**Image tagging and road object detection.**

**Group 8**

|  |  |  |
| --- | --- | --- |
| **Professor** | **Mentor** | **Team members** |
| Prof. Anoop | Sandeep Nagar | Bhaskar Saranga  Rachana Singh Pawar |

Contents

[1. Problem Statement 3](#_Toc128221943)

[2. Objective 3](#_Toc128221944)

[3. Design considerations 3](#_Toc128221945)

[4. MOD Methodologies 5](#_Toc128221946)

[4.1 YOLO (You Only Look Once) 5](#_Toc128221947)

[4.1.1 YOLOV7 6](#_Toc128221948)

[4.1.2 YOLO V8 7](#_Toc128221949)

[4.2 Stage wise development 8](#_Toc128221950)

[5. MOD Observations 9](#_Toc128221951)

[5.1 Recall and Precision 9](#_Toc128221952)

[5.2 mean Average Precision(mAP) 9](#_Toc128221953)

[5.3 Training Losses 10](#_Toc128221954)

[5.3.1 Box Loss and Class Loss 10](#_Toc128221955)

[5.3.2 Objectness Loss 11](#_Toc128221956)

[5.4 Validation Losses 11](#_Toc128221957)

[5.4.1 Box Loss and Class Loss 12](#_Toc128221958)

[5.4.2 Objectness Loss 12](#_Toc128221959)

[6. MOT Methodologies 13](#_Toc128221960)

[6.1 Tracking algorithms 13](#_Toc128221961)

[6.1.1 ByteTrack 13](#_Toc128221962)

[6.1.2 Strong Sort 13](#_Toc128221963)

[6.1.3 OC-Sort 14](#_Toc128221964)

[6.2 Trackers comparison 14](#_Toc128221965)

[7 Literature references 15](#_Toc128221966)

# Problem Statement

In modern times, with the increasing use of images and videos on social media platforms and the internet, there is a need to automate the process of tagging images and detecting objects on roads for various applications. Computer vision algorithms can be used to tag images by detecting and recognizing objects in them, and to detect road objects to ensure safety on roads.

The challenge is to develop an efficient and accurate computer vision system that can automatically tag images and detect objects on roads in real-time. This system should be able to handle various image and video formats, recognize different objects and road conditions, and classify them accurately. It should also be able to operate under different lighting and weather conditions and handle various camera perspectives and angles.

The success of this project will enable better organization and searchability of image databases, and improve the safety of drivers and pedestrians on the roads. This technology can be applied to a wide range of fields, such as surveillance, transportation, and media industries, among others.

# Objective

The objective of this project is to build a road object detection and tracking algorithm model that can automatically detect and classify objects in images and moving video accurately. This solution should take real time factors into considerations such as

* Able to handle various image and video formats.
* Recognize objects in different orientations, scales, and lighting conditions.
* Able to accurately identify the objects on the road including vehicles, pedestrians, traffic signs, and signals in real time
* Able to detect and track objects accurately in real time via object indexing/ tagging.
* Consider practical situations such as occlusion, viewpoint variations(angles), multiple aspect ratios and spatial sizes, object deformation (far off objects).
* Speed and performance of the model.

Evaluate the performance of the computer vision system on various datasets to ensure that it can accurately recognize and tag objects in images and detect objects on the road. The evaluation should be conducted under different weather and lighting conditions to ensure the system's robustness.

Optimize the computer vision system for real-world applications by minimizing the computation time and improving the accuracy of object detection and tagging. The system should also be able to handle different camera perspectives and angles to make it suitable for different applications.

# Design considerations

1. Convolutional Neural Networks (CNNs) - CNNs are a type of deep learning neural network that have been successfully used for object detection and classification in images. CNNs can be trained on a large dataset of labeled images to learn the features of objects and recognize them in new images.

Limitations: CNNs require large amounts of labeled data to train effectively. The architecture of the network and the hyperparameters need to be carefully selected for optimal performance. They are computationally intensive and require high-end hardware to run.

1. Object Detection Frameworks - There are several object detection frameworks available such as You Only Look Once (YOLO), Single Shot Detector (SSD), and Faster R-CNN, which are widely used for object detection in real-time. These frameworks use various techniques such as feature extraction, region proposal, and classification to detect objects accurately.

Limitations: Object detection frameworks can be complex to implement and require significant computational resources. They may struggle with detecting small objects or objects with low contrast. The accuracy of these frameworks is highly dependent on the quality of the training dataset.

