Computer Vision on Road Scenes: Benchmarking, and Open World Object Detection

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Deepak Kumar Singh
2018701010
deepak.singh@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
March 2025

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Computer Vision on Road Scenes: Benchmarking, and Open World Object Detection"** by **Deepak Kumar Singh**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. C.V. Jawahar

To Mummy and Papa,

Acknowledgments

I would like to express my sincere gratitude to my adviser, Prof. C.V. Jawahar, for providing me with ample amount of freedom in research. I quit my job in 2016 to pursue research, and the CVIT lab provided me with every possible avenue to learn and grow as a research student. I'm also immensely grateful to have the opportunity to receive guidance from Prof. Vineeth Balasubramanian, Prof. Chetan Arora, Dr. Anbumani Subramaninan over the course of my research.

I would also like to thank Dr. Ajoy Mondal and Dr. Rohit Saluja for their guidance during my collaboration across research projects. I am also thankful to my fellow project mates, Shyam and Ameet with whom I have had some very useful, valuable and critical discussions in research. I am really fortunate to have my friend and fellow colleague Ashish Menon who has been part of my research journey at IIIT Hyderabad. A major portion of my research approaches were a direct product of our technical discussions. I am equally grateful to the seniors of the CVIT lab; Aditya Sir, Avijit Sir, Praveen Sir, Jerin, for giving insightful reviews and sharing their knowledge and experience. I would also like to thank my other lab mates and colleagues of CVIT; Riya, Kiran, Rudrabha, Harish, Prajwal, Raghava, and Prachi with whom I have had invaluable discussions at multiple levels be it technical, philosophical, or personal. I also thank my other friends of IIIT Hyderabad; Shreyas, Balaji, and Yash, who are associated to various other labs, without whom my journey during the course of MS in IIIT Hyderabad would not have been memorable.

I also take this opportunity to thank the CVIT and IIIT Hyderabad staff members. Siva Kumar, who was the first point of contact for me before joining CVIT and who helped me with the smooth transition. Rohitha and Aaradhana, for handling all the administrative help needed during the course of my research. Nandini, who helped me improve the quality of writing in my conference paper draft, and thesis draft. Dr. Semparithi Aravindan, who is in charge of the ada (GPU) cluster of IIIT Hyderabad for helping with the cluster related compute requirements. Last but not the least, I acknowledge the support and great love of my family, who kept me motivated. This work would not have been possible without their help. I am thankful to my mother, father, and sister. I am also blessed to have an extended family of friends; Nitish, Sharanya, Sailesh, Sunitha, and Lohith, who have always supported me and backed me with all my decisions starting from choosing IIIT Hyderabad for my Masters.

Abstract

In autonomous driving, we have multiple computer vision tasks like object detection, semantic segmentation, and instance segmentation which plays a crucial role in perceiving the environment around the vehicle. Understanding the behaviour and performance of such tasks helps improve and address the key issues that are inherent in the system. There can be issues which are latent in the deep learning architecture and also in the datasets on which the deep learning models are trained and tested. In this thesis, we benchmark the performance of various popular deep learning models on road scene datasets for various computer vision tasks and also formulate *open-world* object detection on road scenes by addressing the inherent issues present in road scene datasets.

In the first part of the work, we aim to understand the performance and behaviour of various deep learning models on road scene datasets; Cityscapes, IDD, and BDD. Object detection, semantic segmentation, and instance segmentation form the bases for many computer vision tasks in autonomous driving. The complexity of these tasks increases as we shift from object detection to instance segmentation. The state-of-the-art models are evaluated on standard datasets such as PASCAL-VOC and MS-COCO, which does not consider the dynamics of road scenes. Driving datasets such as Cityscapes and Berkeley Deep Drive(BDD) are captured in a structured environment with better road markings and fewer variations in the appearance of objects and background. However, the same does not hold for Indian roads. The Indian Driving Dataset(IDD) dataset is captured in unstructured driving scenarios and is highly challenging for a model due to its diversity. This work presents a comprehensive evaluation of state-of-the-art models on object detection, semantic segmentation, and instance segmentation on road scene datasets. We present our analyses and compare their quantitative and qualitative performance on structured driving datasets(Cityscapes and BDD) and the unstructured driving dataset(IDD); understanding the behavior on these datasets helps in addressing various practical issues and helps in creating real-life applications.

In the second part of the work, we model open-world object detection on road scenes since object detection is a crucial component in autonomous navigation systems. Current object detectors are trained and tested on a fixed number of *known* classes. However, in real-world or open-world settings, the test set may consist of objects of *unknown* classes; this results in the *unknown* objects being falsely detected as *known* objects leading to the failure in the decision making of autonomous navigation systems. We propose Open World Object Detection on Road Scenes (ORDER) to resolve the aforementioned problem. We introduce Feature-Mix that widens the gap between known and unknown classes in latent

feature space and improves the *unknown* object detection in the ORDER framework. We identify the inherent problems present in autonomous datasets: i) a significant proportion of the dataset comprises small objects and ii) intra-class bounding box scale variations. We address the problem of small object detection and intra-class bounding box variations by proposing a novel focal regression loss. Further, the detection of small objects is improved by curriculum learning. We present an extensive evaluation of two road scene datasets: BDD and IDD. Our experimental evaluations on BDD and IDD shows consistent improvement over the current state-of-the-art method. We believe that this work will lay the foundation for real-world object detection for road scenes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

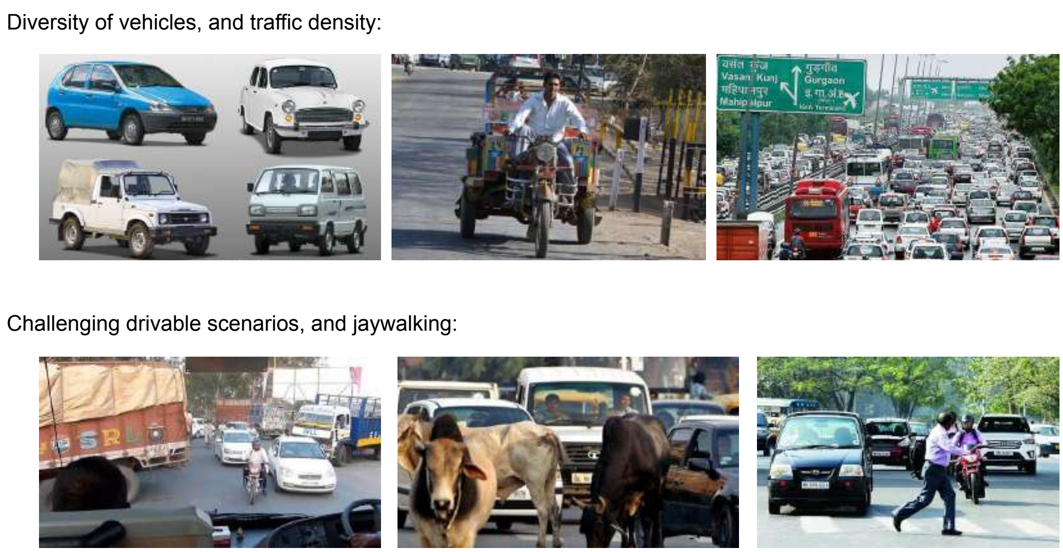


Figure 1.1: Real world conditions of unstructured driving scenarios. *[Image Credits: Google]*

An autonomous vehicle operates on various input modalities, comprising of monocular, stereo camera, fish-eye camera, Light Detection and Ranging (LIDAR), Inertial Measurement Unit(IMU), and Global Positioning System(GPS). Data from all these systems help in decision making, trajectory prediction, and collision avoidance. Even in the image domain, various predictions are necessary to perceive the environment. Tasks like object detection, semantic segmentation, and instance segmentation play a key role in understanding the scene. Each task becomes complex as we move from object detection to instance segmentation; even the label space changes from coarse to granular. Figure 1.1 shows the growing complexity of the real world environment in which a machine learning has to perform.

1.1 Motivation

One of the best way to address real-world challenges is to have a data-centric approach. We can collect datasets that are specific to a region, tasks, or domains to solve the underlying problems. Figure 1.2 shows the various sub-parameters in each of these areas. Any model is as good as the data on which it is trained, having a dataset which can capture and represent the possible scenarios in which a model will be subjected to will increase the flexibility of the model and also reduce the false positive cases. Majorly, road scene applications are susceptible to unpredictable conditions, especially in unstructured road conditions in countries such as India.

We mainly need algorithms and models that are versatile to work in unconstrained scenarios, they should be able to detect unknown objects present in road scenes and continuously update there knowledge without forgetting the previous knowledge.

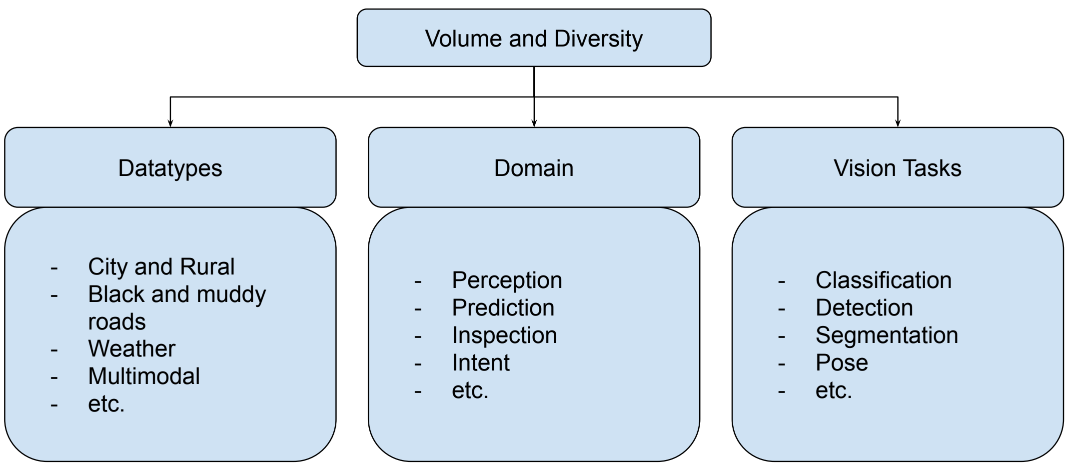


Figure 1.2: Data-centric approach to address and model road scene applications

1.2 Problem Context

The behaviour of various computer vision tasks mainly depends on two factors; a) the underlying data on which a model is trained, b) formulation of the model which inculcates various parameters to model the problem, resulting in a versatile model.

1.2.1 Why we need Benchmarking on Road Scenes

Figure 1.3 shows a table where the columns comprise of the popular object detection, semantic segmentation, and instance segmentation models, the rows constitute the commonly available datasets on which these models can be evaluated. We notice that many of the models' performance is reported on MS-COCO and VOC datasets. In contrast, on road scenes dataset, only fewer models have reported

		Object Detection Models				Semantic Segmentation Models			Instance Segmentation Models		
		Faster R-CNN	SSD	RetinaNet	YOLOv3	PSPNet	ERFNet	DRN	Mask R-CNN	Cascade R-CNN	MS R-CNN
Academic	COCO	✔	✔	✔	✔	✔	✗	✗	✔	✔	✔
	VOC	✔	✔	✗	✗	✔	✗	✗	✗	✔	✗
Road Scenes	Cityscapes	✗	✗	✗	✗	✔	✔	✔	✔	✗	✗
	BDD	✔	✗	✗	✗	✗	✗	✔	✔	✗	✗
	IDD	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Figure 1.3: Popular tasks like object detection, semantic segmentation, and instance segmentation are commonly trained and reported on standard academic datasets. The behavior of the models on road scene datasets are not available.

the behavior on Cityscapes. Road scene datasets differ significantly from standard academic datasets like MS-COCO and VOC. They are application specific and cover the dynamics of road-scene behavior.

		Object Detection Models				Semantic Segmentation Models			Instance Segmentation Models		
		Faster R-CNN	SSD	RetinaNet	YOLOv3	PSPNet	ERFNet	DRN	Mask R-CNN	Cascade R-CNN	MS R-CNN
Academic	COCO	✔	✔	✔	✔	✔	✗	✗	✔	✔	✔
	VOC	✔	✔	✗	✗	✔	✗	✗	✗	✔	✗
Road Scenes	Cityscapes	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	BDD	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
	IDD	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Figure 1.4: Coverage of our evaluation on the popular road scene datasets.

From a road safety perspective, factors governing road infrastructure, accessibility and regulations, and driver behavior contribute significantly to the accidents on Indian roads. Models developed on datasets that are structured and do not represent the dynamics of Indian roads can also perform poorly and affect the critical decision-making of a system. In India, there is a high density of road objects, complex movement patterns, trees on either side of the road cause poor visibility and shadows as well.

1.2.2 Why we need to detect unknown objects

1.2.2.1 Concerns on Road Safety

We can notice in Figure 1.6 that as we move from Structured road scenes to unstructured road scene conditions, the complexity increases various folds. New objects can be present in the scene resulting in false positive matches. Due to complex traffic movement and high density of vehicles on the road, trajectory prediction can be a challenging problem. To address these underlying issues at a fundamental level, we first study the behavior of various computer vision tasks such as object detection, semantic

Figure 1.5: Increasing complexity of various parameters in unstructured driving conditions.

segmentation, and instance segmentation on various road scene datasets, we identity and address two inherent problems present in unstructured and structured road scene datasets in the context of the open world setting. In the following sections we discuss various computer vision tasks which are fundamental in building applications related to road scenes.

1.2.2.2 Incremental Learning of Unknown Objects

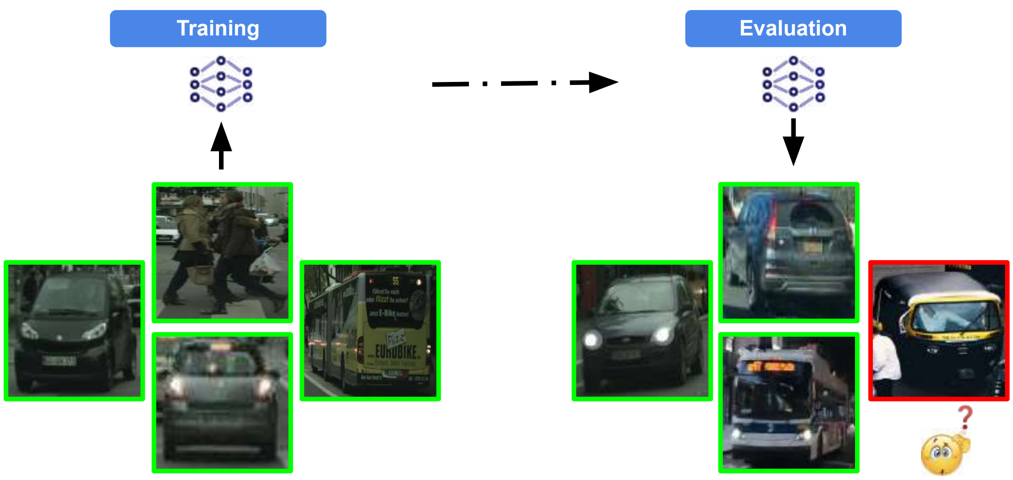


Figure 1.6: Presence of Unknown classes in the real-world.

