



# Image tagging and Road Object Detection

26.02.2023

---

## Group 4

Mentor - Sangeeth

Arnold

Arpita Das

Pramod Kadangotte

## Contents

[Problem Statement](#)

[Objectives](#)

[Dataset Link and Description](#)

[Methodologies](#)

[Traditional Machine Learning Methods for Image Tagging and Road Object Detection](#)

[Deep Learning Methods for Image Tagging and Road Object Detection](#)

[Methodologies for Multiple Object Tracking System](#)

[Design Considerations](#)

[Tools](#)

[Deployments](#)

[Data Preprocessing](#)

[Model Selection](#)

[Tracking System Selection](#)

[Evaluation Metrics](#)

[Ethical Considerations](#)

[Outcomes Achieved in Stages](#)

[Data Collection](#)

[Select a Model for Object Detection](#)

[Preprocessing](#)

[Model Training on BDD10K Data](#)

[Testing and validation on BDD10K Data](#)

[Model Training on BDD100K Data](#)

[Testing and validation on BDD100K Data](#)

[Model Statistics](#)

[Select an Algorithm for Object Tracking](#)

[Object Tracking](#)

[Evaluation of the Tracker](#)

[Deployment](#)

[Challenges](#)

[Real World Applications](#)

[References](#)

[Thank You](#)

## Problem Statement

Image tagging and road object detection are two important computer vision tasks that have a wide range of applications. Image tagging involves automatically assigning descriptive tags to an image that represents its content, while road object detection involves identifying and localizing objects on the road, such as vehicles, pedestrians, and traffic signs.


One of the main challenges in image tagging is the large variation in image content and the subjective nature of the tags. Different people may describe the same image with different tags, making it difficult to define a ground truth for training and evaluation. In addition, image tagging requires a deep understanding of the image content, including object recognition, scene understanding, and context analysis, which makes it a challenging problem for machine learning algorithms.

Road object detection, on the other hand, is a critical task for a wide range of autonomous driving applications, such as collision avoidance, path planning, and traffic monitoring. However, it is a challenging problem due to the large variation in object appearance and the complexity of the road environment. Object detection algorithms need to be able to accurately identify and localize objects in a wide range of lighting and weather conditions, deal with occlusions and clutter, and distinguish between different object classes, such as cars, pedestrians, and bicycles.

In both cases, developing accurate and reliable algorithms is essential for many real-world applications, but the complexity of the tasks and the variability of the input data pose significant challenges. Addressing these challenges requires developing novel algorithms that can handle the large variability of image content and the complexity of the road environment, as well as designing effective evaluation methods to assess the performance of these algorithms.

## Objectives

The objective of image tagging is to develop accurate and reliable algorithms that can automatically assign descriptive tags to images. This objective can be achieved by developing machine learning or deep learning algorithms that can learn to recognize objects, scenes, and contexts in images and associate them with appropriate tags. The algorithms should be able to handle the large variability of image content and produce consistent and meaningful tags that accurately represent the image content. Effective



evaluation methods should also be designed to assess the performance of the algorithms and compare them to human performance. In addition, the algorithms should be designed to handle large-scale datasets and be scalable to enable efficient processing of large numbers of images.

The objective of road object detection and tracking is to develop accurate and reliable algorithms that can identify, localize, and track objects on the road, such as vehicles, pedestrians, and traffic signs. This objective can be achieved by developing object detection and tracking algorithms that can handle the large variation in object appearance and the complexity of the road environment. The algorithms should be able to accurately detect and classify objects in a wide range of lighting and weather conditions, deal with occlusions and clutter, and track objects over time to maintain their identity. Effective evaluation methods should also be designed to assess the performance of the algorithms and compare them to human performance. Additionally, the algorithms should be designed to operate in real time and be computationally efficient to enable deployment in autonomous driving systems. The algorithms should also be able to handle multi-object scenarios where multiple objects are present in the scene and interact with each other.

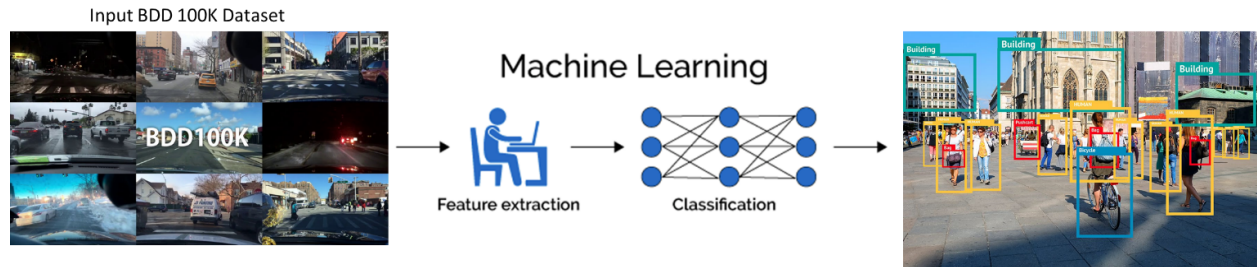
## Dataset Link and Description

### [BDD 100K Dataset](#)

1. The Berkeley Deep Drive (BDD) dataset is one of the largest and most diverse video datasets for autonomous vehicles.
2. The dataset contains 100,000 video clips collected from more than 50,000 rides covering New York, San Francisco Bay Area, and other regions.
3. The dataset contains diverse scene types such as city streets, residential areas, and highways.
4. Furthermore, the videos were recorded in diverse weather conditions at different times of the day