1. Image Segmentation - Image segmentation is a technique that involves dividing an image into different regions based on their characteristics. Segmentation can be used to isolate objects in an image and classify them accurately. Segmentation techniques such as U-Net, Mask R-CNN, and Fully Convolutional Networks (FCN) can be used for accurate image segmentation.

Limitations: Image segmentation can be computationally intensive and may require large amounts of memory to process. The accuracy of the segmentation depends on the quality of the training dataset and the choice of algorithm used.

1. Transfer Learning - Transfer learning is a technique that involves using a pre-trained model as a starting point for a new task. Transfer learning can be used to fine-tune an existing CNN model that has been trained on a large dataset to recognize and classify objects in new images.

Limitations: Transfer learning may not always be effective if the pre-trained model is not suited for the target task. Fine-tuning the model requires careful selection of the layers to be trained and the learning rate. The performance of the model is highly dependent on the quality of the pre-trained model.

1. Data Augmentation - Data augmentation involves creating new data from existing data by applying transformations such as rotation, scaling, and flipping. Data augmentation can be used to increase the size of the training dataset and improve the performance of the model.

Limitations: Data augmentation may not always improve the performance of the model if the transformations applied are not relevant to the target task. Overuse of data augmentation may also lead to overfitting of the model.

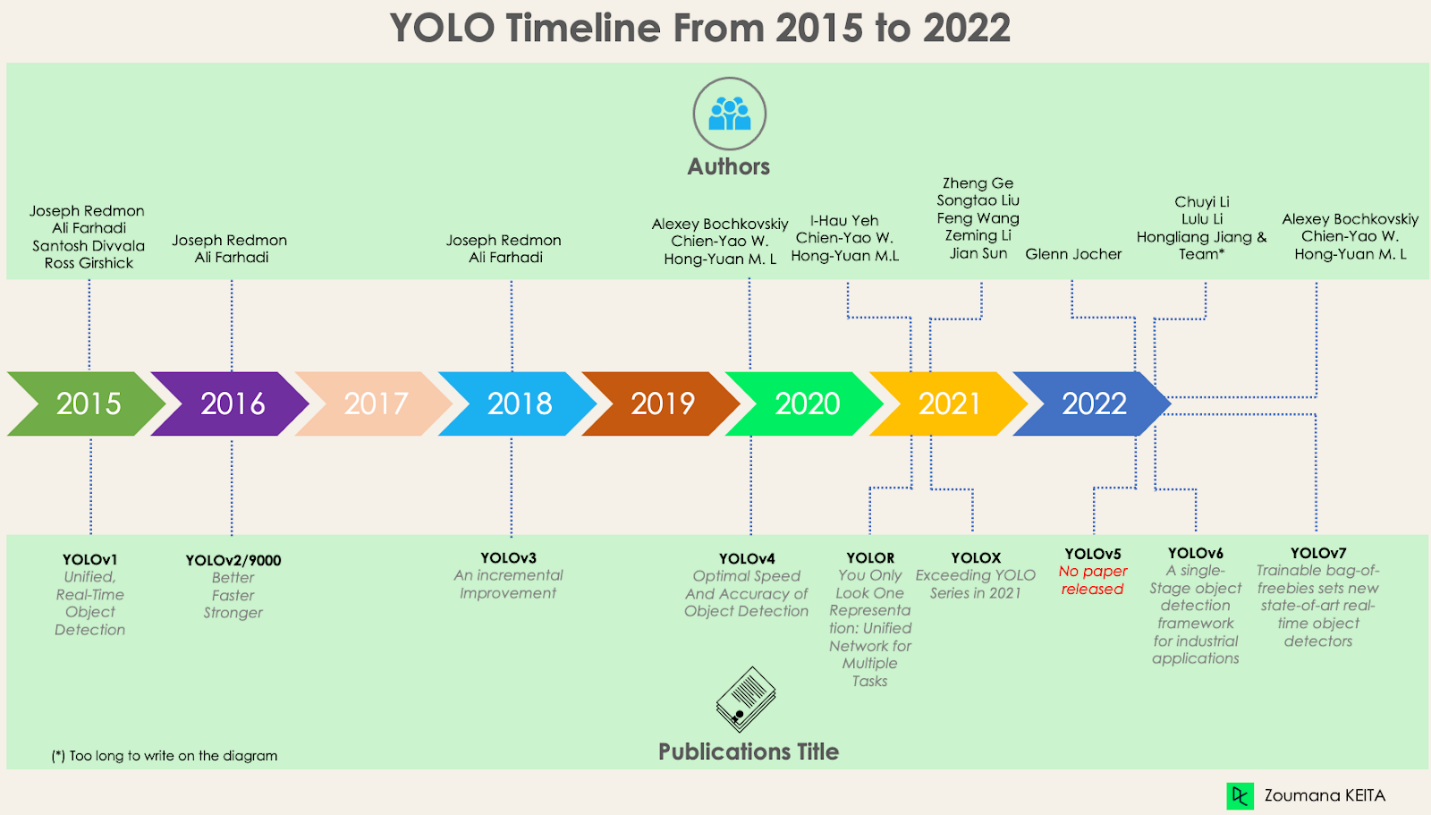
These methodologies can be used individually or in combination to solve the problem of Image Tagging and Road Object Detection with Computer Vision. A comparative analysis of the performance of these methodologies can be performed to identify the most effective approach.