A computer vision model is traditionally modeled to detect the objects on which it was trained, these objects come from a small set of object space which represent the possible variation in size, shape, color, and size. But when the model is deployed in the real-world, it is subjected to classes and instances which are very different from the original object space, these leads to detection in high amount of false-positives. A versatile model should be able to; **a) detect the unknown classes as a separate *unknown* class**, and **b) Leverage the information present in the detected unknown objects space and continually update the model without drastically forgetting the current information present in the model.**

1.3 Thesis Focus

The thesis focuses on two major aspects of road scene datasets: **evaluating the behavior of models on road scene datasets** discussed in Chapter 3 and **addressing the key inherent issues in road scenes for formulating open world object detection** discussed in Chapter 4. We tackle the former by evaluating the popular object detection, semantic segmentation, and instance segmentation models on road scene datasets. We understand the behavior and their fall-backs. Our goal here is also to lay the foundation to focus on key areas and issues present in unstructured road scenes.

When an object detection model is deployed in a real-life environment, it is subjected to instances where the model is not trained. Predicting unknown objects as known objects with high or low confidence can impact the safety of the application. Autonomous driving applications should have very low errors in such scenarios since they affect both the driver and pedestrians. We formulate open-world object detection on road scene datasets. We address two key issues which are highly pronounced in road scenes, i.e., intra-class size variation where objects belonging to the same class have varying bounding boxes, such issues are not prominent in standard academic datasets such as MS-COCO and VOC. The other issue is of imbalance in the dataset, road scene datasets mainly comprise small objects due to the presence of objects which are farther in the scene. We address the former by introducing a novel bounding box regression loss that considers the bounding box area. The latter is resolved by incorporating curriculum learning.

Through our work, we want to focus our attention on addressing the issues present in road scene datasets and also build solutions that are well suited to Indian driving scenarios. Due to the high domain gap among Indian, European, and American roads, approaches addressing these issues need to be built to translate better and incorporate the dynamics present in Indian road scenes.

1.4 Contribution

The major contributions of our thesis are two folds; to understand the behavior of models on road scenes and address the key issues present in road scene datasets:

A. Benchmarking Computer Vision Tasks on various road scenes datasets

1. We study the effect of various computer vision tasks such as object detection, semantic segmentation, and instance segmentation across various road scene datasets.

2. We identify the reasons why certain classes do not perform well compared to others.

In the first work, the motivation emerged when we noticed that there had been no works that studied the behavior of popular models on road scene datasets, figure 1.3 shows the comparison of the availability of results of popular object detection, semantic segmentation, and instance segmentation models on road scenes datasets. We perform a detailed study by covering the missing evaluations, figure 1.4 shows the a comprehensive availability of results on all the three popular road scene datasets, Cityscapes, BDD, and IDD.

B. Formulate Open World Object Detection to address inherent issues of road scene datasets.

1. We propose an approach to address open-world object detection, where the model is subjected to new novel classes during testing, and it incrementally learns the new classes without forgetting the old classes.

2. We identify and address two inherent problems present in road scene datasets by formulating a novel loss function and training strategy.

In the second work, our goal was to formulate open-world object detection on road scenes. In open world scenario, a model is subjected to instances on which it was not trained upon, during this time the model has to identify the unknown instances as *unknown*, so that false detection can be avoided, and the model should also incrementally learn from the unknown instances. We identified two key issues present in popular road scene datasets, IDD and BDD, and addressed the reasons for poor performance in an open-world setting.

1.5 Thesis Organization

We now give a brief description of the structure of the thesis. Chapter 2 provides an overview about the open-world aspect of object detection. Traditional object detection models are formulated on a close-set problem where the training and the test classes are the same, but such models do not consider exposing a model to images that have unknown objects. Chapter 3 addresses the need for evaluating state-of-the-art object detection, semantic segmentation, and instance segmentation on road scene datasets. We evaluate the popular approaches taken in these 3 areas on road scene datasets Cityscapes, IDD, and BDD. Chapter 4 formulates open-world object detection on road scenes by addressing the key issues inherent to road scene datasets i.e. intra-class size variation and skewed distribution towards small objects in road scene datasets. In conclusion, we summarize our finding along with the possible future directions.

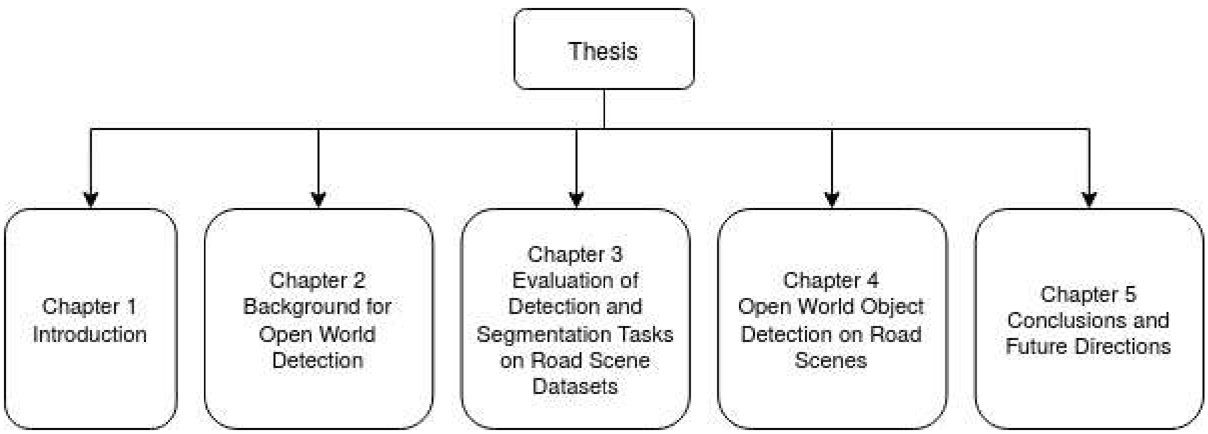


Figure 1.7: Hierarchy of chapters presented in this thesis

Chapter 2

Computer Vision Tasks and Open World Learning

2.1 Computer Vision Tasks

Applications concerning autonomous driving are safety-critical, and having an understanding of the behaviour of the underlying deep-learning model is crucial. This chapter briefly introduces the deep-learning tasks and the popular datasets used to understand road scenes better. We focus on understanding the behavior of object detection, semantic segmentation, and instance segmentation. These tasks increase in complexity as the granularity of the labels increase from object detection to instance segmentation. The following sections explain the basic idea behind the three approaches.

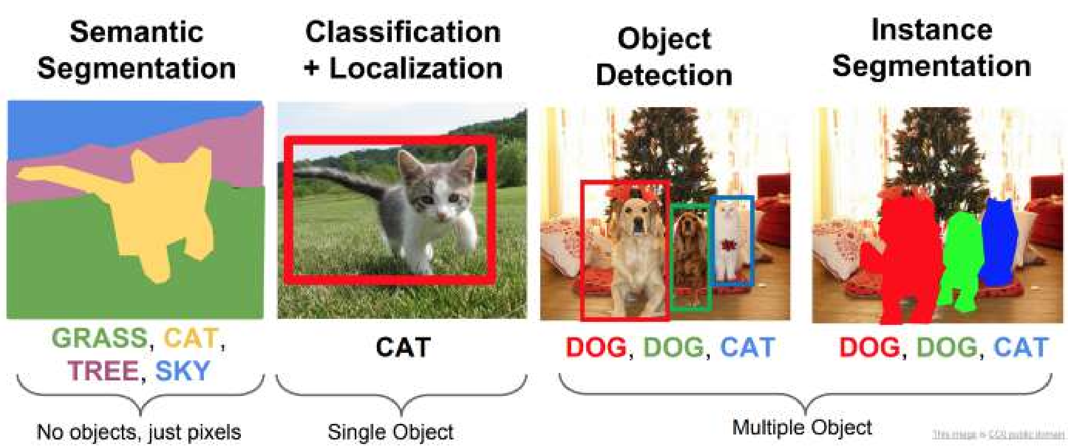


Figure 2.1: Different computer vision tasks in the image domain. [Image Credits: Google]

2.1.1 Object Detection

The task of object detection entails identifying as well as localizing instances of an object. Before deep-learning, most approaches relied on hand-crafted feature extraction, examples include Viola-Jones detector[26] and Histogram of Oriented Gradients (HOG)[11]. Now, object detection is applied from

Figure 2.2: Result of object detection on a road scene. Bounding box and labels are drawn on a predicted object

medical image analysis to self-driving cars. Some key challenges present in object detection include: (i) Class-imbalance, where certain object classes are abundantly labeled than certain rare classes. (ii) Intra-class variation in which instances belonging to the same class have high variation in their appearance due to lighting, pose, or occlusion. Even the variation in the background plays a major role in the overall performance of a model.

2.1.2 Semantic Segmentation

Figure 2.3: Result of semantic segmentation on a road scene. Pixel-wise classification is performed on every frame.

In the semantic segmentation task, we assign a category label to each pixel of an image. Applications such as self-driving vehicles, pedestrian detection, and human-assisted medical diagnosis are some of the areas where semantic segmentation plays a key role. Multiple supervised semantic segmentation methods are available, which assume that enough original images and their corresponding pixel-wise semantic-annotated images are available. The difficult and confusing aspect of semantic segmentation is assigning labels to pixels that are present at the boundary of other objects, and also when an object becomes disconnected due to the presence of another object. For example, a pedestrian in front of a car divides the car into two regions. Figure 2.3 shows a road scene where each pixel is semantically labeled.

2.1.3 Instance Segmentation

In instance segmentation, we assign different labels for each instance of an object, even if it belongs to the same class. In a way, it is a combination of object detection and semantic segmentation task. Instance segmentation is widely applied in robotics, autonomous driving, surveillance, and other areas. Various factors like size, illumination, background, blur, noise, and occlusion. affect the performance of a detection model, indirectly affecting the instance segmentation approach.

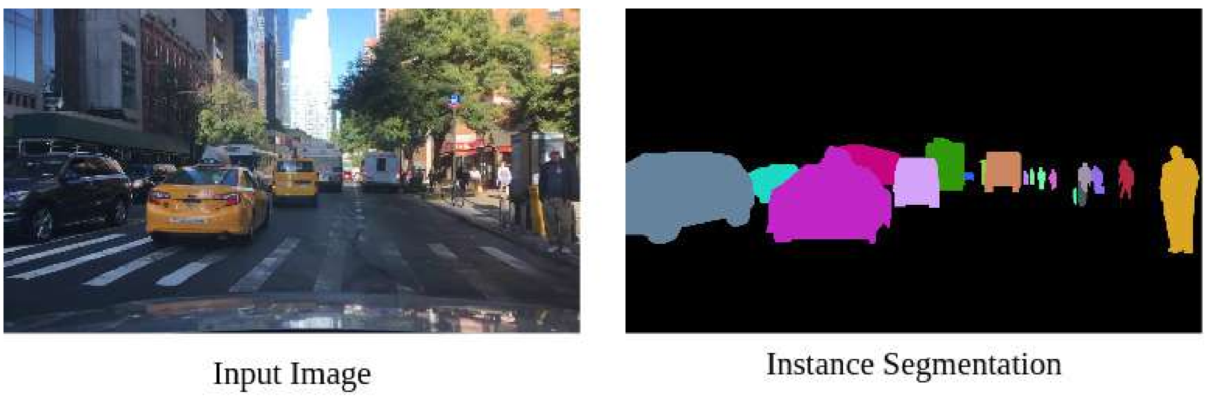


Figure 2.4: Result of instance segmentation on a road scene. Each occurrence of an object is unique and is assigned a different identity/color.

2.2 Road Scene Datasets

There have been efforts across the vision community to build relevant applications to help in addressing road safety issues. Some of the efforts in this direction have led to the creation of these popular datasets which now act as a benchmark for measuring the performance of vision tasks.

2.2.1 Cityscapes

Cityscapes [9] focuses on a semantic understanding of urban street scenes. It provides fine and coarse annotations at pixel-level, extended with instance-level labels for pedestrians and vehicles. There are 30 annotated classes, grouped into 8 categories: flat, construction, nature, vehicle, sky, object, human, and void. nineteen common classes are evaluated for benchmarking. It is collected across 50 different cities at varying times of the year to cover different weather conditions. For the segmentation task, Cityscapes provides dense pixel-level annotations for 5000 images at 1024×2048 resolution, of which 2975 is usually used for training, 500 for validation, and 1525 for test. There are coarse grain segmentation annotations of 19998 images to train data-heavy models.

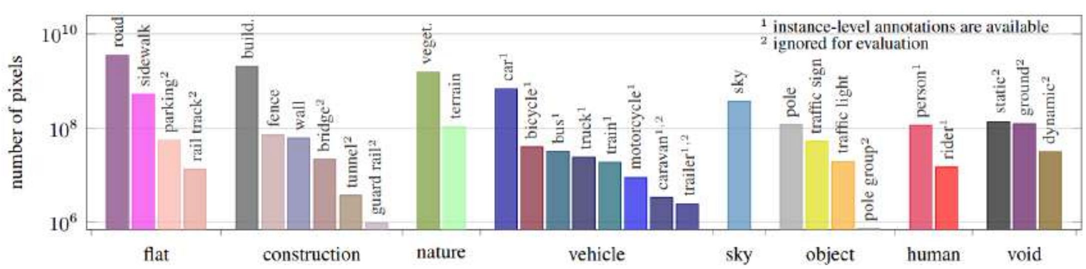


Figure 2.5: Illustrates the pixel count and semantic groups across each category. Image Credit: [9]

2.2.2 Berkeley DeepDrive Dataset

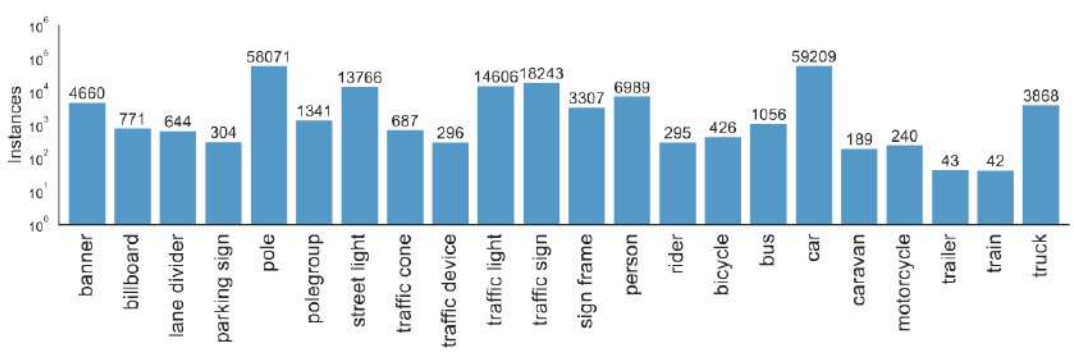


Figure 2.6: Distribution of number of instances across semantic instance segmentation. Image Credit: [69]

Berkeley DeepDrive Dataset(BDD)[69] is currently the largest video dataset on driving scenes with 100K videos and ten tasks. It comprises weather, environmental, and geographic diversity. The ten tasks comprise of image tagging, lane detection, road object detection, semantic segmentation, instance segmentation, drivable area segmentation, domain adaptation, imitation learning, multi-object detection tracking, and multi-object segmentation tracking. Having a set of diverse tasks help in building multi-task learning application. The dataset comprises 720p high-resolution images collected at a frame rate of 30fps, and reading of GPS/IMU are also available to preserve the driving trajectories. The 100K driving videos are collected from more than 50K rides, covering New York, San Francisco Bay Area, and other American cities. The videos are split into training (70K), validation (10K), and testing (20K) sets. For image tasks, every 10^{th} frame of the video is annotated, and for tracking, the entire video sequences are used. This dataset also provides diversity based on time of day and scene types.

