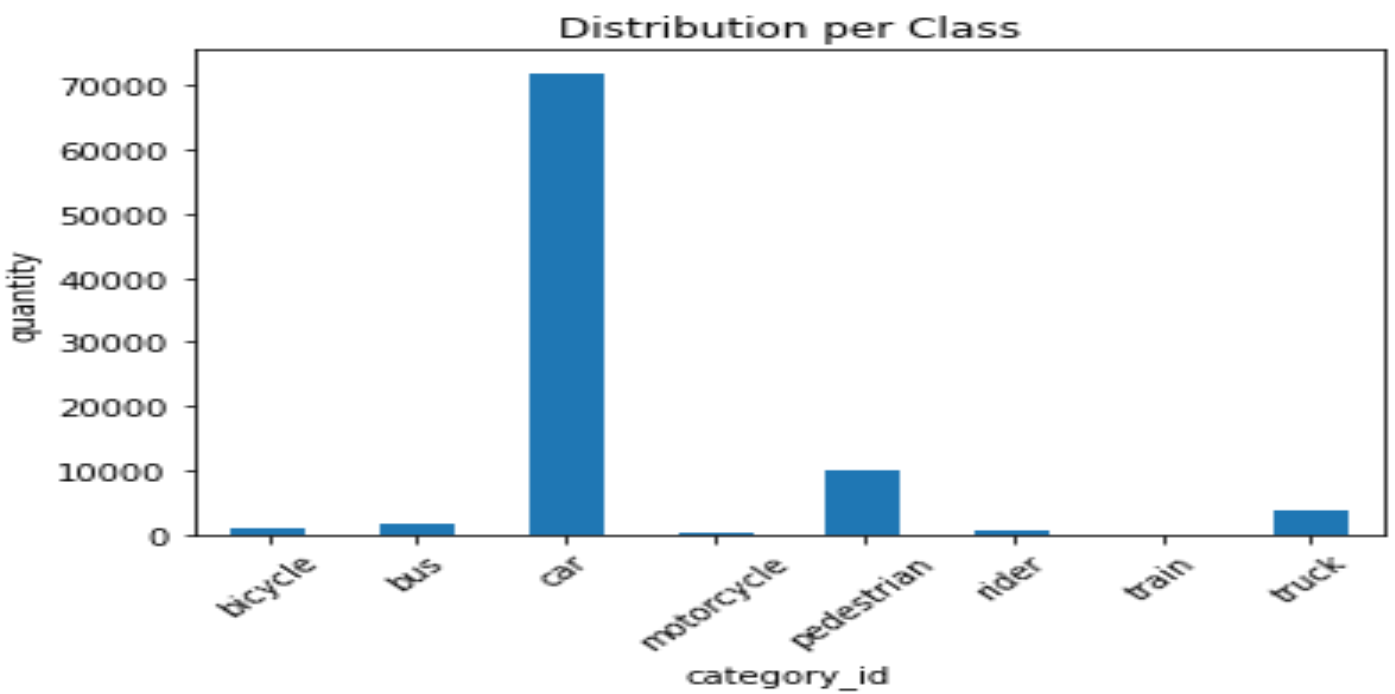
…….....