# MOD Methodologies

## 4.1 YOLO (You Only Look Once)

YOLO model is used to solve our problem statement. YOLO are single stage object detectors. In a YOLO model, image frames are featurized through a backbone. These features are combined and mixed in the neck, and then they are passed along to the head of the network YOLO predicts the locations and classes of objects around which bounding boxes should be drawn. YOLO conducts a post-processing via non-maximum suppression (NMS) to arrive at its final prediction.

We trained the YOLOV7 model with BDD dataset . Later we also trained the latest released YOLOV8 model by ultralytics on IDD dataset based on feedback during interim evaluation of the project.



## YOLOV7

YOLO v7 uses nine anchor boxes, which allows it to detect a wider range of object shapes and sizes compared to previous versions, thus helping to reduce the number of false positives.One of the main advantages of YOLO v7 is its speed. It can process images at a rate of 155 frames per second, much faster than other state-of-the-art object detection algorithms. Hence suitable for self driving cars problem statement.

Chart, line chart

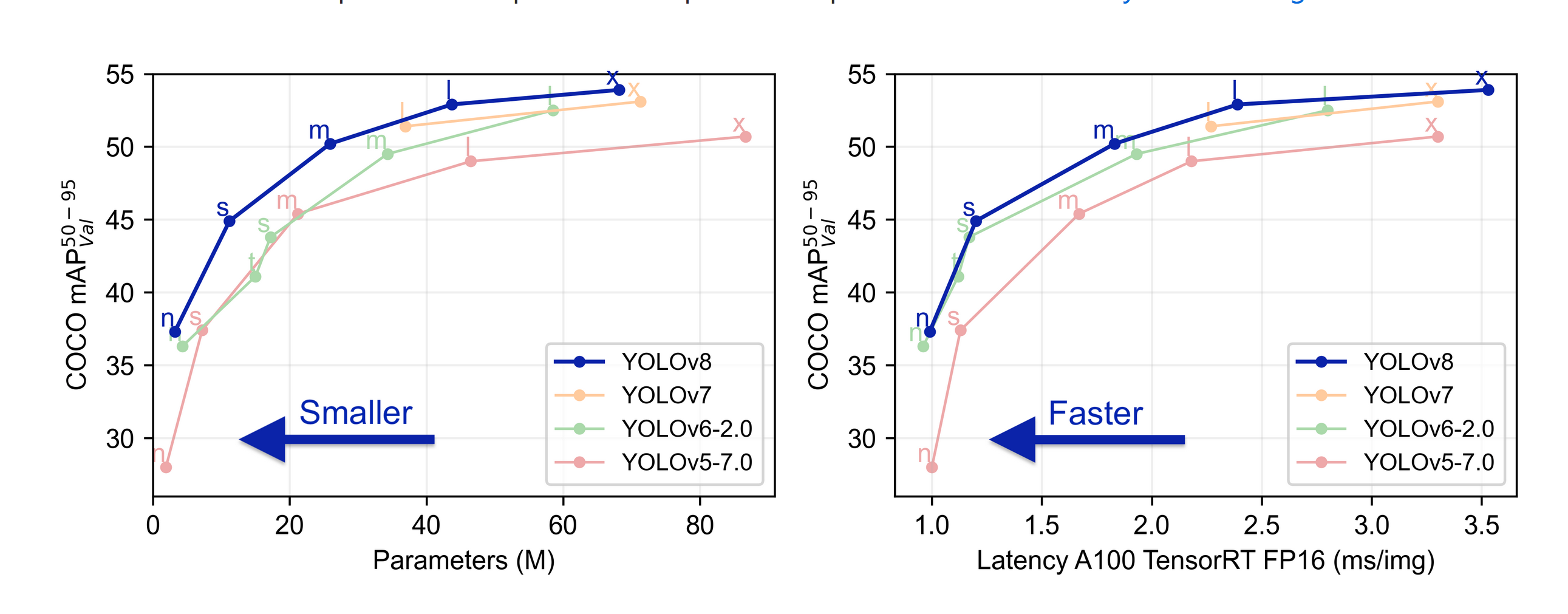
Description automatically generated

Table

Description automatically generated

## YOLO V8

These include a new backbone network, a new anchor-free detection head, and a new loss function. YOLOv8 is also highly efficient and can be run on a variety of hardware platforms, from CPUs to GPUs.YOLOV8 introduces new features and improvements to further boost performance and flexibility.



Table

Description automatically generated

## 4.2 Stage wise development

|  |  |  |
| --- | --- | --- |
| STAGE | STEP PERFORMED | TOOLS |
| 1 | Extract BDD dataset for MOD. Prepare the dataset as per YOLO model requirements.  Extract IDD data from MOD. Prepare the dataset as per YOLO model | Google colab |
| 2 | Download YOLOV7 model pretrained on MS COCO Dataset |  |
| 3 | Retrain the above model on BDD dataset via training and validation.  Retrain the above model on IDD dataset via training and validation | Google colab pro |
| 4 | Test or evaluate the model on BDD MOD 2020 Dataset test data  Test or evaluate the model on IDD MOD Dataset test data | Google colab pro |
| 5 | Upload the pt model to github cloud repo. Build functions to detect image and video file leveraging this model. | github repo |
| 6 | Creating a User Interface using gradio to allow end users to detect images and video. Video is considered as frames of images. | Gradio |
| 7 | Deploy the github repo to Hugging faces to create the web portal | Hugging faces |