2.2.3 India Driving Dataset

IDD [60] is the first road scene dataset created for unstructured environments, while most other popular datasets are from structured environments where the lanes are well-delineated, have a small number of well-defined categories, low variation in objects and background. IDD comprises of 10004 images, with 34 finely annotated classes collected from 182 drive sequences on Indian roads. It also provides a four-level label hierarchy to accommodate the complexity of object labels in Indian roads. The challenges of an unstructured environment are dominant in Indian roads and consist of; ambiguous road boundaries, diversity of vehicles and pedestrians, extensive use of information boards, diversity of ambient conditions. Having such a diverse dataset addresses issues caused due to domain gaps. On Indian road scenes, there is a high variance among novel classes like animals and vehicles. The intra-class variance is high and prominent in vehicles due to the large range of vehicle models.

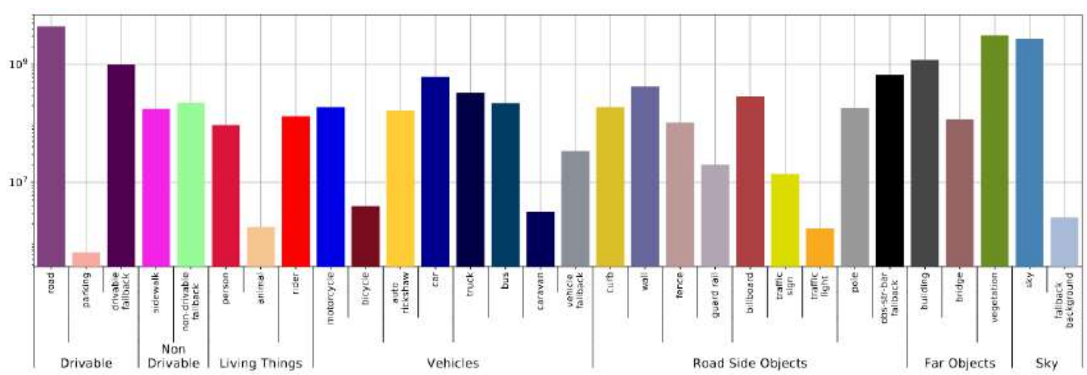


Figure 2.7: Pixel counts of individual label across the labels. Image Credit: [60]

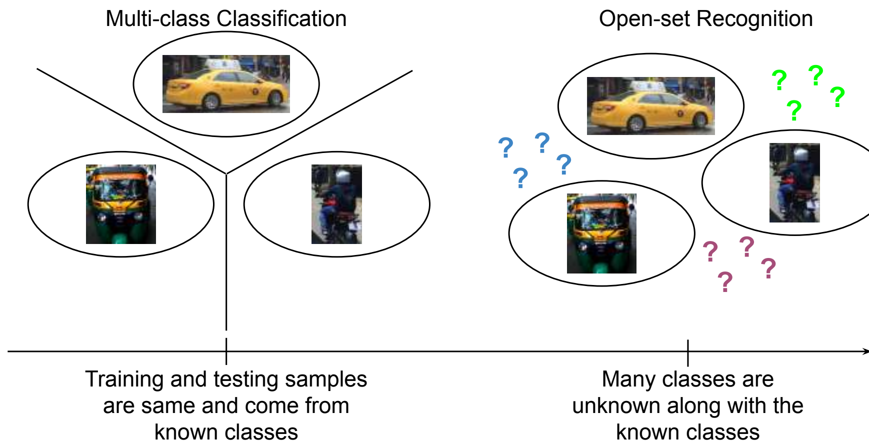


Figure 2.8: Closed-Set Recognition v/s. Open-Set Recognition

2.3 Open-Set Recognition

Traditional machine learning models in computer vision for recognition are formulated on a "closed-set", where the training and the testing classes are the same. In the real world, an application is subjected to scenarios where all the classes are not known, so the test and the train classes are not the same, there are some *unknown* classes as well. This scenario is referred to as "open-set" recognition. In open-set, there are scenarios where even the instances were not seen during training, this arises due to the limitations of labeling objects among the possible variations caused in the real world. The following equation was proposed by Scheirer et al. to quantify the openness caused by classes. It ranges the openness in percentage between 0 and 100, where 0 represents a close-set environment, and higher values represents the degree of openness present in the system.

The approach used by Scheirer et al. [55] comprised of a One-vs-Set machine which also define *open-space risk* which is caused by labeling a sample far from the known training examples. They performed experiments on labeled face data, it comprised of binary SVM linear kernel, binary 1-vs-set machine linear kernel, 1-class SVM linear kernel, and 1-class 1-vs-set linear kernel. The experiment on SVM found separate classes as negative and positive, the negative set is assigned only for known objects. The unknown objects are unclassified and present in open-space, which could be part of a positive set. Probabilistic models were developed by [56, 24] included non-linear classification in the multi-class setting. They suggested compact abating probability(CAP) based on Weibull-calibrated SVM, which decreases the probability of a class when points move from known data to the open-space.

$$\text{openness} = 1 - \sqrt{\frac{2 \times |\text{training classes}|}{|\text{testing classes}| + |\text{target classes}|}}$$

Figure 2.9: Equation for Openness

2.4 Open-World Learning

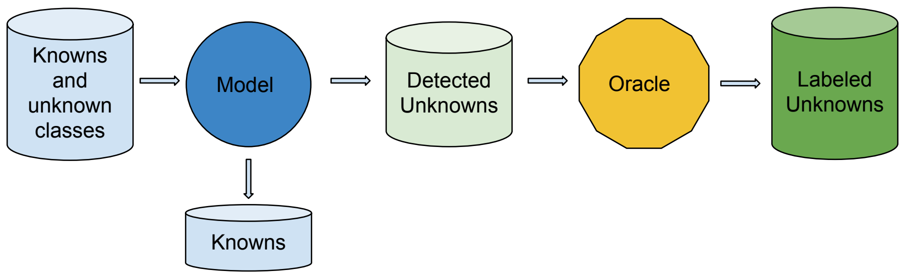


Figure 2.10: Data-flow in Open-World Learning

The work of Bendale and Boult in [2] introduced an OpenMax classifier which used deep networks and Weibull distribution to estimate open-set risk. The primary goal of an open-world system is to identify the *unknowns* and to incrementally learn them when labels for the *unknowns* are available. Many systems assume the existence of an **Oracle** which has global information about the classes and assigns specific labels to the *unknown* classes. Figure 2.10 shows the processing pipeline where we identify unknowns and incrementally update the model. The methodology in OpenMax included rejecting the unseen classes during testing. Traditional deep learning models use SoftMax layer upon the fully connected layer as output. But SoftMax produces probability for the known labeled classes. But OpenMax evaluates likelihood of an input being for an unseen class. It rejects unrelated images, reduces the error rate, and also manages open-space risk. So instead of normalizing the probability performed in SoftMax, OpenMax estimates class by measuring the distance between the model vector aimed at the limited upper classes and activation vector of the input. In this way it generates the likelihood of unknown classes.

In Open-Set Nearest Neighbor (OSNN)[28], to find similarities between two classes, OSSN used a ratio of similarity instead of a similarity score and applied a threshold on it. They also propose a system that measures adaptation in an existing open-set classification system called Normalized Accuracy (NA) and Open-Set-F-Measures (OSFM) and evaluate the performance on Caltech-256 and other datasets.

Recognition systems play an important role in identifying seen and unseen classes of images. The knowledge graph-based approach suggested in [42] identifies unknown visuals and recognize them in the open-world. The standard classification settings can predict only classes that are available in training, and images that are not available in training cannot be accurately predicted by standard classification. In the zero-shot setting, it can predict images that are not available in training but there is some partial information available for novel classes. In an open-world setting, it can predict images that are neither available in training nor any partial information about its classes. Using the knowledge graph embedding model and image embedding model, the knowledge graph uses the properties whereas the image embedding model uses images for training. Finally, the image embedding models use the properties of the knowledge graph to predict open-world images.

2.5 Evaluation Metrics

2.5.1 Wilderness Impact

To measure the effect caused by introducing *unknowns* in object detection Dhamija et al. in [13] introduced *Wilderness Impact*.

$$\text{Wilderness Impact } (WI) = \frac{P_{\mathcal{K}}}{P_{\mathcal{K} \cup \mathcal{U}}} - 1$$

Here, $P_{\mathcal{K}}$ refers to the precision of the model when evaluated on *known* classes, and $P_{\mathcal{K} \cup \mathcal{U}}$ is the precision when evaluated on *known* and *unknown* classes, measured at a recall level. To measure the frequency of unknowns in images, [13] also introduced *wilderness*, defining a ratio *Wilderness Ratio*.

$$\text{Wilderness Ratio } = \frac{\#imageswith\mathcal{U}_{\mathcal{M}}}{\#imageswith\mathcal{K}}$$

Here K constitute of images which has *known objects* or *objects of interest* which the object detection model is trained to detect. U_M consists of images which has instances which are not labeled and are *unknown*, this happens because every possible category cannot be labeled in the real world.

2.5.2 Absolute Open-Set Error

Dimity et al. [44] introduced this metric which quantifies a model by counting the number of instances which gets assigned the label of a *known* class but are actually *unknown* instances. The labels of a known category is assigned when IoU ≥ 0.5 between the predicted bounding box and the ground truth bounding box. We should also note that in an ideal situation, the Absolute Open-Set Error(A-OSE) should be 0 when tested on a set of *known* objects.

2.6 Datasets used in Open-World Learning

Due to the peculiar requirement of including a novel class during testing, traditional datasets needed to be modified to conform to the experiment settings. Here is a short list of the popular vision datasets commonly used in open-world learning settings.

Caltech-256[18]: It has 256 categories of objects with a total of 30607 images. Each category is further labeled with 3 stages; *good*, *bad*,*none*, *good* represents clear vision, *bad* indicates clutters, and none indicates that the image does not belong to the associated category.

ImageNet[12]: This is a large scale datasets of visual objects, the structure has been inspired from *WordNet*. It has 80000 *synsets* of *WordNet*, the basic version of the dataset contains 3.2 million images with 12 sub-trees and 5247 *synsets*. The sub-trees consist of the following categories: flower, furniture, fish, fruits, tools, bird, reptile, amphibians, geological formulations, vehicles, musical instruments.

Tiny-ImageNet[31]: This is a collection of 100000 images that are retrieved from internet. It has 200 categories of images of which 100000 are for training, 10000 for validation, and 10000 is for testing.

The images are of 32×32 and 64×64 pixels. This dataset is suitable for general purpose algorithms due to its high-level of noise and low resolution.

CIFAR-10[30]: This dataset was developed by Canadian Institute of Advanced Research, and consists of 10 classes; dog, frog, automobile, bird, horse, ship, truck, airplane, cat, and deer. This is a subset of CIFAR-100, which consists of 100 classes. It comprises of 60000 colored images with resolution of 32x32 pixels, each class has 600 images, the dataset is split into two sets with 50000 images for training and 10000 for testing.

MNIST[32]: This is a handwriting dataset created by Modified National Institute of Standard and Technology(MNIST). There are 60000 training images and 10000 testing images, the training set is commonly split to form the validation set. The dataset is primarily used for optical character recognition hence each image comprises a digit with 28x28 pixels, and each pixel is either white or black.

Fashion-MNIST[67]: This dataset is similar to MNIST, comprising 60000 training images and 10000 test images. Each image is of 28x28 pixels, and instead of being binary images, they are gray-scale images. There are 10 classes and is commonly used for benchmarking machine learning and artificial intelligence algorithms.

SVHN[46]: The Street View House Number(SVHN) dataset contains 600000 labeled digits cropped from real street view images. There are two types of images, one contains full numbers and the other comprises of cropped digits. The cropped digits are character-level ground-truth and each digit is of a resolution of 32x32 pixels, the full numbers consists of full-size original images in high-resolution with bounding boxes on each character. The dataset is split into 73257 images for training, 26032 images for testing, the extra images are commonly reserved as training images.

Chapter 3

Evaluation of Detection and Segmentation Tasks on Driving Datasets

Object detection, semantic segmentation, and instance segmentation form the bases for many computer vision tasks in autonomous driving. The complexity of these tasks increases as we shift from object detection to instance segmentation. The state-of-the-art models are evaluated on standard datasets such as PASCAL-VOC and MS-COCO, which do not consider the dynamics of road scenes. Driving datasets such as Cityscapes and Berkeley Deep Drive(BDD) are captured in a structured environment with better road markings and fewer variations in the appearance of objects and background. However, the same does not hold for Indian roads. The Indian Driving Dataset(IDD) dataset is captured in unstructured driving scenarios and is highly challenging for a model due to its diversity. This work presents a comprehensive evaluation of state-of-the-art models on object detection, semantic segmentation, and instance segmentation on road scene datasets. We present our analyses and compare their quantitative and qualitative performance on structured driving datasets(Cityscapes and BDD) and the unstructured driving dataset(IDD); understanding the behavior on these datasets helps in addressing various practical issues and helps in creating real-life applications.

3.1 Introduction

In computer vision, the granularity of the label increases as we move from object detection to instance segmentation. In object detection we perform classification and localization of the objects of interest, but in semantic segmentation we also consider the boundary of each object during classification. In instance segmentation, we differentiate each instance of the object during segmentation. Figure 3.1 captures the increasing complexity in each of the task. For autonomous driving applications, datasets like Cityscapes [9], BDD [69] are collected in structured conditions whereas IDD [60] is collected under unstructured driving conditions. Academic datasets such as PASCAL-VOC [14] and MS-COCO [37] are commonly used to benchmark object detection, semantic segmentation, and instance segmentation.