## Methodologies

### Traditional Machine Learning Methods for Image Tagging and Road Object Detection



Steps involved in the above process flow:

1. Collect a large dataset(BDD100K) of road images along with the annotations for the bounding boxes around the objects of interest.
2. Extract meaningful features from the images, such as object appearance, texture, and shape, using various Machine Learning methods.
3. Apply Machine Learning methods to do the classification and tagging. SVMs are a popular supervised learning algorithm that can be used for object detection. SVMs can separate data points in a high-dimensional space, which can be useful for detecting objects in images.

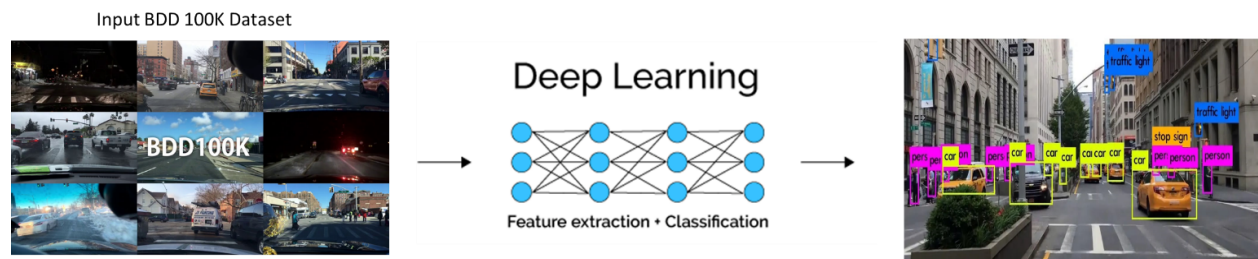
Random Forests are an ensemble learning method that can be used for object detection. They consist of multiple decision trees, each of which votes on the classification of an object.

Some well-known Machine Learning methods are-

- Scale-Invariant Feature Transform (SIFT)
- Histogram of Oriented Gradients (HOG) features
- Viola-Jones object detection framework

Deep learning methods are better than traditional machine learning methods for object detection and tracking because they can learn complex features, are trained end-to-end, adapt to new data, perform better, and handle large datasets.

## Deep Learning Methods for Image Tagging and Road Object Detection



Steps involved in the above process flow:

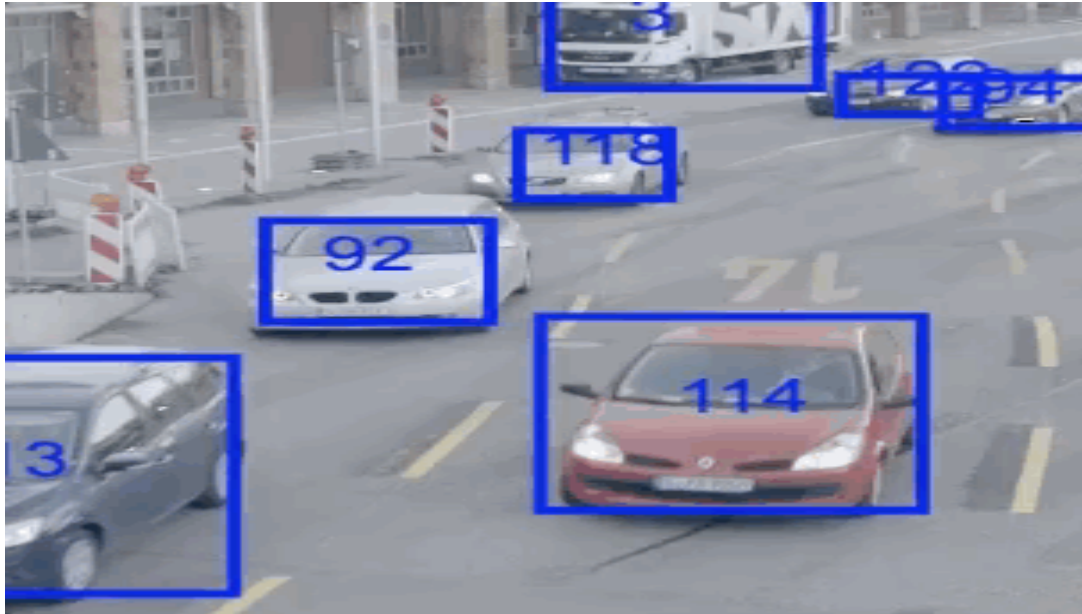
1. Collect a large dataset(BDD100K) of road images along with the annotations for the bounding boxes around the objects of interest.
2. Normalize the images depending on the Deep Learning method to be used and extract meaningful features from the images, such as object appearance, texture, and shape, using deep learning techniques like CNNs.
3. Train an object detection model on the features extracted from the images and detect the objects of interest, such as vehicles, pedestrians, and traffic signs.
4. Test and Evaluate the performance of the object detection and tracking model using standard evaluation metrics such as Intersection over Union (IoU) and Average Precision (AP).