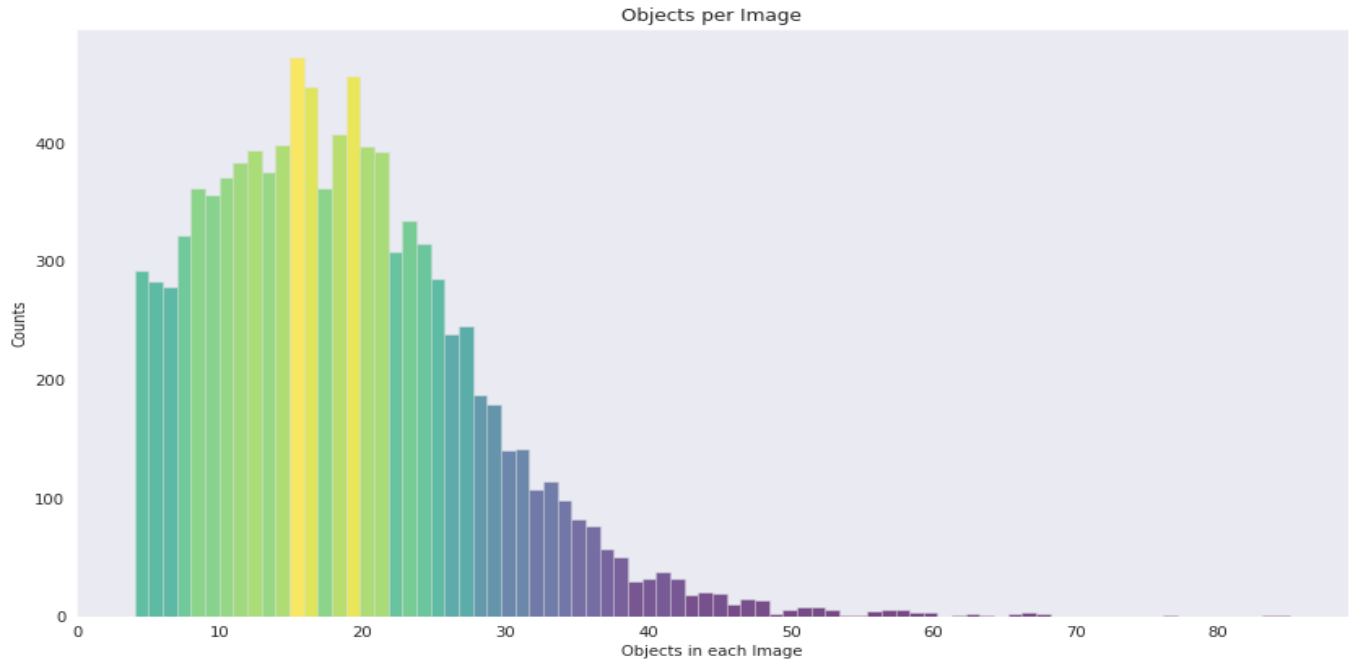
```
{'category': 'pedestrian ',
   'attributes': {'occluded':
True,
    'truncated': False,
    'trafficLightColor':
'none'},
   'manualShape': True,
   'manualAttributes': True,
   'box2d': {'x1': 629.261939,
    'y1': 375.249876,
    'x2': 643.944715,
    'y2': 410.488545},
   'id': 355},
```

CLASSES AND DISTRIBUTION:

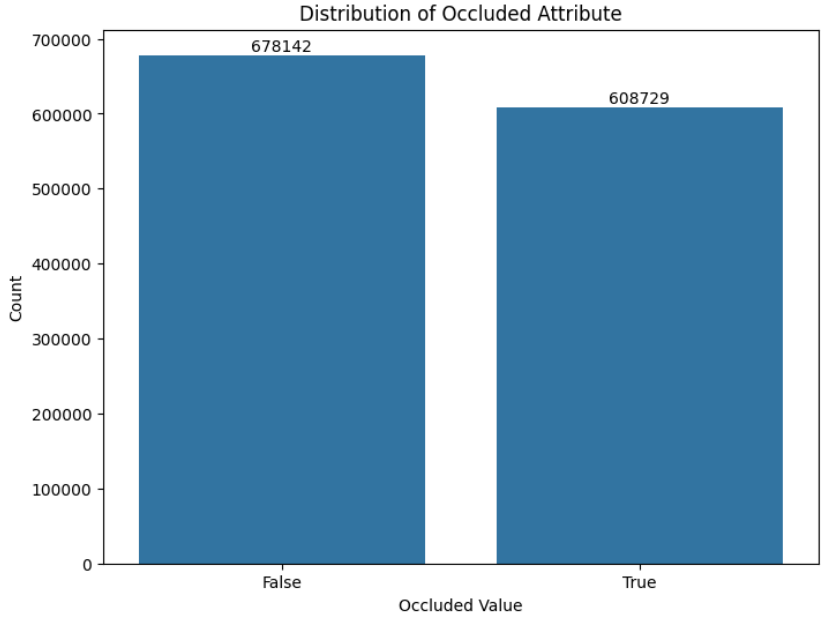
- **Classes**: Includes diverse categories like cars, pedestrians, truck, bus, and cyclists, essential for autonomous driving tasks.