Our **contributions** in this work comprises of evaluation and analyses of various state-of-the-art deep learning models on object detection, semantic segmentation, and instance segmentation with structured

Input Image Object Detection Semantic Segmentation Instance Segmentation

Figure 3.1: Popular tasks of computer vision on road scenes. We can notice that the granularity of the label becomes more complex as we move from Object Detection to Instance Segmentation. (Best viewed in color and zoomed)

datasets - Cityscapes [9] and BDD [69], and an unstructured driving dataset - IDD [60]. We evaluate the performances with:

(i) four object detection models: Faster R-CNN [52], SSD [39], RetinaNet [36], and YOLOv3 [51]

(ii) three semantic segmentation architectures: PSPNet [72], ERFNet [54], and DRN [70], and

(iii) three instance segmentation techniques — Mask R-CNN [20], Cascade Mask R-CNN [5], and Mask Scoring R-CNN [22]

To our knowledge, this is the first comprehensive work to use driving datasets instead of standard academic datasets such as PASCAL-VOC[14] and MS-COCO[37], to perform quantitative and qualitative analyses of various deep learning models on multiple tasks. Understanding the behavior of state-of-the-art object detection, semantic segmentation, and instance segmentation techniques on driving sequences play a vital role in creating real-life applications.

3.2 Related Work

3.2.1 Object Detection:

Existing Deep Convolutional Neural Network (DCNN) based object detectors are of two categories: (i) two-stage detectors and (ii) one-stage detectors. The two-stage detectors comprise of a region-proposal step, region classification and regression step. Some popular works include [17, 16, 52], several modified architectures [20, 5] have also been developed to improve detection accuracy. Though two-stage detectors produce high detection accuracy, they cannot be used for real-time applications due to their high computation time. In contrast, one-stage detectors predict boxes from input images directly without a region proposal step and hence are time efficient, lending their use for real-time applications. The notable work of Redmon *et al.* in YOLO [48] laid the foundation for several other versions such as YOLOv2 [50] and YOLOv3 [51]. Other popular one-stage object detectors are SSD [39], MT-DSSD [1], RetinaNet [36], M2Det [73], and RefineDet [71].

3.2.2 Semantic Segmentation:

Deep Convolutional Neural Network (DCNN) based semantic segmentation techniques demonstrate improvements by replacing the fully-connected layer in image classification network with convolution layers, calling it Fully Convolutional Network (FCN) [41]. Several methods [7, 72] have been developed to overcome the limitations of FCN [41]. While methods [72, 8, 41] have been developed by combining multi-scale features to improve the segmentation performance, another approach involves semantic segmentation based on structure prediction.

Running DCNNs on mobile platforms (e.g., drones, robots, and smartphones) require networks to work real-time on embedded devices with space and memory constraints. Some lightweight networks for real-time semantic segmentation do exist [64, 54, 62] in the literature.

3.2.3 Instance Segmentation:

Instance segmentation assigns different labels to each instance of an object belonging to the same category. Pose estimation, surveillance, robotics, and self-driving cars are some of the areas where instance segmentation plays a key role. Instance segmentation techniques are of two categories: (i) two-stage, and (i) one-stage. Some of the latest works for two-stage approaches constitutes Mask R-CNN [20], Cascade Mask R-CNN [5], Mask Scoring R-CNN [22], CenterMask [33], BCNet [29]. The popular examples of one-stage methods are PolarMask [68], YOLOACT [4], and SOLO [61].

3.3 Experiments

3.3.1 Datasets

We aim to understand the effects of various state state-of-the-art models on diverse road scene datasets for our experiments. We considered two structured driving datasets; Cityscapes and Berkeley DeepDrive(BDD). In these two datasets, there is low variation in the appearance of objects and also in the background, the road infrastructure is also well delineated with proper markings on the road. The same assumptions do not hold for Indian driving conditions. For unstructured driving sequences, we consider the Indian Driving Dataset(IDD) dataset.

3.3.1.1 Cityscapes

The Cityscapes [9] dataset is a large scale dataset with urban scenes collected in 50 different cities across Europe. It provides 5000 frames of high-quality pixel-level(fine) annotations, and a large set of 20000 weakly(coarse) annotated frames. There are 30 labeled classes, and each image can consist of multiple instances of each class.

3.3.1.2 Berkeley DeepDrive (BDD)

Berkeley DeepDrive (BDD) [69] is a diverse, and large-scale dataset of visual driving scenes. It consists of over 100K video clips. Each video is about 40 seconds long, 720p, and 30 fps. The videos are recorded using mobile phones, under different weather conditions and are collected from multiple cities in the United States. The dataset is split into training (70K), validation (10K), and testing (20K) sets.

3.3.1.3 Indian Driving Dataset (IDD)

The Indian Driving Dataset (IDD) [60] is a dataset of road scenes from unstructured environments in India. It consists of 10004 images, finely annotated with 34 classes collected from 182 drive sequences on Indian roads. A four-level label hierarchy provides varying degrees of complexity. It also has fallback class to accommodate unknown road objects.

3.3.2 Setup

For object detection and instance segmentation, we use the popular frameworks Detectron2 [65] and mmdetection [6]. Each detector model is trained with a batch of 8 images, learning rate of 0.02, momentum of 0.9, and weight decay factor of 0.0001. In the case of semantic segmentation, we use the hyper-parameters as suggested by the literature of the respective models. Each instance segmentation model is trained on a base learning rate of 0.01 with other hyper-parameters being same as the object detection training.

We evaluate object detection performance with the widely used mean Average Precision (mAP) [52, 39, 36, 51] metric. We denote AP for bounding box as AP_{box} and mask as AP_{mask}. We also provide class-wise AP_{box} and AP_{mask} for class-wise analysis at a threshold of 0.5. For semantic segmentation evaluation, we use the common metric of mIoU [41, 72, 54]. For instance segmentation evaluation, we use the Average Precision (AP) metric [34, 20, 5]. As in [37], we calculate AP by varying the Intersection over Union (IoU) threshold from 0.5 to 0.95 with a step of 0.05.

3.4 Results

3.4.1 Object Detection

3.4.1.1 Baselines:

We choose Faster R-CNN [52] as a two-stage detector, and SSD [39], YOLOV3 [51], and RetinaNet [36] as one-stage detectors for the object detection task.

Table 3.1: Results of object detection: Class-wise Average Precision (AP) on Cityscapes, BDD, and IDD datasets using Faster R-CNN, SSD, YOLOv3, and RetinaNet. The last row indicates mean Average Precision (mAP). **m. cycle:** indicates motorcycle, **t. light:** indicates traffic-light, **t. sign:** indicates traffic-sign, and **veh. flbk:** indicates vehicle-fallback. **FR, Y3, and RN:** indicates Faster R-CNN, YOLOv3 and RetinaNet, respectively. The bold values indicate the category wise best result among all the methods on respective datasets.

Class	Cityscapes				BDD				IDD			
	FR	**SSD**	**Y3**	**RN**	**FR**	**SSD**	**Y3**	**RN**	**FR**	**SSD**	**Y3**	**RN**
person	49.9	30.8	43.5	**51.0**	62.2	45.8	59.1	**65.0**	55.9	41.6	50.2	**57.2**
truck	35.2	32.4	30.9	**38.8**	61.9	59.7	58.7	**63.2**	**68.4**	61.3	57.8	66.8
m. cycle	36.4	32.4	34.8	**41.0**	45.5	34.7	46.0	**46.5**	**70.6**	62.8	63.1	68.6
rider	55.9	37.1	51.3	**56.9**	**48.3**	33.3	47.7	46.8	**59.5**	49.0	54.3	58.1
bus	**63.7**	58.3	58.3	60.9	61.7	60.2	59.4	**62.5**	**74.1**	69.5	67.3	73.5
bicycle	47.4	39.7	44.6	**50.7**	50.0	39.8	46.9	**51.7**	**54.8**	39.3	41.3	52.8
car	67.0	65.0	66.4	**69.8**	79.4	75.9	76.2	**80.6**	**71.0**	65.6	64.5	**71.0**
train	40.9	**47.6**	39.5	45.5	0.0	0.0	**3.3**	0.0	0.0	0.0	0.0	0.0
t. light	-	-	-	-	**64.3**	53.9	57.1	63.1	28.5	13.8	25.5	**29.4**
t. sign	-	-	-	-	**69.8**	64.6	66.5	69.2	**39.5**	27.6	27.3	38.3
caravan	-	-	-	-	-	-	-	-	0.0	0.0	0.0	0.0
auto	-	-	-	-	-	-	-	-	**74.1**	66.9	67.9	73.6
trailer	-	-	-	-	-	-	-	-	0.0	0.0	0.0	0.5
animal	-	-	-	-	-	-	-	-	26.4	20.1	20.2	**28.1**
veh. flbk	-	-	-	-	-	-	-	-	**10.0**	7.9	6.7	**10.0**
mAP	49.6	42.9	46.2	**51.8**	54.3	46.8	52.1	**54.8**	**42.2**	35.0	36.4	41.9

3.4.1.2 Discussion:

Table 3.1 presents object detection results using Faster R-CNN, SSD, YOLO3, and RetinaNet on Cityscapes, BDD, and IDD datasets. From the table, we observe that RetinaNet performs better than all other detectors on the structured driving datasets: Cityscapes and BDD. While Faster R-CNN obtains the best detection results among all the used models on the unstructured driving dataset: IDD. In the case of IDD, we also observe that all the used methods completely fail to detect objects like *train*, *caravan*, and *trailer* (AP very close to 0). It happens because of fewer amount of annotated images for those categories and unstructured road conditions. For similar reason, all the methods obtain less than 30% AP scores for object categories such as *traffic-light*, *traffic-sign*, *animal*, and *vehicle-fallback*. We also observe from the table that for the object category *train*, all models perform better on Cityscapes than IDD and BDD datasets. This is because of domain shift and frequent presence of the object in Cityscapes than on IDD and BDD. We find similar observation for object categories *traffic-light* and *traffic-sign* on which all the used models perform better on BDD than IDD due to geographic domain shift.

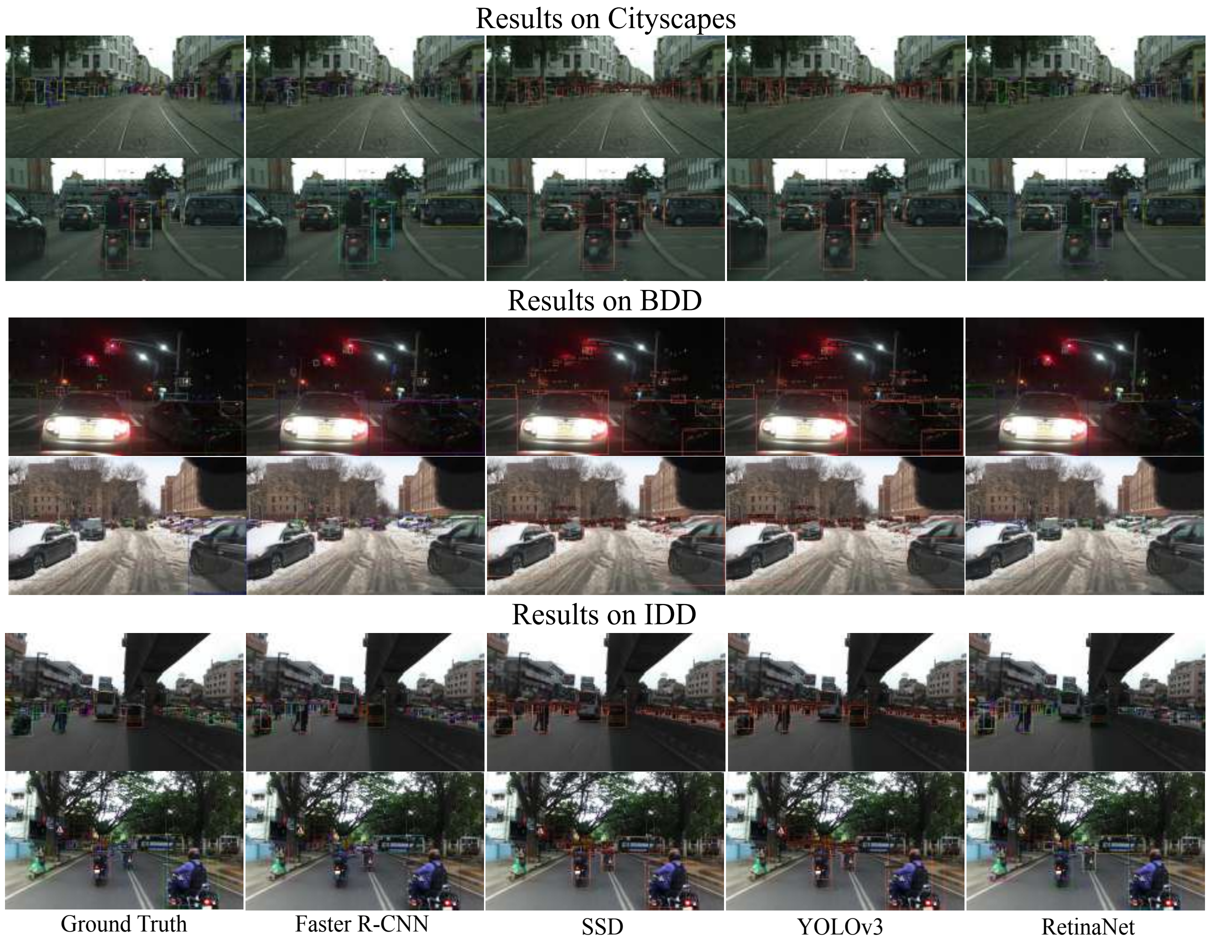


Figure 3.2: Some qualitative results of object detection on Cityscapes, BDD, and IDD datasets.(Best viewed in color and zoomed)

We present some qualitative results on few selected frames of Cityscapes, BDD, and IDD using the models: Faster R-CNN, SSD, YOLOv3, and RetinaNet in Figure 3.2. We choose frames under various complex conditions to establish the robustness of the used models. In Cityscapes, one of the selected frame is of an empty road with multiple pedestrians walking on the left and right sides of the road, and another image of a crowded traffic. We notice that Faster R-CNN, YOLOv3, and SSD detect accurate boundaries of all motorcycles and cars. But RetinaNet fails to detect the boundaries of few cars. In the case of BDD, the selected frames are of moving cars on a road at nighttime and during snowfall. In both cases, YOLOv3 and SSD detect all cars accurately. While Faster R-CNN and RetinaNet fail to detect cars that are far away from the camera.