Some well-known Deep Learning methods are-

- Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN)
- You Only Look Once (YOLO)
- Deformable convolutional networks
- Refinement Neural Network for Object Detection (RefineDet)
- Retina-Net

Deep learning models have shown better performance than traditional machine learning methods in many computer vision tasks, including object detection and tracking. For example, deep learning-based object detection systems such as Faster R-CNN and YOLO have achieved state-of-the-art performance on standard benchmarks such as COCO and PASCAL VOC. Hence we have used YOLOv5 as the Object Detection model for BDD100K dataset.

## Methodologies for Multiple Object Tracking System



Steps involved in the above process flow:

1. Apply a tracking algorithm to track the objects over time by estimating their position and velocity from consecutive frames.
2. Incorporate multi-object tracking methodologies to track multiple objects simultaneously and address object interactions such as occlusions, merging, and splitting.

Some well-known multi-object tracking methodologies are-

- SORT
  - DeepSORT
  - Object Tracking with OpenCV
  - Object Tracking MATLAB
  - MDNet
3. Evaluate and compare the performances of the object detection and tracking models using standard evaluation metrics such as MOTA, MOTP, MODA, etc.

DeepSORT is a good object-tracking algorithm choice, and it is one of the most widely used object-tracking frameworks. We have integrated the YOLOv5 model with DeepSORT to achieve multiple object-tracking targets.



## Design Considerations

### Tools

TensorFlow, PyTorch, Keras

### Deployments

Streamlit Cloud

### Data Preprocessing

Raw image data should be converted using python scripts depending on the Deep Learning method to be used.

### Model Selection

Choosing an appropriate deep learning model for image tagging and object detection is important, as different models may be more suitable for different types of image data. Careful evaluation of different models and selection of the most appropriate one is critical to achieving good performance.

YOLO (You Only Look Once) is a popular object detection model that has gained popularity in recent years due to its speed and accuracy. While there are many other object detection models available, YOLO has several advantages that make it stand out from the rest.

Here are some of the key reasons why YOLO is considered better than other object detection models:

- **Speed:** YOLO is known for its high speed, as it can detect objects in real time, making it ideal for applications that require quick analysis of large volumes of video or image data.
- **Accuracy:** YOLO achieves high accuracy in object detection by using a single neural network that predicts bounding boxes and class probabilities directly from full images in a single pass.
- **Localization:** YOLO is designed to localize objects accurately, meaning that it can detect objects that overlap or occlude each other, which is challenging for other object detection models.
- **Simplicity:** YOLO is a simple and easy-to-use model that requires minimal preprocessing of input images or videos. It also has a small number of hyperparameters, which makes it easier to tune and optimize.



- **End-to-end:** YOLO is an end-to-end model that can handle the entire object detection pipeline, from preprocessing input images to predicting bounding boxes and class probabilities, without requiring any intermediate steps.

However, it is worth noting that other object detection models may be more appropriate for certain use cases or have different trade-offs between speed and accuracy.


## Tracking System Selection

The system should be designed to handle multi-object scenarios, where multiple objects are present in the scene and interact with each other. This requires the use of advanced data association techniques and algorithms that can track multiple objects simultaneously. The system should be designed to perform in real-time to enable it to be used in autonomous driving systems. This requires the algorithms to be computationally efficient and optimized for deployment on specialized hardware.

DeepSORT (Deep Learning for Multi-Object Tracking) is a state-of-the-art object tracking model that combines deep learning and classic tracking algorithms. Here are some of the key reasons why DeepSORT is considered better than other tracking models:

- **Accurate:** DeepSORT uses deep learning to extract rich features of each object and match them across frames. This allows the model to track objects accurately even under challenging conditions, such as occlusion, scale changes, and appearance variations.
- **Robust:** DeepSORT is robust to noisy and missing detections. It can handle situations where some detections are missing or inaccurate and still maintain the tracking of objects.
- **Scalable:** DeepSORT is scalable to track multiple objects in real time. This means that it can handle a large number of objects simultaneously, which is important for tracking in crowded scenes.
- **Flexible:** DeepSORT is flexible to be used with different types of detectors and tracking algorithms. It can be integrated with various object detectors, such as YOLO, Faster R-CNN, and SSD, to extract features for tracking.
- **Open-source:** DeepSORT is an open-source model, which means that it can be easily customized and extended for different applications.

In summary, DeepSORT's accuracy, robustness, scalability, flexibility, and open-source nature make it a powerful and popular object-tracking model. However, it is worth noting that the performance of DeepSORT depends on the quality of object detection and the



hyperparameters used, and other tracking models may be more appropriate for certain use cases or have different trade-offs between speed and accuracy.

## Evaluation Metrics

It is important to carefully select and evaluate the performance of the object detection and tracking system using appropriate metrics, such as Intersection over Union (IoU) and Average Precision (AP), to ensure that it meets the required performance criteria.

## Ethical Considerations

The object detection and tracking system should be designed with ethical considerations in mind, such as ensuring the privacy of individuals and avoiding biased or discriminatory outcomes. It is important to carefully consider the impact of the system on all stakeholders and design it to be fair, transparent, and trustworthy.