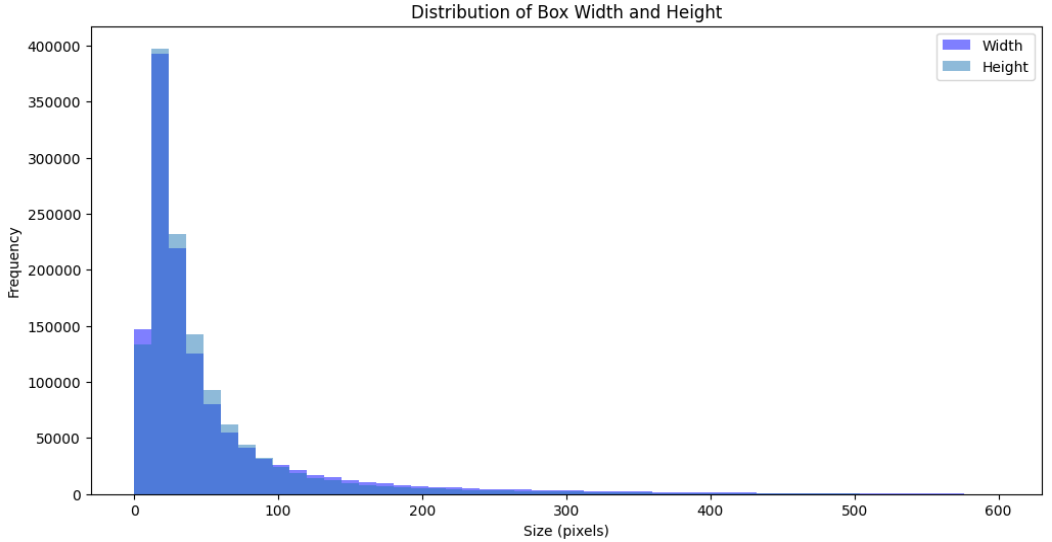
- **Distribution**: The class distribution is imbalanced, reflecting real-world scenarios where certain objects (e.g., cars) dominate.