In nighttime scenarios we notice that there are some false detection of *traffic-light* and *traffic-sign* caused due to headlights of vehicles. Notice that even in such challenging scenarios, all the models detect all trainable classes. However, Faster R-CNN and RetinaNet fail to detect few people accurately due to heavy occlusions caused by crowded vehicles and the shadows cast by the trees.

3.4.2 Semantic Segmentation

3.4.2.1 Baselines:

We choose three popular models PSPNet, ERFNet, and DRN to benchmark semantic segmentation task on driving sequences.

3.4.2.2 Discussion:

Quantitative score (mIoU) produced by the models are given in Table 3.2. We train and evaluate a model on various pair-wise combinations of datasets. From the table, we observe that all models achieve the best performance on the Cityscapes dataset. Even the trained model on Cityscapes is often considered a baseline for the segmentation of driving sequences.

It is also interesting to note that the trained model on Cityscapes does not generalize well. Using the model trained on Cityscapes to infer on BDD and IDD (data distribution is different from Cityscapes) resulted in lower performance (almost half of original). From the table, we also infer that Cityscapes is the simplest to learn while IDD is slightly more difficult, and BDD is the most difficult to learn. The trained model on Cityscapes performs poorly on out-of-distribution data points (i.e., IDD and BDD). The trained model on IDD also performs poorly on out-of-distribution data (i.e., BDD and Cityscapes), but it is relatively better than the model trained on Cityscapes. The trained model on BDD performs the best on out-of-distribution data (i.e., IDD and Cityscapes).

Table 3.2: Quantitative Results on semantic segmentation: Results of three different models: PSPNet, ERFNet, and DRN. The model is trained on one dataset but evaluated on all the other three datasets. Values in bold indicates best result among all the methods on respective test dataset.

Training Set	Test Set								
	Cityscapes			BDD			IDD		
	PSP Net	ERF Net	DRN	PSP Net	ERF Net	DRN	PSP Net	ERF Net	DRN
Cityscapes	**76.99**	72.20	71.35	35.06	29.37	38.72	38.46	31.37	40.30
BDD	43.75	33.95	50.77	47.40	37.84	**56.34**	39.70	30.10	46.19
IDD	42.69	28.31	46.43	39.51	28.89	41.91	62.95	59.39	**74.69**

Figure 3.3 shows visual results of semantic segmentation on Cityscapes, BDD, and IDD datasets using DRN. In the case of Cityscapes, the example images shown are (i) of a road with few moving cars and has adjacent buildings on both sides, (ii) a road with a single car and dense buildings and trees on both sides, (iii) a big truck is crossing a road with shadow cast by road side's trees and buildings, and (iv) multiple people are crossing a road. DRN produces few false segmentation for all images. While the example images of BDD are of (i) a clean road with few moving cars, (ii) a road with many cars and shadow cast by the road side adjacent trees, (iii) moving trucks and cars on a road with dense adjacent

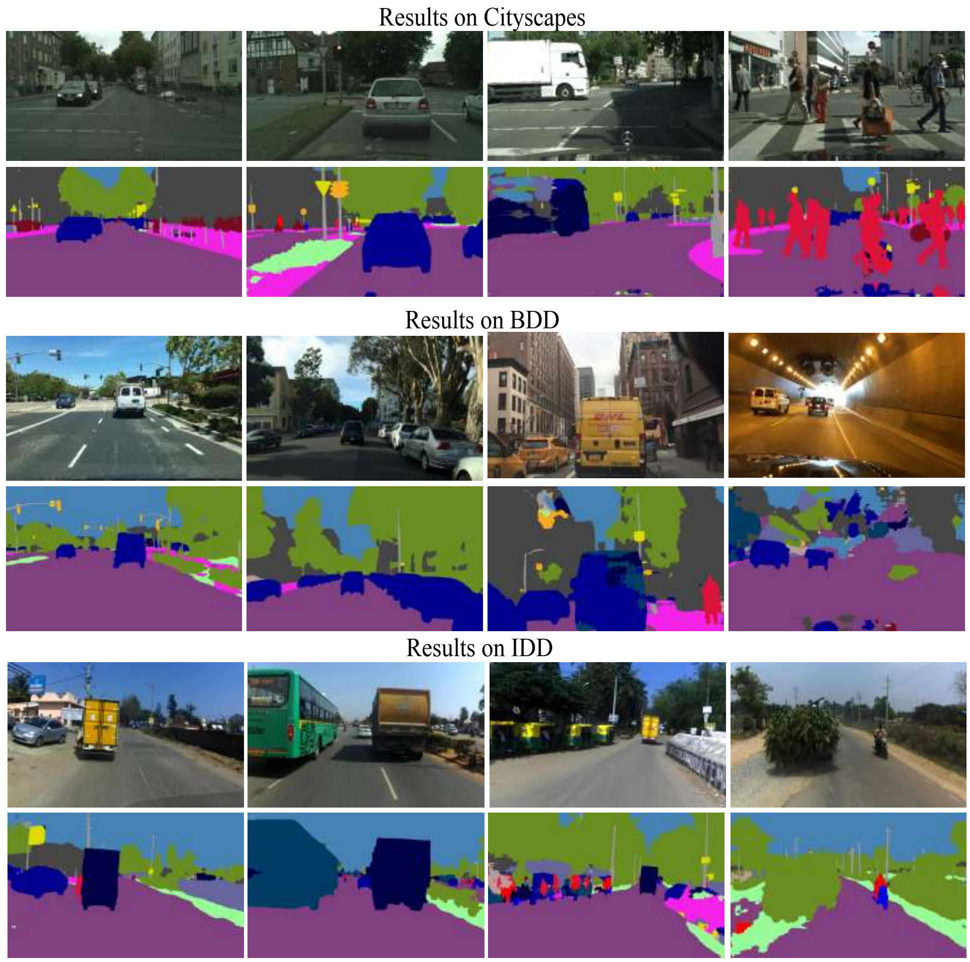


Figure 3.3: Some qualitative results of semantic segmentation on few selected frames of Cityscapes, BDD, and IDD datasets using DRN-D 38.(Best viewed in color and zoomed)

big buildings, and (iv) car moving under the tunnel. In this case, DRN segments well on first two images. However, due to dense adjacent buildings and lights in the tunnel, it produces few false segmentation for the third and fourth images. The selected images from IDD contain (i) a clean road with two cars, (ii) road with a bus and a truck, overtaking each other, (iii) road with dense autos and dense trees on the road's side, and (iv) road with one moving motorcycle. DRN performs reasonably well for all images except the third image where the performance drops due to dense vehicles and adjacent road side trees.

3.4.3 Instance Segmentation

3.4.3.1 Baselines:

We use three popular existing models — Mask R-CNN (MR), Cascade R-CNN (CR), and Mask Scoring R-CNN (MSR) to benchmark instance segmentation tasks on driving datasets.

3.4.3.2 Discussion:

Table 3.3 shows the class-wise AP_{box} and AP_{mask} scores on Cityscapes, BDD, and IDD datasets. We notice that for all object categories, Mask R-CNN obtains the best AP_box and AP_mask scores among all methods for Cityscapes and IDD. While Cascade Mask R-CNN obtains the best AP_{box} and AP_{mask} scores for majority of object classes (except *motorcycle* and *rider*) in case of BDD. For Cityscapes, Mask Scoring R-CNN produces the worse AP_{box} and AP_{mask} scores which are 9.5% and 11.4% lower than that of Mask R-CNN. For *motorcycle* and *train* object categories of Cityscapes, the performances of the two techniques — Cascade Mask R-CNN and Mask Scoring R-CNN drops more than 10% compared to Mask R-CNN. In BDD, we notice that for the object category car, all methods obtain AP_{box} and AP_{mask} scores more than 44%. We also notice that Cascade Mask R-CNN obtains the best average AP_{box} and AP_{mask} and Mask R-CNN obtains the worse results among the used methods. In case of IDD, we notice that the AP_{box} score is higher (more than 39%) for commonly found road objects such as *autorickshaw*, *truck*, *bus*, and *car*. While the AP_{mask} score for *person* and *rider* classes is very alike due to similar looking visual features. Among the object categories, all methods achieve the lowest AP_{box} and AP_{mask} for *vehicle-fallback*. MR obtains the best AP_{box} and AP_{mask} scores compared to other techniques. While MSR produces the worse average AP_{box} and AP_{mask} scores which are 4.8% and 4%, respectively less than MR.

From the table, we also observe that the performances of all methods are worse on BDD as compared to Cityscapes and IDD. This is because sequences of BDD are captured under several complex conditions. The quantitative results highlight that BDD is more complex than IDD and Cityscapes for instance segmentation. While IDD is more complex than Cityscapes for the same task.

Figure 3.4 shows the qualitative results of few randomly selected frames. Images of the first and second rows are from Cityscapes. Both images show multiple cars moving on the left and right sides of the road. From the figure, we notice that Mask R-CNN produces better results on overlapping cars. However, both Cascade Mask R-CNN and Mask Scoring R-CNN fail to segment instances of a car that are far away from the camera. The image in the third row shows a crowded traffic junction and multiple pedestrians are crossing the road in BDD. We notice that Cascade Mask R-CNN is able to segment instances of small pedestrians more accurately compared to Mask R-CNN and Mask Scoring R-CNN. Images of the fourth row presents multiple moving cars on a road, with shadows cast by road-side trees. In this case, Mask Scoring R-CNN obtains the best results. Images of fifth and sixth rows are taken from IDD, and include multiple overlapping vehicles of varying scales on a road with dense buildings on both sides. In both the cases, Mask R-CNN produces better results than Cascade Mask R-CNN and Mask scoring R-CNN.

3.5 Summary

In this work, we used various state-of-the-art models for object detection, semantic segmentation, and instance segmentation tasks and evaluate their characteristics on structured and unstructured driving

Table 3.3: Results on instance segmentation: Class-wise AP_{box} and AP_{mask} scores on Cityscapes, BDD, and IDD datasets. **MR, CM, MSR:** indicates Mask R-CNN, Cascaded Mask R-CNN, and Mask Scoring R-CNN, respectively. Values in bold indicates the best results among all the methods on respective datasets.

		Cityscapes			BDD			IDD		
Class	**Metric**	**MR**	**CM**	**MSR**	**MR**	**CM**	**MSR**	**MR**	**CM**	**MSR**
person	AP_{box}	**41.6**	32.3	33.8	31.0	**37.3**	32.7	**35.6**	31.2	29.1
	AP_{mask}	**34.0**	23.9	25.7	25.3	**32.7**	30.2	**31.5**	27.1	25.2
truck	AP_{box}	**35.2**	23.7	25.7	28.8	**34.0**	28.2	**54.1**	52.7	49.5
	AP_{mask}	**35.7**	23.6	26.8	27.9	**33.4**	27.9	**53.1**	50.3	49.6
motorcycle	AP_{box}	**29.5**	18.4	23.7	25.3	28.7	**28.9**	**39.9**	38.5	35.1
	AP_{mask}	**22.5**	12.6	15.5	15.5	16.8	**17.8**	**32.2**	30.6	28.2
rider	AP_{box}	**43.7**	36.9	37.6	21.0	20.5	**21.6**	**39.5**	37.4	34.3
	AP_{mask}	**29.2**	21.0	21.8	08.8	11.7	**11.8**	**29.4**	27.3	24.4
bus	AP_{box}	**60.1**	52.5	42.5	30.8	**35.5**	27.7	**49.5**	48.1	43.4
	AP_{mask}	**58.8**	49.5	42.0	30.0	**35.4**	28.9	**47.9**	45.3	43.8
bicycle	AP_{box}	**34.7**	25.9	30.4	11.5	**13.5**	10.7	**24.3**	20.5	20.4
	AP_{mask}	**22.9**	15.7	15.8	05.6	**08.3**	07.9	**14.3**	12.1	11.7
train	AP_{box}	**28.8**	14.7	9.5	0.0	0.0	0.0	-	-	-
	AP_{mask}	**42.1**	25.2	13.2	0.0	0.0	0.0	-	-	-
car	AP_{box}	**58.4**	50.8	52.6	48.0	**52.4**	47.6	**54.1**	51.9	48.6
	AP_{mask}	**52.5**	44.2	45.1	44.1	**48.6**	45.4	**50.2**	47.1	45.4
autorickshaw	AP_{box}	-	-	-	-	-	-	**55.5**	53.9	50.0
	AP_{mask}	-	-	-	-	-	-	**52.4**	48.9	47.1
vehicle-fallback	AP_{box}	-	-	-	-	-	-	**04.5**	03.8	02.8
	AP_{mask}	-	-	-	-	-	-	**03.9**	03.2	02.5
Average	AP_{box}	**41.5**	31.9	31.9	24.5	**27.7**	24.6	**39.7**	37.5	34.8
	AP_{mask}	**37.2**	26.9	25.7	19.7	**23.3**	21.2	**34.9**	32.4	30.9

datasets: Cityscapes, BDD, and IDD. To our knowledge, this work is the first comprehensive report on analyses of models for tasks with driving datasets. In the object detection task, all the used methods performed significantly better on object category *train* in Cityscapes than on BDD and IDD. Due to the unstructured nature, object detection tasks on IDD performed lower compared to Cityscapes and BDD. Cityscapes is the easiest dataset for object detection task among the three datasets being used. In semantic segmentation, we notice that all models perform better on Cityscapes than on BDD and IDD. We also notice that the DRN model performs consistently well across all the driving datasets, in comparison to other models. In instance segmentation, we observe that Mask R-CNN performs better than all other models on Cityscapes and IDD, while Cascade Mask R-CNN performs better for majority of the object categories of BDD. Looking at the complexity of dataset for different tasks, we notice

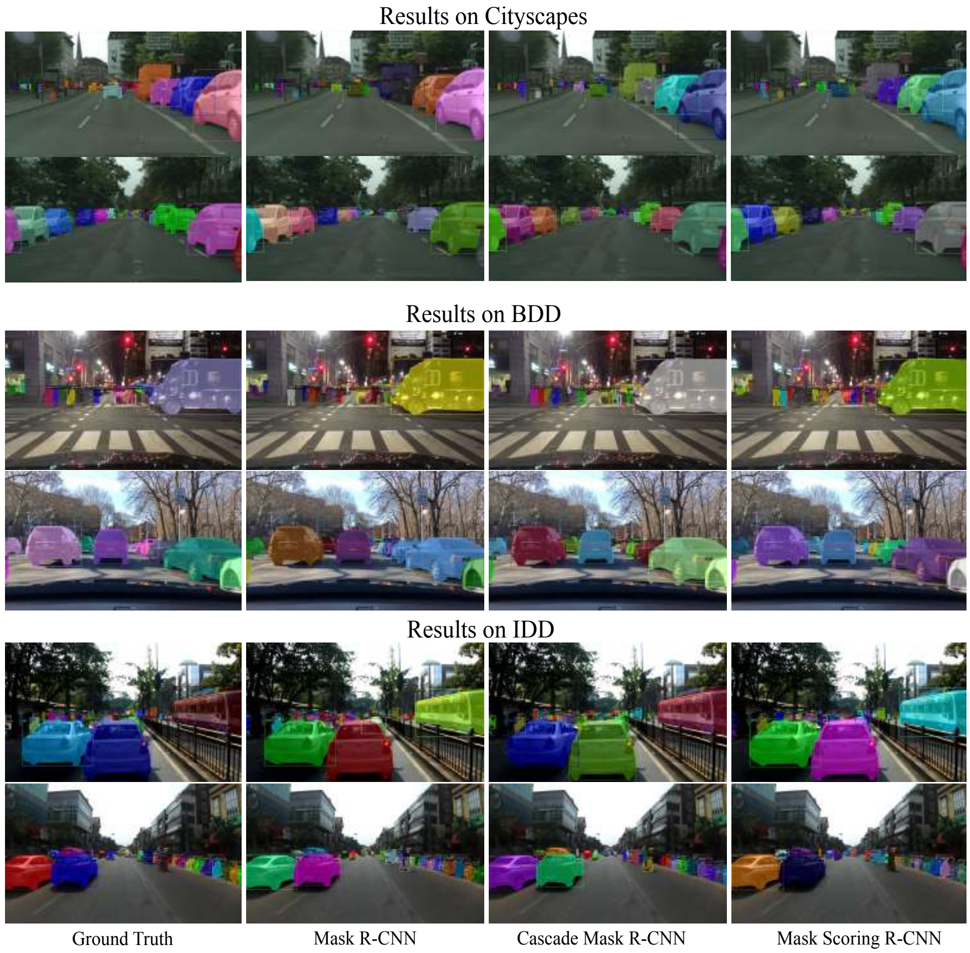


Figure 3.4: Some qualitative results on instance segmentation showing both mask and bounding boxes of few selected frames from Cityscapes, BDD, and IDD datasets.(Best viewed in color and zoomed)

that for instance segmentation and semantic segmentation tasks, BDD is a more complex dataset than Cityscapes and IDD.