## Outcomes Achieved in Stages

### Data Collection

The first stage in image tagging and road object detection is to collect a large dataset of images that will be used to train and test the tagging algorithm. Here we are using the BDD100K dataset for training and evaluation.

### Select a Model for Object Detection

YOLO's speed, accuracy, simplicity, localization, and end-to-end capabilities make it a powerful and popular object detection model. Here we are using the YOLOv5 model for object detection.

### Preprocessing

A small subset of data is taken to convert the raw data in the format that is accepted by the selected model. The outcome of this step is a python script that takes BDD100K data and converts it to YOLO supported format.

### Model Training on BDD10K Data

Once the images are converted to YOLO-supported format, the next stage is to use computer vision algorithms to detect objects in the images, such as cars, pedestrians, and

other obstacles. We took a pre-trained (on COCO dataset) YOLOv5 deep learning model and trained it on a subset of the BDD10K dataset. We optimized its parameters to achieve the best performance.

## Testing and validation on BDD10K Data

The trained model is then tested on the remaining unseen subset of the BDD10K dataset to evaluate its performance and ensure that it is able to tag new images accurately. Training and Testing on the BDD10K dataset were repeated until we achieved good accuracy.

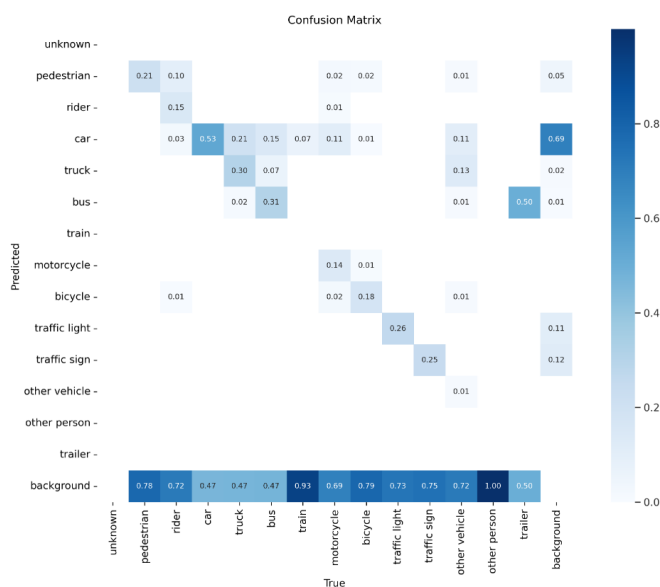
## Model Training on BDD100K Data

Once we achieved good performance on the BDD10K dataset, we trained the YOLOv5 model on a subset of the BDD100K dataset. Model parameters were updated according to the requirement.

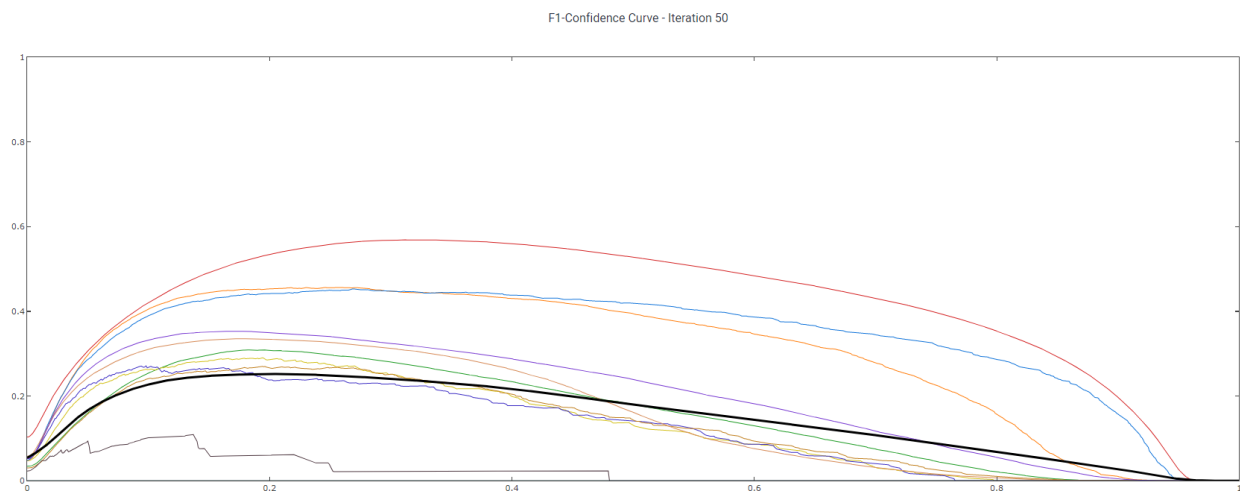
## Testing and validation on BDD100K Data

The trained model is then tested on the remaining unseen subset of the BDD100K dataset to evaluate its performance and ensure that it is able to tag new images accurately. We also tested the model with some unseen images (not part of BDD100K dataset) and the model could detect objects in the images, such as cars, motorcycles, buses, pedestrians, etc.