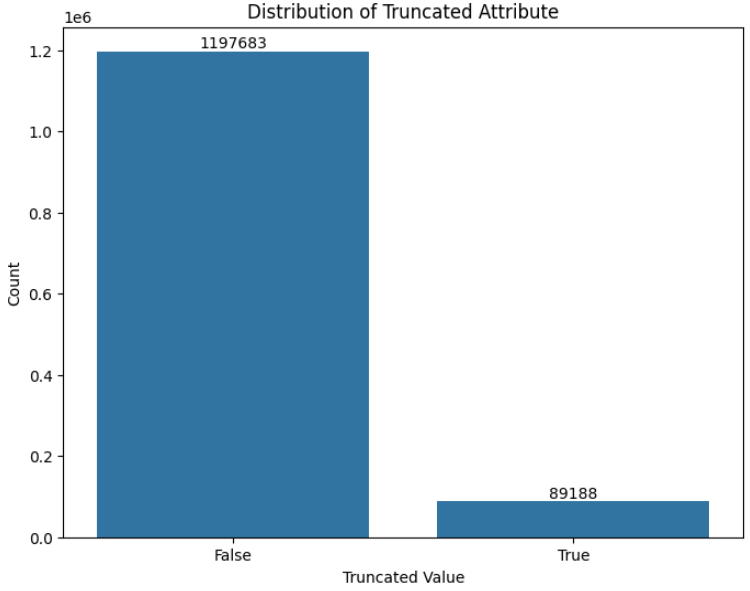


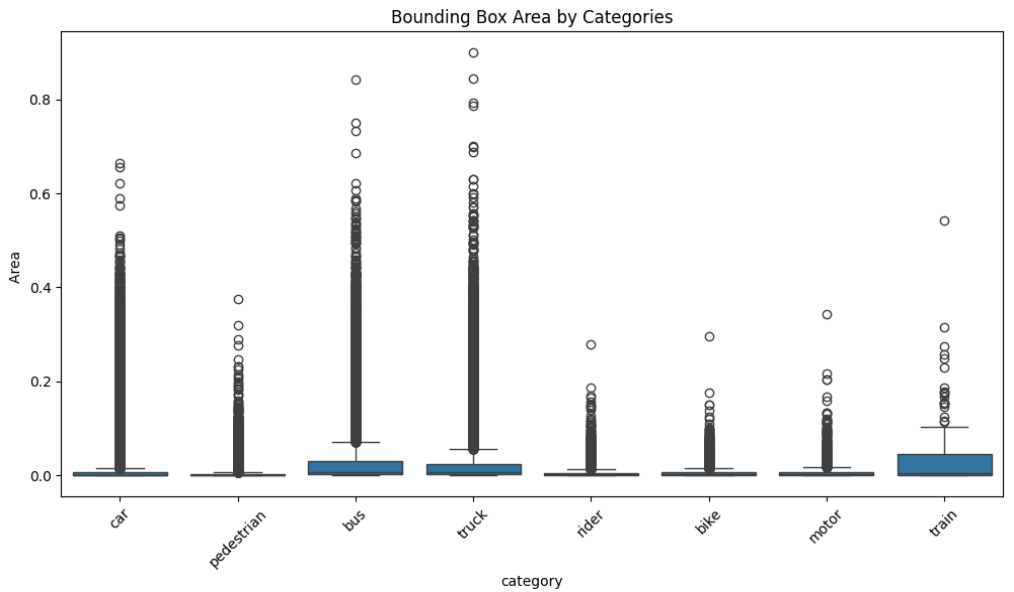


The average number of objects per image is: 17.78


Distribution of Occluded Attribute
678142
608729
Count
Occluded Value
False
True


Distribution of Box Width and Height
Width
Height
Frequency
Size (pixels)


Distribution of Truncated Attribute
1e6
1197683
89188
Count
Truncated Value
False
True

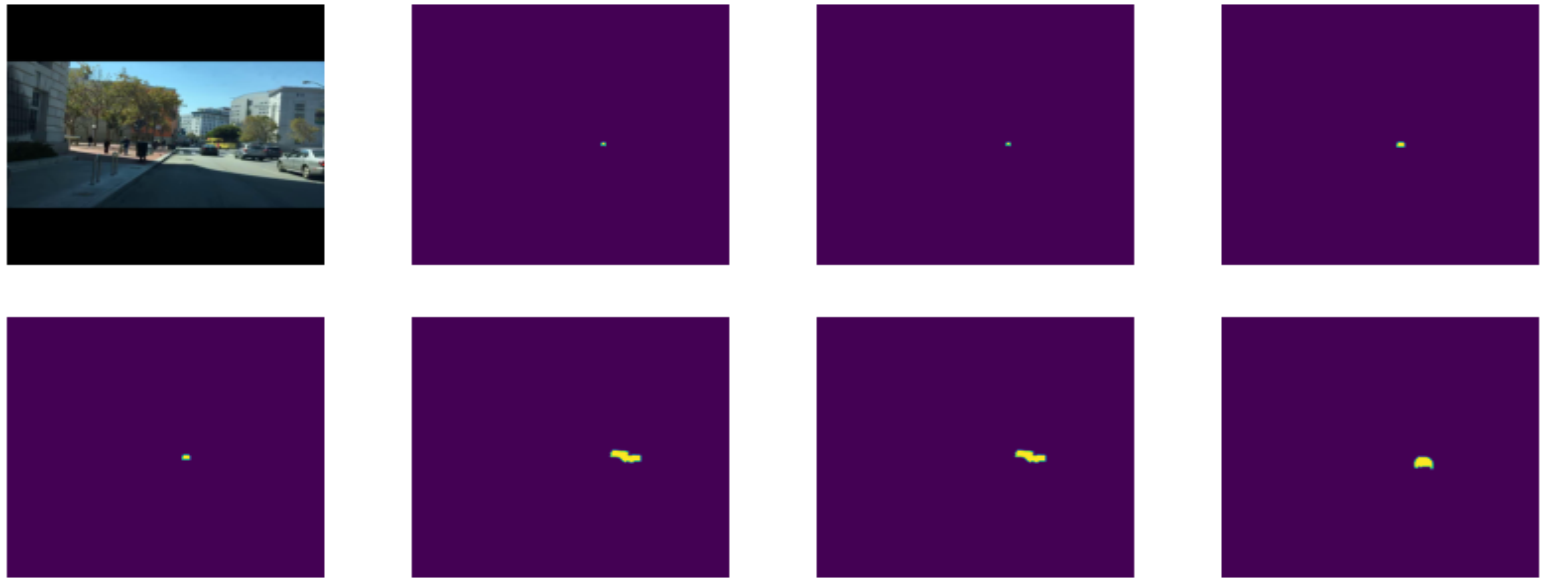

Bounding Box Area by Categories
Area
category
car
pedestrian
bus
truck
rider
bike
motor
train

DATA INSPECTION

IMAGE RESIZING: Images are resized to 1024x1024 with zero-padding, and bounding boxes are regenerated from masks for consistency.