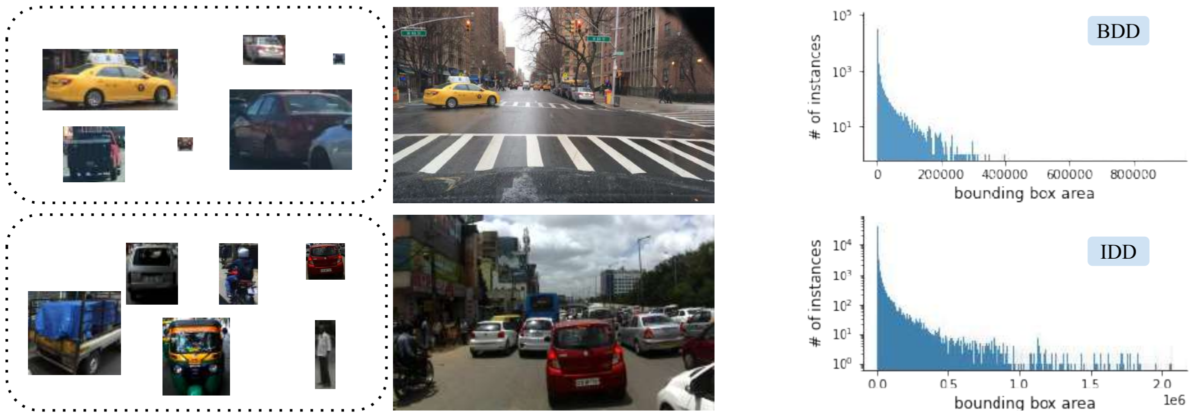
Chapter 4

Open World Object Detection on Road Scenes

Object detection is a crucial component in autonomous navigation systems. Current object detectors are trained and tested on a fixed number of *known* classes. However, in real-world or open-world settings, the test set may consist of objects of *unknown* classes; this results in the *unknown* objects being falsely detected as *known* objects leading to the failure in decision making of autonomous navigation systems. We propose Open World Object Detection on Road Scenes (ORDER) to resolve the aforementioned problem. We introduce Feature-Mix that widens the gap between known and unknown classes in latent feature space and improves the *unknown* object detection in the ORDER framework. We identify the inherent problems present in autonomous datasets: i) a significant proportion of the dataset comprises small objects and ii) intra-class bounding box scale variations. We address the problem of small object detection and intra-class bounding box variations by proposing a novel focal regression loss. Further, the detection of small objects is improved by curriculum learning. We present an extensive evaluation on two road scene datasets: BDD and IDD. Our experimental evaluations on BDD and IDD shows consistent improvement over the current state-of-the-art method. We believe that this work will lay the foundation for real-world object detection for road scenes.

4.1 Introduction

Building a robust autonomous navigation system that can reliably maneuver in a real-world scenario is a challenging task. Object detection [52, 10, 40, 49, 63, 15, 35] plays an important role for autonomous navigation systems in identifying and localizing objects in a road scene. Current object detection models are trained on a *closed-set*, where all the *test* classes are *known* at training time [55]. However, in an *open-world* setting the test set has objects from *unknown* categories. Therefore, present object detectors do not generalize well in an open world setting. Recently, Joseph *et al.* [27] introduced Open World Object Detector (ORE) framework that performs open-world object detection. The performance of ORE was evaluated on generic dataset: MS-COCO [37] and PASCAL-VOC [14]. However, we find that the ORE framework shows poor performance when applied to challenging domains, such as road scene datasets. The challenges include: a) *unknown* objects are hard to detect; b) the proportion



(a) (b)

Figure 4.1: **(a)**: We can observe intra-class and inter-class scale variation prominently in some categories like car and pedestrian. This issue is prominent in road scene datasets.**(b)**: Shows the distribution of bounding box area in BDD and IDD, we notice that there are relatively more small bounding boxes than large bounding boxes.

of small objects (from both *known* and *unknown* set) is significant (fig. 4.1 [b]), and c) the presence of intra-class scale variation (fig. 4.1 [a]). The problem of intra-class scale variation is highly pronounced found in road scene datasets. Generic datasets such as MS-COCO and PASCAL-VOC consist of images captured close to the object resulting in smaller variations in scale. Similarly, in aerial object detection dataset [66], the objects are captured at distinctly high altitudes resulting in a consistent intra-class object size.

We propose Open World Object Detection on Road Scenes (ORDER) that addresses the aforementioned problems. We introduce Feature-Mix, inspired by Open-Mix [74] where we combine multiple *unknown* and *known* class instances to improve *unknown* object identification. It is important to note that Open-Mix takes a single instance of known and unknown, hence, it cannot combine multiple *unknown* and *known* class instances that are generally present in road scenes. Feature-Mix overcomes the limitation of Open-Mix by mixing *unknown* and *known* class instances at the feature level, allowing it to mix multiple instances of *known* and *unknown* classes. Next, we propose focal regression loss that handles intra-class variation by including bounding box area and improves small object detection by penalizing more to small bounding boxes than large ones. We further improve small object detection by training the ORDER framework in a curriculum manner. Improving the small object detection and handling the intra-class variation reduces the chances of *known* class detected as *unknown* and improves the *known* object detection. We validate the performance of ORDER and the competitive baselines on the Indian Driving Dataset (IDD) [60] and Berkeley Deep Drive (BDD) [69] datasets. We observe that the ORDER shows state-of-the-art performance on open-world evaluation metrics: Wilderness Impact (WI) and Absolute Open-Set Error (A-OSE). Additionally, an extensive ablation study is also performed on ORDER to show the contribution of proposed components individually.

The key contributions of our work are:

- To the best of our knowledge, ORDER is the first work that addresses open-world object detection for road scene datasets.
- Introduces Feature-Mix, which is integrated in ORDER that significantly improves the feature representation of *unknowns*.
- Identifies and addresses two inherent issues in road scene datasets: intra-class scale variation by proposing focal regression loss and small object detection by curriculum learning.

4.2 Related Works

4.2.1 Object Detection:

The goal of an object detection model is to predict the bounding boxes and the class for an object present in an image. The current set of detectors can be divided into two categories: single-stage detectors and two-stage detectors. Two-stage detectors such as Faster R-CNN [52], R-FCN [10] depend on Region Proposal Network (RPN) that generates the region proposals based on an objectness score, which are further refined to get desired object bounding box. Whereas single-stage detectors such as SSD [40], YOLO [49], and SqueezeDet [63] consist of a single network that can predict the object bounding boxes and classes. There are also several works of object detection [38, 63, 23] on road scene datasets. We choose the two-stage detector over the single-stage detector since the former demonstrates better performance.

4.2.2 Open Set Detection:

Object detectors trained on fixed set of training and testing classes are not robust in identifying unknown encountered in real-world. Miller *et al.* [44] introduced the open-set object detection for real-world scenarios. They utilized dropout sampling to get the uncertainty present in the object detector and used it to identify *unknown* objects. Next, Miller *et al.* [43] deployed various merging strategies for Monte Carlo (MC) dropout on object detector and evaluated in open-set conditions. Recently, Dhamija *et al.* [13] provided a detailed analysis of object detector performance in the open-set setting and proposed an evaluation metric Wilderness Impact that quantifies the performance of object detection model in real-world.

4.2.3 Open World Detection:

Joseph *et al.*[27] introduced the problem of *open-world* object detection and proposed Open World Object Detector (ORE). The key idea of ORE is to identify *unknown* classes and incrementally learn the distinct unknown classes when the labels of those classes are available. ORE uses Faster R-CNN as a base detector and improves its ability to identify *unknown* classes by adding contrastive clustering

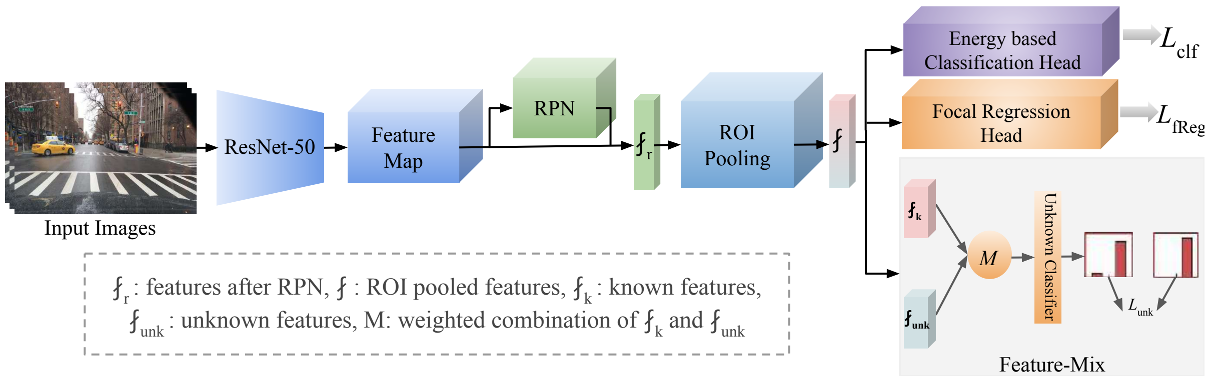


Figure 4.2: Illustration of ORDER framework. f is RoI pooled features consisting of *known* class features f_k and *unknown* class features f_{unk} that are mixed in Feature-Mix *M* block. L_{clf}, L_{fReg} and L_{unk} denotes the classification loss, focal regression loss, and feature-mix loss, respectively.

and an energy-based classifier. However, the ORE is not designed to handle intra-class scale variation explicitly, which is prominently present in road scene datasets. Our proposed framework addresses the issue of scale variation by introducing curriculum-based training along with a novel *Focal Regression Loss* (section 4.3.4). We also improve the feature representation of *unknown* by introducing *Feature-Mix* (see sections 4.3.3 and 4.5.5).

4.3 Methodology

4.3.1 Problem Setting

We begin the formulation of Open World Object Detection (OWOD) by considering a set of *known* classes as C_k and a set of *unknown* classes C_{unk}. The *known* classes will have ground-truth bounding boxes, whereas *unknown* classes will be encountered at test time. We train a detection model D on a set of *known* classes and simultaneously ensure that *unknown* class instances are also detected. Next, the detected set of *unknown* class instances could be forwarded to a human annotator to obtain the ground-truth labels. The detection model is incrementally trained on the new ground-truth labels rather than training the entire model from scratch giving an improved detection model $\hat{D}$. We continue the process of the detection model adapting to a new set of classes and detecting *unknown* class objects over a lifetime. In the experimental setting, we define the set of classes as task T.

4.3.2 ORDER: Open World Object Detection on Road Scenes

ORDER uses Faster-RCNN [21] detector that is molded accordingly to detect *known* and *unknown* objects. It consists of three output heads: Energy Based Classification head, Focal Regression head, and Feature-Mix head. Energy Based Classification head and Focal Regression head are used to learn to differentiate between *known* and *unknown* features by using L_{clf} and L_{unk}. Focal Regression head learns the object bounding boxes. Figure 4.2 shows the pipeline of the ORDER framework. The ORDER framework is inspired from ORE [27], however, it largely differs in terms of novel components that we introduced to handle the challenges present in road scene datasets. We will discuss the key novelties of ORDER framework in detail.

4.3.3 Feature-Mix

Unknown class identification is an essential component of an open-world object detector. In autonomous navigation systems, improving *unknown* class identification will increase the chances of avoiding unfavorable situations. We propose a Feature-Mix approach that improves *unknown* class identification by incorporating knowledge of *known* classes. The key intuition behind Feature-Mix is to mix the features of *known* and *unknown* and suppress the activation caused by known features, so that the latent difference between *known* and *unknown* feature is maximized. We begin Feature-Mix formulation by taking Region of Interest (RoI) pooling output features f consisting of *known* class features f_k and *unknown* class features f_{unk}. We mix the known and *unknown* features by:

$$f_{mix} = \lambda f_k + (1 - \lambda) f_{unk}, \tag{4.1}$$

where, λ is sampled from *beta distribution* parameterized with α and β. Now, the *unknown* classifier C_{unk} utilizes f_{mix} to identify *unknown* objects trained by using a loss L_{unk} given by:

$$L_{unk} = -y log(softmax(C_{unk}(f_{mix}))), \tag{4.2}$$

$$y = argmax(log(softmax(C_{unk}(f_{mix})))) \tag{4.3}$$

y represents the ground-truth label. We use a small held-out validation set, as proposed in ORE [27], consisting of known and unknown data samples to train Feature-Mix.