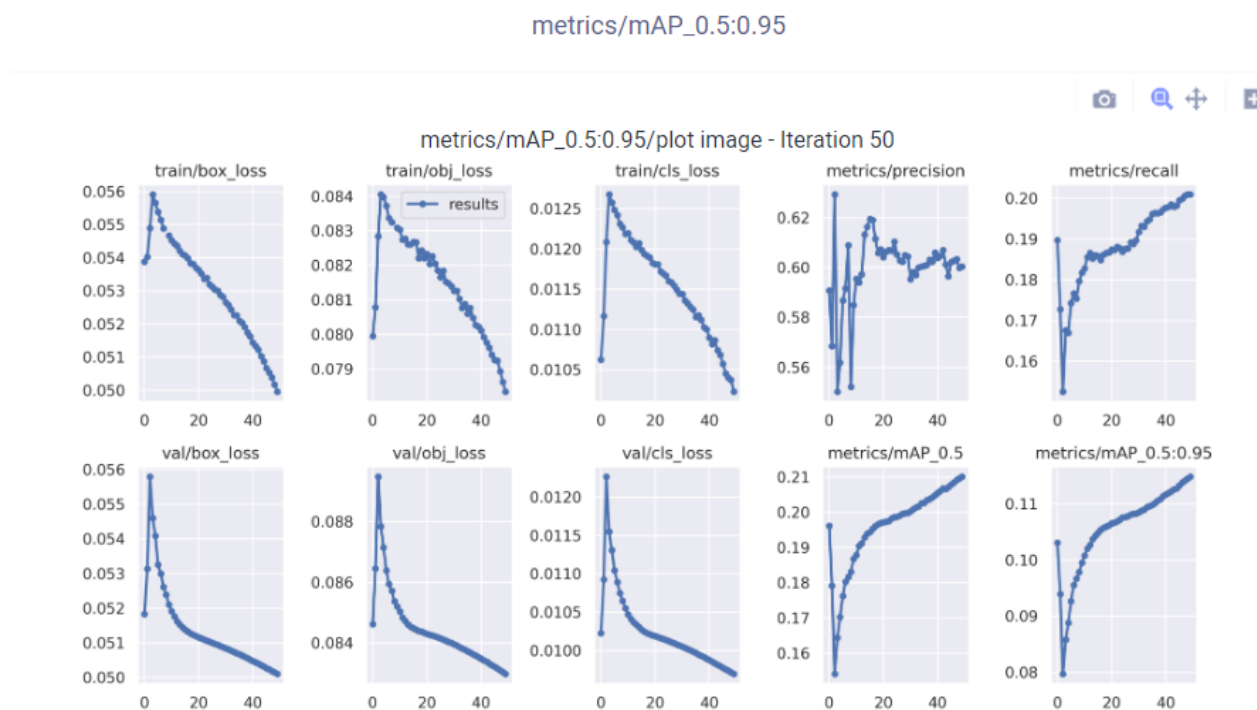
## Model Statistics



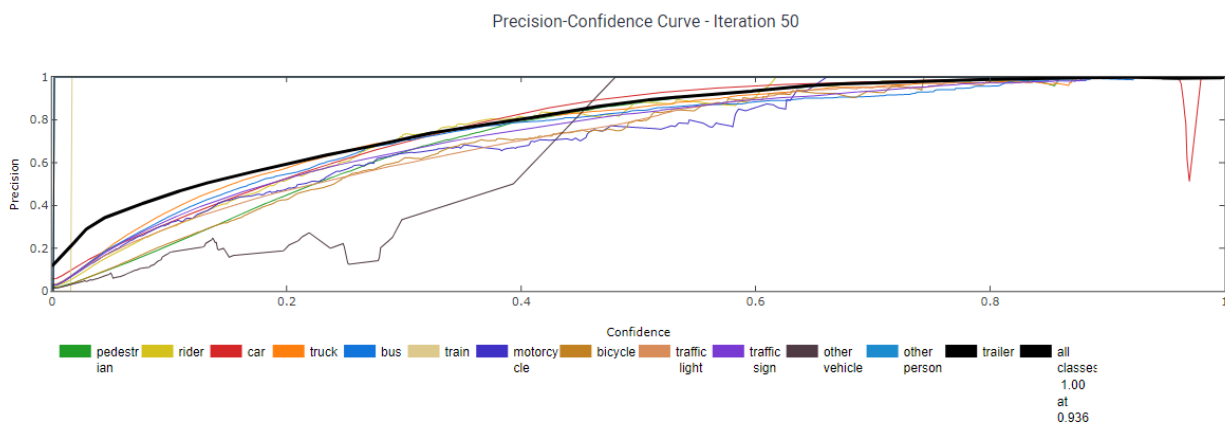
The above graph shows the confusion matrix of object classes that are validated by the trained model.