RESIZING INSTANCE MASKS: Masks are cropped to bounding boxes and resized to smaller dimensions (e.g., 56x56) for efficiency.

INSTANCE MASKS BEFORE RESIZING:



RESIZED MINI MASK VISUALIZATION:

ANCHOR GENERATION:
Anchors are predefined reference boxes generated for each Feature Pyramid Network (FPN) level, sorted by pyramid level, feature map sequence, and aspect ratio.

- **DATA GENERATION:**

A generator processes input data, sampling 2000 random ROIs per image, producing batches of ROIs, ground truth, and detection targets for training.

- **ROI REFINEMENT:**

Refinements align ROIs with predictions, ensuring high-quality proposals; positive ROI rates were analyzed, yielding ~33% positives over 10 images.

About Mask R-CNN and Model Training

Mask R-CNN

- MASK R-CNN IS BUILT ON TOP OF FASTER R-CNN .
- MASK R-CNN EXTENDS FASTER R-CNN BY ADDING A BRANCH FOR PREDICTING SEGMENTATION MASKS ON EACH REGION OF INTEREST (ROI), IN PARALLEL WITH THE EXISTING BRANCH FOR CLASSIFICATION AND BOUNDING BOX REGRESSION.
- THE FIRST STAGE OF FASTER R-CNN, CALLED THE REGION PROPOSAL NETWORK (RPN), IS RESPONSIBLE FOR PROPOSING CANDIDATE OBJECT BOUNDING BOXES.
- MASK R-CNN ADOPTS THE SAME TWO-STAGE PROCEDURE AS FASTER R-CNN, WITH AN IDENTICAL FIRST STAGE (WHICH IS THE REGION PROPOSAL NETWORK). IN THE SECOND STAGE, IN PARALLEL TO PREDICTING THE CLASS AND BOX OFFSET, MASK R-CNN ALSO OUTPUTS A BINARY MASK FOR EACH ROI.

Key Metrics for Evaluating Instance Segmentation Performance

AP (AVERAGE PRECISION):

- MEASURES OVERALL INSTANCE SEGMENTATION PERFORMANCE ACROSS VARIOUS **IOU** (INTERSECTION OVER UNION) THRESHOLDS.

- **AP** GIVES A COMPREHENSIVE EVALUATION OF MODEL PERFORMANCE.

AP50 & AP75:

- **AP50**: PRECISION AT AN IOU THRESHOLD OF 0.5 (MODERATE LOCALIZATION ACCURACY).

- **AP75**: PRECISION AT AN IOU THRESHOLD OF 0.75 (HIGH LOCALIZATION ACCURACY).

- THESE METRICS EVALUATE PERFORMANCE AT DIFFERENT LEVELS OF LOCALIZATION ACCURACY.

APS, APM, APL:

- **APS**: Average Precision for small objects.
- **APM**: Average Precision for medium-sized objects.
- **APL**: Average Precision for large objects.
- These metrics assess performance across different object scales.

Model Performance Comparison:

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

MODEL INITIALIZATION:
Mask R-CNN is initialized with COCO pre-trained weights for faster convergence.

Stage One: Training Heads Only
We begin by loading COCO-pretrained weights, but we exclude the final layers responsible for classification and mask prediction, since BDD100K has a different set of classes.

Stage Two: Fine-tuning All Layers
After the head layers are well-trained, we unfreeze the entire network and fine-tune all layers using a lower learning rate to avoid destroying the pre-trained weights.