4.3.4 Focal Regression Loss

Popular detection methods use Smooth-L1 [52], and Generalized Intersection over Union (GIoU) [53] loss for bounding box regression. However, these losses do not explicitly incorporate the knowledge of intra-class scale variation prominent in road scene datasets. We introduce Focal Regression Loss (L_{fReg}) that addresses the problem of a) detecting small objects by penalizing more for small bounding boxes b) intra-class variation by including bounding box information. We formulate L_{fReg} by adding a regulating component $(1 - IoU)^{\gamma^*}$ to squared IoU loss, where $\gamma^* \in [0, \infty)$ is a focusing parameter. L_{fReg} can be denoted as:

$$L_{fReg} = (1 - IoU)^{\gamma^*} \|1 - IoU\|_2^2 \tag{4.4}$$

$$\gamma^* = \gamma + log(log(\hat{Ar}_{bbox_{gt}})), \hat{Ar}_{bbox_{gt}} = Ar_{Img}/Ar_{bbox_{gt}} \tag{4.5}$$

$\hat{Ar}_{bbox_{gt}}$ and $Ar_{bbox_{gt}}$ represents the inverse-normalized and unnormalized bounding box area, respectively. Inverse-normalization gives large values for small bounding boxes and small value for large bounding boxes. .$\hat{Ar}_{bbox_{gt}}$ results from dividing image area Ar_{Img} by $Ar_{bbox_{gt}}$. γ^* consists of tunable parameter γ and double logarithmic of inverse-normalized bounding box area. We apply double logarithmic to inverse-normalized bounding box area because a) it prevents overshooting of γ^* when inverse-normalized bounding box area is tiny, b) it smoothens out the variation in the inverse-normalized bounding box area making the training more stable. Note that for small bounding boxes, the value of γ^* would be high, resulting in more penalization as compared to large bounding boxes that are easy to detect.

4.3.5 Curriculum Training

Detecting smaller objects [25] is a harder task than detecting object instances with larger sizes. In autonomous navigation datasets such as BDD and IDD, the proportion of small objects is significant fig. 4.1. As per [58], smaller objects are considered harder to learn than larger objects. Hence, we adopt a curriculum learning [3, 19] strategy to gradually train the network from *easy* samples (large objects) to *hard* samples (small objects). We divide the training dataset into three sets: S_{easy}, S_{medium}, and S_{hard}, based on the bounding box area. For an individual task T_i, $i \in \{1, 2, 3\}$, we train the detection model in three steps that can be formulated as:

$$T_i = \begin{cases} S_{easy} & I_1;\ \textit{if } Ar_{bbox} < Ar_{easy} \\ S_{easy} + S_{medium} & I_2;\ \textit{if } Ar_{bbox} < Ar_{medium} \\ S_{easy} + S_{medium} + S_{hard} & I_3 \end{cases} \tag{4.6}$$

I_1, I_2, and I_3 are the number of iterations each group of the sets are trained. Ar_{easy} and Ar_{medium} are the area thresholds for selecting large and medium bounding boxes. Ar_{easy} is below 32^2 px. Ar_{medium} is below 96^2 px. Ar_{hard} is above 96^2 px.

4.4 Experiments and Results

4.4.1 Datasets Protocol

We adapt the standard evaluation protocol of ORE [27] to demonstrate the efficacy of ORDER. For a given dataset, we divide it into a set of classes. Each class set is denoted by task T_t, t represents the time-stamp of the model having access to only classes of T_t. The dataset can be represented as $\{T = T_1, \cdots T_t, \cdots\}$. At a given time-stamp t, the classes of $\{T_\tau : \tau \leq t\}$ are considered as *knowns* and the classes of $\{T_\tau : \tau > t\}$ as *unknown*s. We follow the protocol discussed above discussed to divide the IDD and BDD datasets into tasks.

The IDD dataset consists of 15 classes. We divide the dataset into three tasks, and each task consists of 5 classes. The BDD dataset consists of 10 classes. We divide the dataset into three tasks; the first task consists of 4 classes, and the rest have 3. For each task, we randomly choose the classes to avoid any bias. The statistics of training and testing instances and the classes for each task are given in table 4.1. We take a set of $3K$ images from each dataset for validation.

4.4.2 Evaluation Metrics

We use mean Average Precision (mAP) to evaluate the performance of the model on *known* classes. The IoU threshold for the mAP is taken as 0.5 in accordance with [57, 47, 27]. Now, to quantify the performance of a model for *unknown* identification, we use Wilderness Impact (WI) [13] metric. The WI measures the model's sensitivity to *unknowns* over a range of frequency of frames that may have *unknowns*. The WI is equated as:

$$\text{Wilderness Impact } (WI) = \frac{P_{\mathcal{K}}}{P_{\mathcal{K}\cup\mathcal{U}}} - 1$$

Here, $P_{\mathcal{K}}$ refers to the precision of the model when evaluated on *known* classes, and $P_{\mathcal{K}\cup\mathcal{U}}$ is the precision when evaluated on *known* and *unknown* classes, measured at a recall level(R) of 0.8 in all experiments. Ideally, the WI needs to be close to 0, demonstrating that the precision does not change when *unknowns* are introduced to the test set. Absolute Open-Set Error (A-OSE) [44] is another metric that shows the *unknown* detection performance of a model. It is defined as total number of *unknown* objects getting classified as a *known* object.

4.4.3 Implementation Details

We use the modified Faster R-CNN with ResNet-50 [21] backbone according to ORE. The shape parameters α and β are chosen to be 1. The contribution of L_{unk} in total loss is 0.001 and 0.1 for IDD and BDD, respectively. The values of hyperparameter γ present in Focal Regression Loss is 0.4 and 0.1 for IDD and BDD, respectively. For the Curriculum training, I_1, I_2, and I_3 values are $36K$ for Ar_{easy} and $72K$ for Ar_{medium} and Ar_{easy} on both IDD and BDD datasets. We train our models on 4 GPUs with a batch size of 8 images.

4.4.4 Results on BDD

We now discuss the results of our experiments on the BDD dataset. As a baseline, we train Faster-RCNN on the first task and finetune it on consecutive tasks as shown in the first-row of table 4.2 (top). The ORE reduces both WI and A-OSE (lower the better) compared to baseline for the first two tasks[1] of BDD. However, ORE drops in overall mAP by 2 (approx.) compared to baseline for the two tasks (columns 4 and 9 of table 4.2 [BDD]). On the contrary, ORDER improves mAP by 0.5 and 1.4 for the two tasks, reducing WI by 0.015 and 0.013 compared to the baseline. ORDER also reduces the AOSE by a considerable margin of 9769 and 11385 as compared to the baseline. For Task 3 of BDD, ORDER

[1]Note that all the classes are known for Task 3; hence, the two metrics do not hold.

Table 4.1: Table shows the division of the datasets into various tasks. For each task the group of classes, training and testing statistics are shown.

	IDD			BDD		
	Task 1	Task 2	Task 3	Task 1	Task 2	Task 3
Categories	person, bicycle, train, truck, traffic sign	traffic light, bus, car, rider, trailer	motorcycle, caravan, animal, autorickshaw, vehicle fallback	pedestrians, motorcycle, traffic sign, bus	bicycle, train, traffic light	car, rider, truck
# training images	23217	25586	25146	61208	40416	69103
# test images	7558	8462	8085	8756	5803	9898
# train instances	103614	155404	122498	345639	195123	734030
# test instances	30025	55204	40831	50270	27938	107825
# unknown instances	96035	40831	0	136763	107825	0

Table 4.2: Quantitative performance of ORDER on road scene datasets. We notice that ORDER shows good performance in identifying unknown classes by giving lower Wilderness Impact and Average Open Set Error and simultaneously performs well in detecting known classes by giving high mean Average Precision. Best results are highlighted in bold.

BDD											
Task IDs (→)	Task 1			Task 2					Task 3		
	WI	A-OSE	mAP (↑)	WI	A-OSE	mAP (↑)			mAP (↑)		
	(↓)	(↓)	Current known	(↓)	(↓)	Previously known	Current known	Both	Previously known	Current known	Both
Faster-RCNN [52] + Finetuning	0.04563	12628	46.01	0.02351	14738	42.86	18.31	32.34	28.38	37.96	31.26
ORE [27]	0.03244	6186	44.43	0.01807	5028	37.54	18.65	29.44	27.80	40.70	31.67
ORDER	**0.02994**	**2859**	**46.50**	**0.00983**	**3353**	40.65	24.89	**33.90**	34.35	45.25	**37.62**

IDD											
Task IDs (→)	Task 1			Task 2					Task 3		
	WI	A-OSE	mAP (↑)	WI	A-OSE	mAP (↑)			mAP (↑)		
	(↓)	(↓)	Current known	(↓)	(↓)	Previously known	Current known	Both	Previously known	Current known	Both
Faster-RCNN [52] + Finetuning	**0.09559**	21539	**35.79**	0.06279	21134	21.25	27.79	24.52	23.84	23.48	23.72
ORE [27]	0.10702	13149	35.01	**0.05999**	13507	18.17	26.49	22.33	25.76	22.04	24.52
ORDER	0.09984	**10353**	35.20	0.06460	**10879**	20.13	29.88	**25.01**	25.08	24.48	**24.88**

attains a massive gain in overall mAP of around 6.36 and 5.95, compared to the baseline and ORE (last column of table 4.2 [BDD]).

4.4.5 Results on IDD

On the IDD dataset, we observe in table 4.2 that the WI is comparable for the three models. ORDER, however, achieves the best A-OSE for the first two tasks of IDD, reducing it by a margin of 11186 and 10255 compared to baseline and 2796 and 2628 compared to ORE. ORDER's overall mAP is comparable to ORE for Task 1 of IDD and is highest for the remaining tasks (columns 9 and 12 of table 4.2 [IDD]).

It is also interesting to note that ORDER performs better than ORE for all the columns in table 4.2 (refer to last two rows of the tables).

Table 4.3: Ablation study of proposed components in ORDER on Task 1 of IDD. Best results are highlighted in bold.

Regression Loss	Feature Mix	Curriculum Training	WI	A-OSE	mAP
Smooth-L1 [52]	✗	✗	0.10702	13149	35.01
Focal Regression	✗	✗	0.11021	13084	**36.58**
Focal Regression	✓	✗	0.10996	10563	33.90
Focal Regression	✓	✓	**0.09984**	**10353**	35.20

4.5 Discussion and Analysis

4.5.1 Ablative Study of ORDER

We perform ablative studies to validate the performance of the proposed components of ORDER qualitatively. Table 4.3 shows the results on Task 1 of IDD. We observe that using all the proposed components shows significant improvement on WI, A-OSE, and mAP over the model trained with only Smooth-L1 loss (row 1 of table 4.3). It is also essential to infer from the first two rows of table 4.3 that the proposed focal regression loss shows significant improvement in mAP compared to Smooth-L1.

4.5.2 Performance Comparison of Focal Regression Loss

We demonstrate the efficacy of our proposed focal regression loss in better identifying *known* objects. We compare the proposed loss with Smooth-L1 [52], GIoU [53], and Least Square IoU [45]. Table 4.4

Table 4.4: (a) Performance of ORDER when trained on various bounding box losses. (b) Sensitivity analysis of Feature-Mix loss contribution. All the experiments are conducted on the Task 1 of IDD. Best results are highlighted in bold.

(a)

Loss	mAP
Smooth-L1 [52]	34.01
GIoU [53]	32.53
Least Square IoU [45]	32.44
Focal Regression (Ours)	**35.20**

(b)

Loss weight	WI	A-OSE	mAP
1	0.10153	10088	35.13
0.1	0.10169	**10069**	35.12
0.01	0.10108	10145	35.14
0.001	**0.09985**	10353	**35.20**

(a) shows the mAP on all the losses trained on Task 1 of IDD. We find that *Focal Regression Loss* gives the best performance among all the losses.

4.5.3 Sensitivity Analysis of Feature-Mix:

We show the variation in the performance of ORDER by changing the contribution of the feature-mix in the total loss. Table 4.4 (b) shows the performance of ORDER on Task 1 of IDD having various loss weights denoting the fraction of feature-mix loss contributed towards total loss. We find that tuning the feature-mix weights to 0.001 gives the best performance on almost all the evaluation metrics.

4.5.4 Qualitative Results

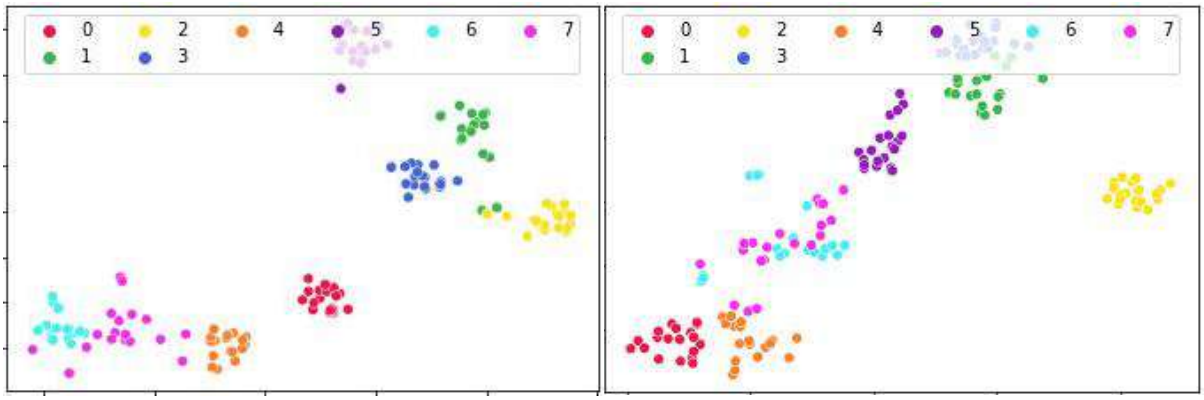
Qualitative results demonstrating the ORDER's capability to i) handle intra-class scale variations, ii) detect small objects, and iii) discriminate *knowns* from *unknowns* can be seen in fig. 4.3. We show the sample results of the model trained on task 2 of IDD and BDD datasets. As shown, ORDER performs better than ORE for the three different cases. The key observations are that ORE misses several known objects (especially cars in IDD and pedestrians in BDD) and demonstrates confusion among detected unknown and known objects (especially traffic signs in BDD). On the contrary, ORDER performs considerably better for such cases with high confidence.