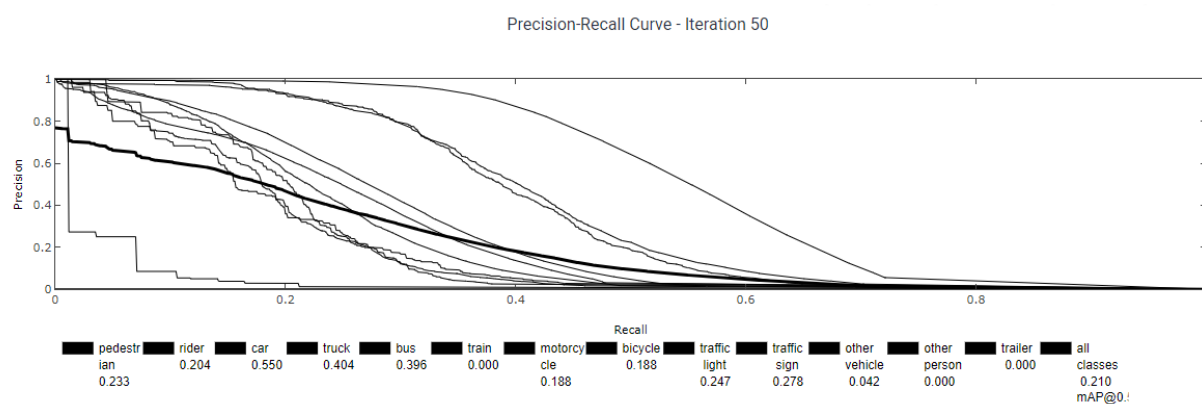
The above graph shows F1 Confidence curve of the model



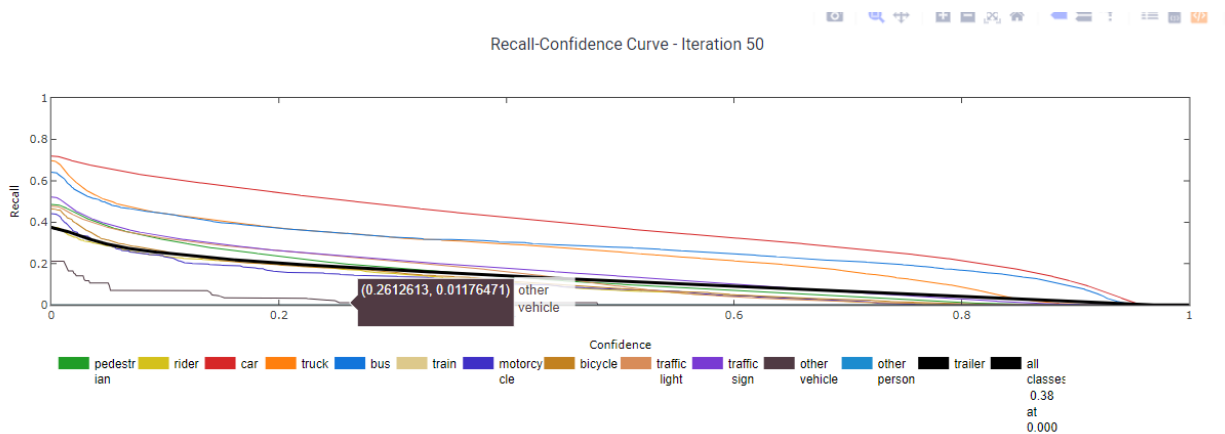
The above graphs show the training and validation losses metrics/precision and metrics/recall curves.



The above report shows Precision-Confidence curves for the trained model



The above report shows Precision-Recall curves for the trained model.



The above report shows Recall-Confidence curves for the trained model.

## Select an Algorithm for Object Tracking

Once objects have been detected, the next stage is to track them over time as they move through the scene. This involves using algorithms that can handle occlusions, lighting changes, and other challenges. DeepSORT's accuracy, robustness, scalability, flexibility, and open-source nature make it a powerful and popular object-tracking model. We selected DeepSORT to integrate with YOLOv5(trained on BDD100K dataset) for object tracking.

## Object Tracking

After integrating the DeepSORT algorithm with our YOLOv5 object detection model, we tested the algorithm on multiple videos from BDD100K video dataset. DeepSORT was able to track objects accurately even under challenging conditions, such as occlusion, scale changes, and appearance variations. We also tested the Tracker on multiple random traffic videos and got satisfactory tracking results.

## Evaluation of the Tracker

We integrated SORT and DeepSORT with our YOLOv5 model trained on BDD100K dataset and compared the performance of the tracker using different metrics like HOTA, CLEARMOT, and Identity metrics. We observed that for the same input video tracking data, DeepSORT is having better MOTA, and MODA values than SORT.

### Metrics for SORT:

```

Notebook
Evaluating 1 tracker(s) on 1 sequence(s) for 8 class(es) on BDD100K dataset using the following
Evaluating qdtrack
1 eval_sequence(0000f77c-6257be58, qdtrack) 0.5079 sec
All sequences for qdtrack finished in 0.51 seconds
HOTA: qdtrack-cls_comb_cls_av HOTA DetA AssA DetRe DetPr AssRe A
COMBINED 1.1888 1.0197 1.4673 1.2369 3.2848 4.1644 1
CLEAR: qdtrack-cls_comb_cls_av MOTA MOTP MODA CLR_Re CLR_Pr MTR P
COMBINED -65.209 8.4561 -65.187 1.0102 2.6828 0 6
Identity: qdtrack-cls_comb_cls_av IDF1 IDR IDP IDTP IDFN IDFP
COMBINED 0.93548 0.64387 1.7099 58 1068 371
Count: qdtrack-cls_comb_cls_av Dets GT_Dets IDs GT_IDs
COMBINED 429 1126 12 47