# MOD Observations

## 5.1 Recall and Precision

Recall and precision are commonly used metrics to evaluate the performance of object detection algorithms. Recall measures the percentage of objects that were correctly detected out of the total number of objects in the image. In other words, it quantifies the ability of the algorithm to find all the objects in the image. Precision, on the other hand, measures the percentage of objects detected by the algorithm that are relevant or correctly identified. In other words, it quantifies the accuracy of the detections made by the algorithm.

|  |  |  |
| --- | --- | --- |
|  | **Recall** | **Precision** |
| **Yolo V7** | 0.56 | 0.79864763 |
| **Yolo V8** | 0.51 | 0.79261965 |

we can observe that **YOLO V7 has a higher recall** value compared to YOLO V8, indicating that it is better at detecting all the objects in the image. However, **YOLO V7 has a slightly higher precision** value, indicating that it is better at accurately identifying the detected objects. Overall, both versions of YOLO seem to perform reasonably well.

## 5.2 mean Average Precision(mAP)

mAP is a commonly used metric to evaluate the accuracy of object detection algorithms. It measures the average precision across different object classes and different intersection-over-union (IoU) thresholds between the predicted bounding boxes and ground truth bounding boxes. In general, a higher mAP indicates better performance of the object detection algorithm.

|  |  |  |
| --- | --- | --- |
|  | **mAP@0.5** | **mAP@0.5:95** |
| **Yolo V7** | 0.63 | 0.3958016 |
| **Yolo V8** | 0.59 | 0.3913267 |

Based on these metrics, we can observe that **YOLO V7 has a slightly higher mAP** value compared to YOLO V8, indicating that it performs better in terms of accuracy in detecting objects. However, when evaluated at a higher IoU threshold of 0.5:95, YOLO V7 and V8 both show a drop in mAP values, indicating that they are less accurate when considering stricter criteria for object detection. Overall, both versions of YOLO have good mAP scores.

## 5.3 Training Losses

Training loss measures the error between the predicted and ground truth values during the training process of the model. A lower training loss indicates that the model is learning and improving better during training.

#### 5.3.1 Box Loss and Class Loss

|  |  |  |
| --- | --- | --- |
|  | **Training Box Loss** | **Training Class Loss** |
| **Yolo V7** | 0.03 | 0.003 |
| **Yolo V8** | 0.99 | 0.560 |

Based on these metrics, we can observe that YOLO V7 has significantly lower training losses compared to YOLO V8. Specifically, YOLO V7 has a much lower train box loss and train class loss, indicating that the model is better at predicting the bounding box and class of the objects in the image during training. On the other hand, YOLO V8 has a much higher training loss, indicating that it is less accurate in predicting the bounding box and class of the objects in the image during training.