Model Evaluation

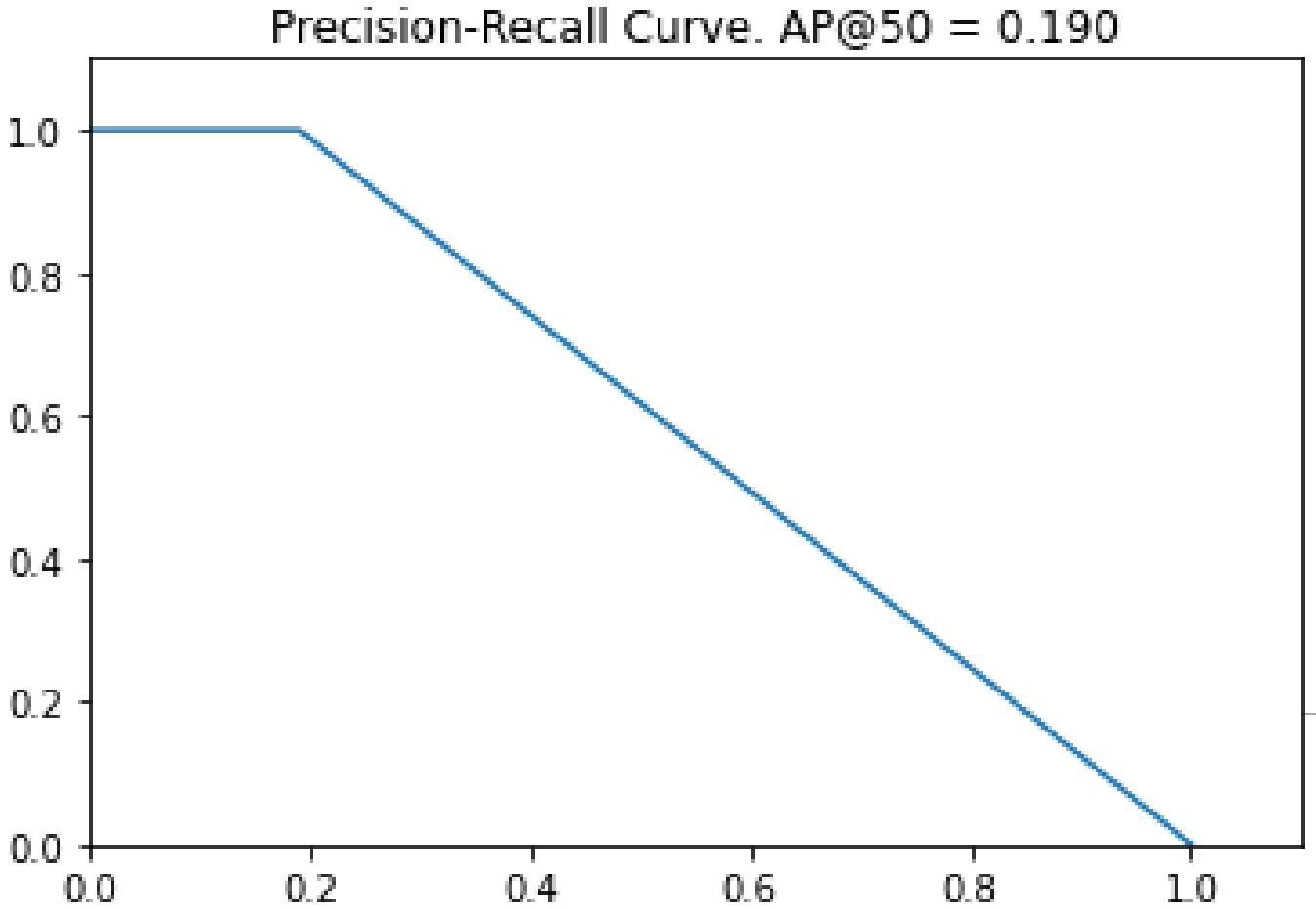
MODEL PERFORMANCE:

- The performance review reveals that the model is weak in identifying certain classes with fewer examples.

Dataset	No. of Images Evaluated	mAP @ IoU=0.5
Validation Set	200	0.49
Test Set	200	0.55

Mean Average Precisions on some test and validation set images:

- Low mAP value suggests that although detections are somewhat aligned, many are not precise or are missing entirely.
- Poor performance on rare classes likely results in low mean



Precision-Recall Curve at at IoU threshold = 0.5:

- **Initially, the model is highly precise but low recall (confident predictions on few easy objects).**
- **As the model tries to recall more objects, it begins to include less confident or incorrect detections, lowering precision.**
- **Class imbalance or confusing background could be hurting generalization.**

FUTURE SCOPE AND PERFORMANCE IMPROVEMENT STRATEGY:

- Data Augmentation: Using techniques like rotation, cropping, and color jitter to enhance generalization.
- Class Balancing: Applying techniques like resampling to handle class imbalance.
- Backbone Upgrade – Replacing backbone (e.g., ResNet50 → ResNet101) for better feature extraction.

Thankyou