```

## Metrics for DeepSORT:

Notebook

Evaluating qdtrack

1 eval\_sequence(0000f77c-6257be58, qdtrack) 0.4664 sec

All sequences for qdtrack finished in 0.47 seconds

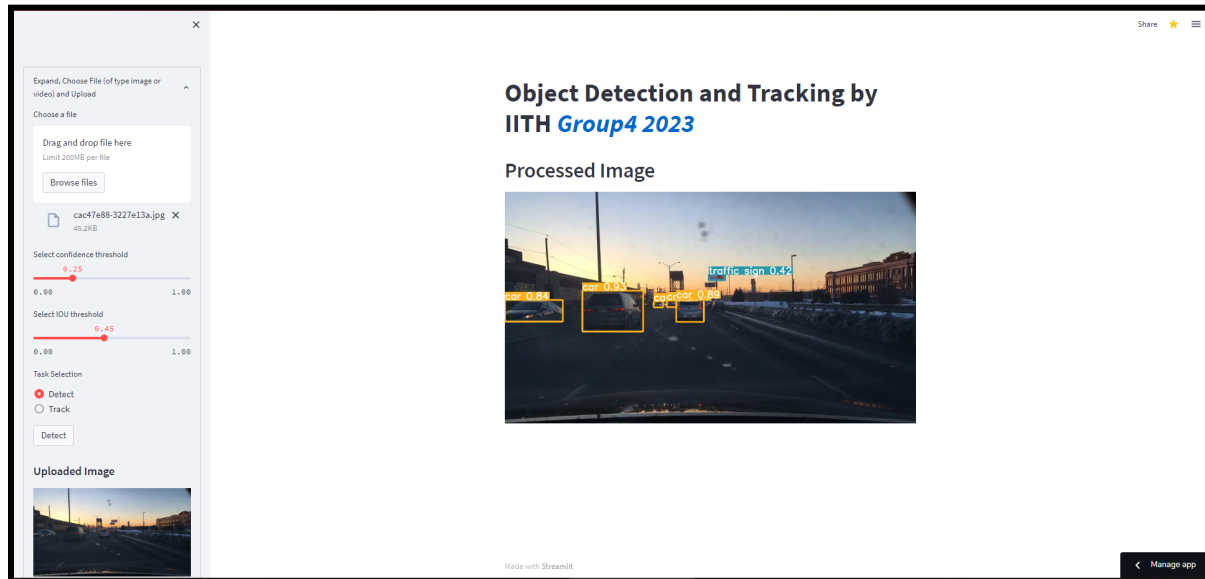
HOTA: qdtrack-cls_comb_cls_av COMBINED	HOTA 1.1288	DetA 0.97768	AssA 1.3959	DetRe 1.1909	DetPr 3.093	AssRe 4.0158
CLEAR: qdtrack-cls_comb_cls_av COMBINED	MOTA -3.1417	MOTP 8.3876	MODA -3.1194	CLR_Re 0.8467	CLR_Pr 2.1991	MTR 0
Identity: qdtrack-cls_comb_cls_av COMBINED	IDF1 0.74003	IDR 0.51248	IDP 1.331	IDTP 46	IDFN 1076	IDFP 386
Count: qdtrack-cls_comb_cls_av COMBINED	Dets 432	GT_Dets 1122	IDs 5	GT_IDs 47		

## Deployment

In the final stage, we deployed the object detection and tracking system in a production environment using the Streamlit application, which can be used to automatically detect objects in new images, and videos and accurately track objects in videos.