4.5.5 Latent Feature Visualization

We show the visualization of latent features of ORDER and ORE. These features are obtained after RoI pooling from the model trained on BDD Task 2 and then visualized using t-SNE [59]. Figure 4.4 shows the clusters formed by latent features belonging to various classes. 7 represents the *unknown* class and the rest as *known* class labels. We can observe the features cluster of ORDER have better

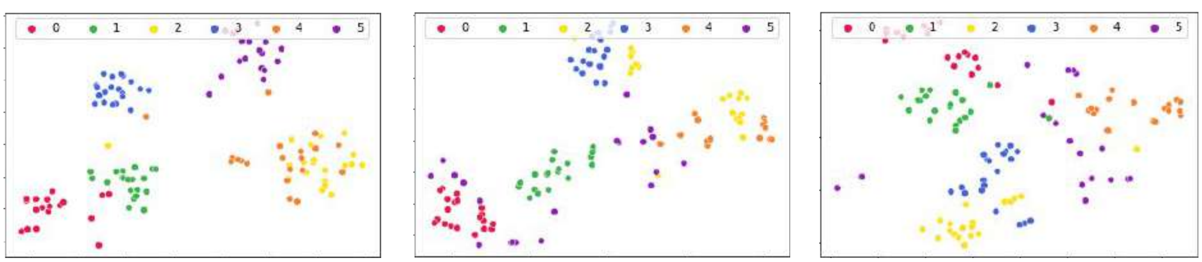
Figure 4.3: Qualitative comparison: Row a) images are from the IDD dataset, and b) and c) are from the BDD dataset. The results are inferred from the models trained on Task 2 of the BDD and IDD datasets. In row a), we observe that ORDER is able to detect smaller objects with high confidence. It is interesting to note that the highlighted boxes of a) has `car` instances shows intra-class scale variation. ORDER handles the intra-class scale variation within the `car` instance by detecting it on varying scales. In b) and c), we can see ORDER detects safety-critical classes such as `pedestrian` and `traffic sign` better than ORE. We also notice that ORDER performs better at recognizing overlapping *known* and *unknown* objects and has high confidence in unknown and known predictions. For easy distinction, the redred bounding boxes denote *unknown* predictions, whereas the greengreen bounding boxes denote the distinct *known* classes. The blueblue and purplepink boxes represents the cropped region. **Best viewed when zoomed**.

quality compared to ORE and can better differentiate *unknown* class feature label (7) with the known class feature (6).



ORDER ORE [27]

Figure 4.4: We show the t-SNE plots of latent features of ORDER and ORE on Task 2 of BDD. Class label 7 denotes the *unknown* class, and the remaining classes are *known*. We can see that ORDER clearly separates class 7 with 6, whereas in ORE, these classes are intertwined. We can also notice that the separability between the smaller objects such as 0 (*traffic sign*) and 4 (*traffic light*) is better in ORDER.



Ours ORE [27] Faster-RCNN+FT

Figure 4.5: We show the t-SNE plots of latent features of our method, ORE [27], and the baseline Faster-RCNN+FT(Fine-Tuned) on Task 1 of IDD. Class label 5 denotes the *unknown* class, and the remaining classes are *known*. Our method has tighter clusters that are well separated. Due to the introduction of Feature-Mix, we can notice distinguishable *unknown* cluster which is well separated from *known* classes.

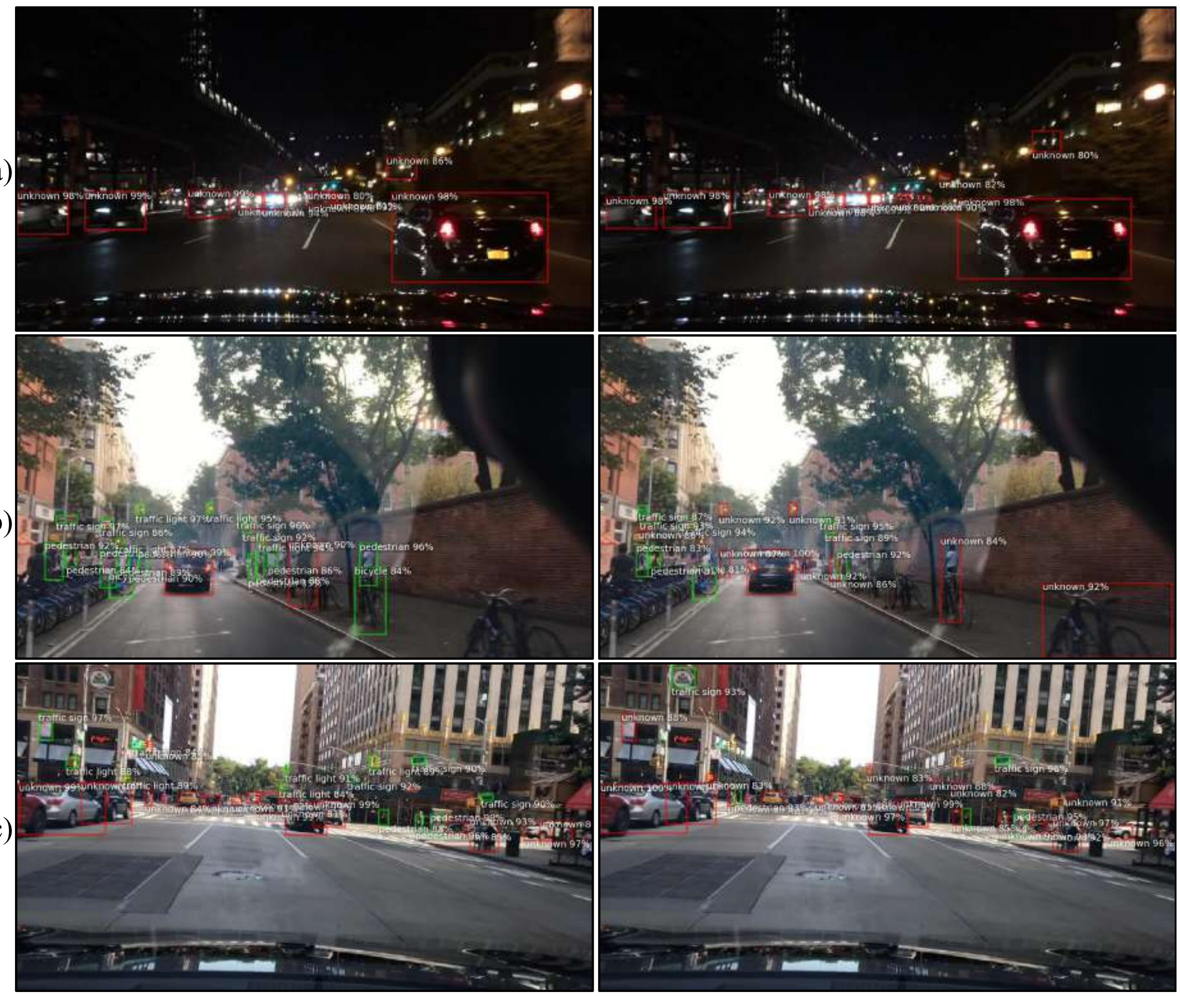
4.6 Quantitative results on incremental learning and extreme conditions

The objects which are present in a high dense, and far way places produce higher false positives in both the models. This issue is dominant in traffic signs, since many symbols are have very similar spatial appearance and vary very slightly in semantics. Figure 4.6 shows the extreme cases where the confusion is high between ORDER and ORE framework.

a)

b)

c)



ORDER ORE [27]

Figure 4.6: Each row represents the qualitative results on ORDER and ORE framework. In row b) we can notice that the `bicycle` class is not being detected by ORDER whereas it is being detected as *unknown* by ORE, which is a mis-detection by both the frameworks. Rows a) and c) shows better *known* and *unknown* detections by ORE.

In open-world object detection, a major aspect is to incrementally learn new classes as the model is exposed on unknowns classes during deployment. Figure 4.7 shows how a model is incrementally learnt on IDD dataset on 3 tasks. The object instances which are unknown in Task 1 are highlighted in red, when the model learns them incrementally, those classes do not get detected as unknown and are highlighted in green

Figure 4.7: The images in row 1,2, and 3 show the result of ORDER on Task 1,2, and 3 respectively. We can notice that the `rider` and `motorcycle` class which are *unknown* in the results of Task 1, are subsequently learnt in Task 2 and Task 3. **Best viewed when zoomed**.

4.7 Summary

In this chapter, we introduced the ORDER framework which is designed to handle *Open World Object Detection* challenges on road scene datasets. We demonstrate that ORDER outperforms the current state-of-art open world detector. The key contribution consists of *Feature-Mix* that improves the *unknown* object identification. Further, we handle the intra-class scale variation and small object detection by proposed *Focal Regression Loss* and curriculum learning. Currently, ORDER trains on the tasks that belong to a single road scene dataset. In the future work, we plan to extend ORDER to be trainable on tasks that belong to multiple road scene datasets captured in different geographic locations. We hope this work will open doors for further research to make vision models more robust in real-world scenarios, resulting in safer and reliable autonomous navigation systems.

Chapter 5

Conclusions and Future Directions

In this thesis, we highlighted the problems caused in unstructured driving scenarios and how a data-centric approach can help in addressing the issues and build a robust model which can be deployed in real-world scenarios. To begin with, we focused on understanding the behavior of popular computer vision tasks on road scene datasets, after understanding some key problems inherently present in road scene datasets, we formulate open-world object detection focused on driving scenarios. We began the dissertation by introducing popular deep learning models used in object detection, semantic segmentation, and instance segmentation; popular road scene datasets currently available for the mentioned tasks; the motivation and need for evaluating and benchmarking the models across datasets; road safety concerns with respect to autonomous driving scenarios. We then provide some background on open-world detection, we explained the key evaluation metrics which have been proposed to include the openness into evaluation, we also showed how open-world differs from close-world approach of modeling any computer vision problem, we also showed which popular vision datasets are available for open-world modeling.

Next, we dived into showing the quantitative and qualitative results on popular object detection, semantic segmentation, and instance segmentation models on road scene datasets. Later, we focused on modeling open-world object detection on road scenes. We identified two key challenges present in road scene datasets; intra-class size variation and skewed distribution of small objects. We address these issues by proposing a novel loss called focal-regression to address the former, and curriculum learning to address the latter.

We formulated a framework which is designed to handle *Open World Object Detection* challenges on road scene datasets. The key contribution consists of improving the *unknown* object identification. Further, we handle the intra-class scale variation and small object detection by proposing a novel loss, we also suggest an alternate training strategy using curriculum learning. This work also highlights the importance of addressing the issues related with domain gap, which can affect models which are aimed at domain generalization on Indian road scenes. Addressing the inherent key issues like class imbalance, tiny object detection can be the initial future steps.

Every dataset has some inherent bias which induces unintentional problems. The unstructured road scenario of Indian roads gives rise to various problems out of which, annotating the dataset is of prime concern. If we can leverage the temporal knowledge available in video scenes, we can create models which can not only improve the necessary metrics of a model but also extend the model on multi-task learning where behavior and intent predictions can also be performed.

Rejection capability of unknown classes can make the system more reliable as the unknown instances which the model is subjected to comes from an infinite set of object possibilities. Training on multiple domain across varying geographic location is still an open research problem. A self-supervised approach to identify the hidden classes present in the identified unknown classes is a promising area to push the boundaries of open-world object detection.

Related Publications

Conference:

1. **Deepak Kumar Singh**, Ameet Rahane, Ajoy Mondal, Anbumani Subramanian, C. V. Jawahar. **Evaluation of Detection and Segmentation Tasks on Driving Datasets.** *In The 6th International Conference on Computer Vision and Image Processing (CVIP), 2021*

2. **Deepak Kumar Singh**, Shyam Nandan Rai, Joseph K J, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, C. V. Jawahar. **ORDER: Open World Object Detection on Road Scenes.** *In The NeurIPS 2021 Workshop on Machine Learning for Autonomous Driving (ML4AD)*

3. **Deepak Kumar Singh**, Shyam Nandan Rai, Joseph K J, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, C. V. Jawahar. **New Objects on the Road? No Problem, We'll Learn Them Too** *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*

Bibliography

[1] R. Araki, T. Onishi, T. Hirakawa, T. Yamashita, and H. Fujiyoshi. MT-DSSD: Deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection. In *ICRA*, 2020.

[2] A. Bendale and T. Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015.

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[4] D. Bolya, C. Zhou, F. Xiao, and Y. Lee. YOLACT: Real-time instance segmentation. In *ICCV*, 2019.

[5] Z. Cai and N. Vasconcelos. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. on PAMI*, 2019.

[6] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv*, 2019.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. on PAMI*, 2018.

[8] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016.

[9] 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.

[10] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[13] A. Dhamija, M. Gunther, J. Ventura, and T. Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1021–1030, 2020.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[15] G. Ghiasi, T.-Y. Lin, and Q. V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2019.

[16] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[19] G. Hacohen and D. Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR, 2019.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *CVPR*, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang. Mask scoring R-CNN. In *CVPR*, 2019.

[23] D.-T. Iancu, A. Sorici, and A. M. Florea. Object detection in autonomous driving - from large to small datasets. In *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6, 2019.

[24] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer, 2014.

[25] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu. A survey of deep learning-based object detection. *IEEE access*, 7:128837–128868, 2019.

[26] M. Jones and P. Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3(14):2, 2003.

[27] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian. Towards Open World Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5830–5840, 2021.

[28] P. R. M. Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017.

[29] L. Ke, Y.-W. Tai, and C.-K. Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *CVPR*, 2021.

[30] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[31] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[32] Y. LeCun, C. Cortes, and C. Burges. MNIST handwritten digit database. *ATT Labs*, 2010.

[33] Y. Lee and J. Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020.

[34] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan. Proposal-free network for instance-level object segmentation. *IEEE trans. on PAMI*, 2017.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[38] G. Liu. *Real-Time Object Detection for Autonomous Driving Based on Deep Learning*. PhD thesis, Texas A&M University-Corpus Christi, 2017.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2015.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[42] V. Lonij, A. Rawat, and M.-I. Nicolae. Open-world visual recognition using knowledge graphs. *arXiv preprint arXiv:1708.08310*, 2017.

[43] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2348–2354. IEEE, 2019.

[44] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

[45] X. Ming, F. Wei, T. Zhang, D. Chen, and F. Wen. Group sampling for scale invariant face detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3456, 2019.

[46] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[47] C. Peng, K. Zhao, and B. C. Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109–115, 2020.

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[50] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, 2017.

[51] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018.

[52] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[53] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[54] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. ERFNet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. on Intelligent Transportation Systems*, 2018.

[55] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[56] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014.

[57] K. Shmelkov, C. Schmid, and K. Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017.

[58] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166, 2021.

[59] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[60] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019.

[61] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li. SOLO: Segmenting objects by locations. In *ECCV*, 2020.

[62] Y. Wang, Q. Zhou, J. Xiong, X. Wu, and X. Jin. ESNet: An efficient symmetric network for real-time semantic segmentation. In *PRCV*, 2019.

[63] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 129–137, 2017.

[64] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang. CGNet: A light-weight context guided network for semantic segmentation. *IEEE Trans. on Image Processing*, 2020.

[65] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[66] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018.

[67] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[68] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020.

[69] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. BDD100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[70] F. Yu, V. Koltun, and T. A. Funkhouser. Dilated residual networks. *arXiv*, 2017.

[71] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018.

[72] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[73] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling. M2Det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, 2019.

[74] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.