## Streamlit UI



## Set of Detected Images in the deployed system

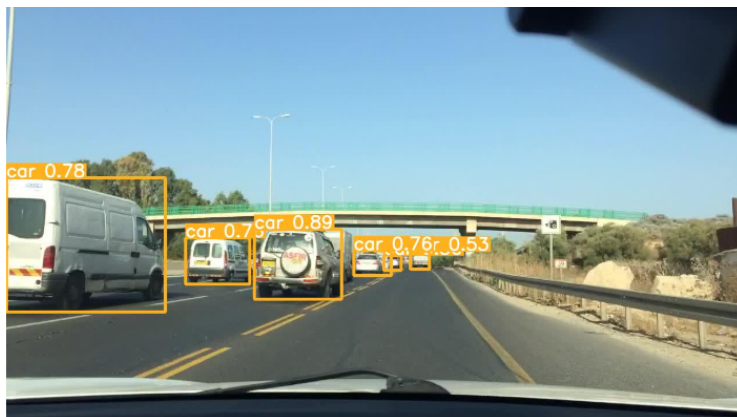


Figure 1

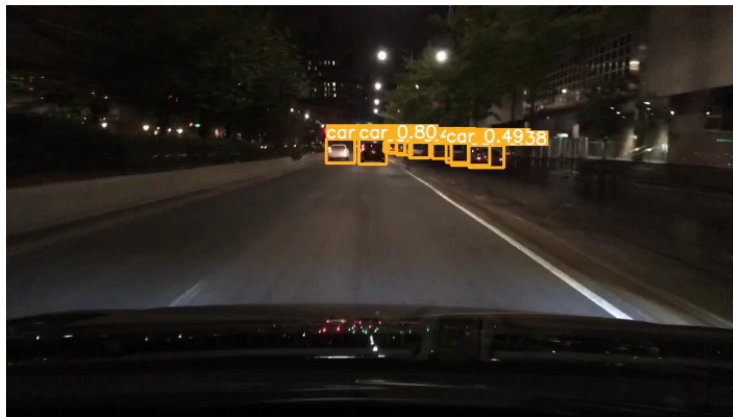


Figure 2

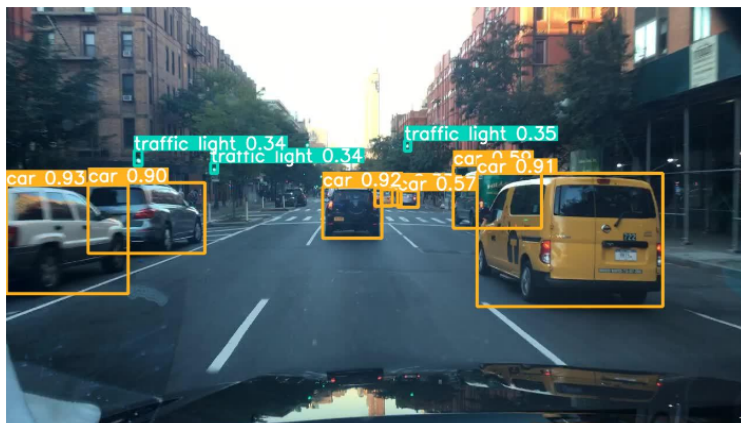



Figure 3

The above three samples show that the detection model works well for most images in day or night. There are images taken during extreme weather that do not respond well to detection. It is felt that more images under different weather conditions, when used for training the model, will help in detection.

## Challenges

An object viewed from different angles may look completely different. Images can be complex and ambiguous, making it challenging to generate accurate and meaningful tags that reflect the content of the image. Depending on the viewpoint variation, the same objects can be classified as different entities by the object detection systems.



Objects on the road can be partially or completely occluded, making it difficult to accurately detect and track them. Sometimes only a small portion of an object (as little as a few pixels) may be visible. The effects of illumination are drastic on the pixel level. Objects exhibit different colors under different illumination conditions. For example, an outdoor surveillance camera is exposed to different lighting conditions throughout the day, including bright daylight, evening, and night light. An image of a pedestrian looks different in these varying illuminations. This affects the capability of the detector to detect objects robustly. Sometimes the objects of interest may blend into the background, making them hard to identify.

Class imbalance is also one of the major challenges in the case of classification problems. Object detection algorithms need to not only accurately classify and localize important objects, but they also need to be incredibly fast at prediction time to meet the real-time demands of video processing. Hence that speed for real-time detection is required. Object detection and tracking must be performed in real-time to be useful for autonomous driving systems, requiring computationally efficient algorithms that can be deployed on specialized hardware.

Tracking multiple objects simultaneously and accurately can be challenging, particularly when objects interact with each other, occlude with each other, or merge and split.

There are ethical considerations to consider when using object detection and tracking technology in areas such as privacy and data security. There may be concerns about the use of the technology for surveillance or potential biases and discrimination in the algorithms.

## Real World Applications

An image tagging and road object detection system can be very helpful in the following real-world scenarios:

- **Self-driving cars** can use this system to detect and classify objects(e.g., road signs or traffic lights). These systems use cameras and other sensors to detect and track objects on the road, enabling the vehicle to create 3D maps and make informed decisions about steering, acceleration, and braking.
- **Parking occupancy detection** using this system to achieve extremely high accuracy in different illuminance and weather conditions.
- **Traffic flow analysis** using this system to accurately monitor and analyze traffic density in different areas (e.g., at intersections), helping to design better traffic management systems and improve road safety.

- **Surveillance and Security systems** use Object detection and tracking to monitor public areas and identify potential security threats.
- **Infrastructure maintenance:** Road object detection and tracking can be used to monitor the condition of roads and bridges, identifying areas where repairs or maintenance are needed. This can help to prevent accidents and ensure that infrastructure is kept in good condition.
- **City planning** with this system by providing real-time location information from social media pictures that are captured by different users.

## References

1. Object Detection with Deep Learning: A Review Zhong-Qiu Zhao, Member, IEEE, Peng Zheng, Shou-tao Xu, and Xindong Wu, Fellow, IEEE
2. YOLOv4: Optimal Speed and Accuracy of Object Detection Alexey Bochkovskiy\* alexeyab84@gmail.com Chien-Yao Wang\* Institute of Information Science Academia Sinica, Taiwan kinyiu@iis.sinica.edu.tw Hong-Yuan Mark Liao Institute of Information Science Academia Sinica, Taiwan liao@iis.sinica.edu.tw
3. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun
4. HOTA: A Higher Order Metric for Evaluating Multi-object Tracking: Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé & Bastian Leibe
5. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics: Keni Bernardin & Rainer Stiefelwagen
6. Simple Online and real-time Tracking with a Deep Association Metric: Nicolai Wojke, Alex Bewley, Dietrich Paulus

## Thank You

This project has helped us sharpen our analytical skills and develop a deeper understanding of different Computer Vision methodologies.

Thank you, Professor Anoop, and mentor Sangeeth for your invaluable support and guidance to complete the project successfully.