Overall, the training loss metrics suggest that **YOLO V7 is a better** performing model compared to YOLO V8 during the training process. However, it is important to note that these metrics alone are not sufficient to make a conclusive comparison between the two models, and further analysis is required to determine their relative performance on a test dataset.

#### 5.3.2 Objectness Loss

The train obj loss is a measure of the error between the predicted objectness score and the ground truth objectness score during the training process of the model. The objectness score is a measure of how likely a given bounding box contains an object.

|  |  |
| --- | --- |
|  | **Train Objectness Loss** |
| **Yolo V7** | 0.013 |
| **Yolo V8** | 1.013 |

Based on these metrics, we can observe that **YOLO V7 has a much lower train objectness loss** compared to YOLO V8. Specifically, YOLO V7 has a train obj loss of 0.013, which is significantly lower compared to YOLO V8's train obj loss of 1.013. This indicates that the model is better at predicting the objectness score of the bounding box during training.

we can see that **YOLO V7 outperforms** YOLO V8 in terms of **all three training loss metrics**. Specifically, YOLO V7 has lower values for train box loss, train class loss, and train obj loss compared to YOLO V8. This indicates that YOLO V7 is better at predicting bounding box coordinates, class probabilities, and objectness scores during training.

## 5.4 Validation Losses

In object detection, validation losses are used to evaluate the performance of a model on a validation dataset during the training process. Validation losses are computed by comparing the predicted output of the model with the ground truth output on a set of validation images that are not used during the training process.

Validation losses are important to monitor during the training process as they provide insight into how well the model is generalizing to new, unseen data. A decrease in validation loss over time indicates that the model is learning to generalize better and improving its performance on the validation dataset.

#### 5.4.1 Box Loss and Class Loss

The validation box loss measures the error between the predicted bounding box coordinates and the ground truth bounding box coordinates on a validation dataset during the training process. The validation class loss measures the error between the predicted class probabilities and the ground truth class probabilities on a validation dataset during the training process.

|  |  |  |
| --- | --- | --- |
|  | **Validation Box Loss** | **Validation Class Loss** |
| **Yolo V7** | 0.053 | 0.022 |
| **Yolo V8** | 1.043 | 0.648 |

Based on the values reported in the table, we can see that **YOLO V7 outperforms** YOLO V8 in terms of **both validation loss metrics**. Specifically, YOLO V7 has a lower validation box loss and validation class loss compared to YOLO V8. This indicates that YOLO V7 is better at predicting bounding box coordinates and class probabilities on the validation dataset during training.

#### 5.4.2 Objectness Loss

The validation objectness loss measures the model's ability to correctly identify objects in an image. Specifically, it measures the error between the predicted objectness score and the ground truth objectness score on a validation dataset during the training process.

|  |  |
| --- | --- |
|  | **Validation Objectness Loss** |
| **Yolo V7** | 0.013 |
| **Yolo V8** | 1.013 |

From the values reported in the table, we can see that **YOLO V7 has a lower validation objectness loss** compared to YOLO V8, indicating that it is better at identifying objects in the validation dataset during training. On the other hand, YOLO V8 has a higher validation objectness loss, suggesting that it struggles to identify objects accurately on the validation dataset

Based on the values reported in the table, we can see that **YOLO V7 outperforms** YOLO V8 in terms of **all three validation loss metrics**. Specifically, YOLO V7 has lower validation box loss, validation class loss, and validation objectness loss compared to YOLO V8. This indicates that YOLO V7 is better at predicting bounding box coordinates, class probabilities, and identifying objects in the validation dataset during training.

# MOT Methodologies

Object tracking is a [deep learning](https://www.v7labs.com/blog/deep-learning-guide) process where the algorithm tracks the movement of an object. In other words, it is the task of estimating or predicting the positions and other relevant information of moving objects in a video. Object tracking usually involves the process of [object detection](https://www.v7labs.com/blog/object-detection-guide). Here’s a quick overview of the steps:

* 1. Object detection, where the algorithm classifies and detects the object by creating a bounding box around it.
  2. Assigning unique identification for each object (ID).
  3. Tracking the detected object as it moves through frames while storing the relevant information.

## Tracking algorithms

#### ByteTrack

A simple, effective and generic association method, tracking by associating almost every detection box instead of only the high score ones. For the low score detection boxes, it utilize their similarities with tracklets to recover true objects and filter out the background detections StrongSort

#### Strong Sort

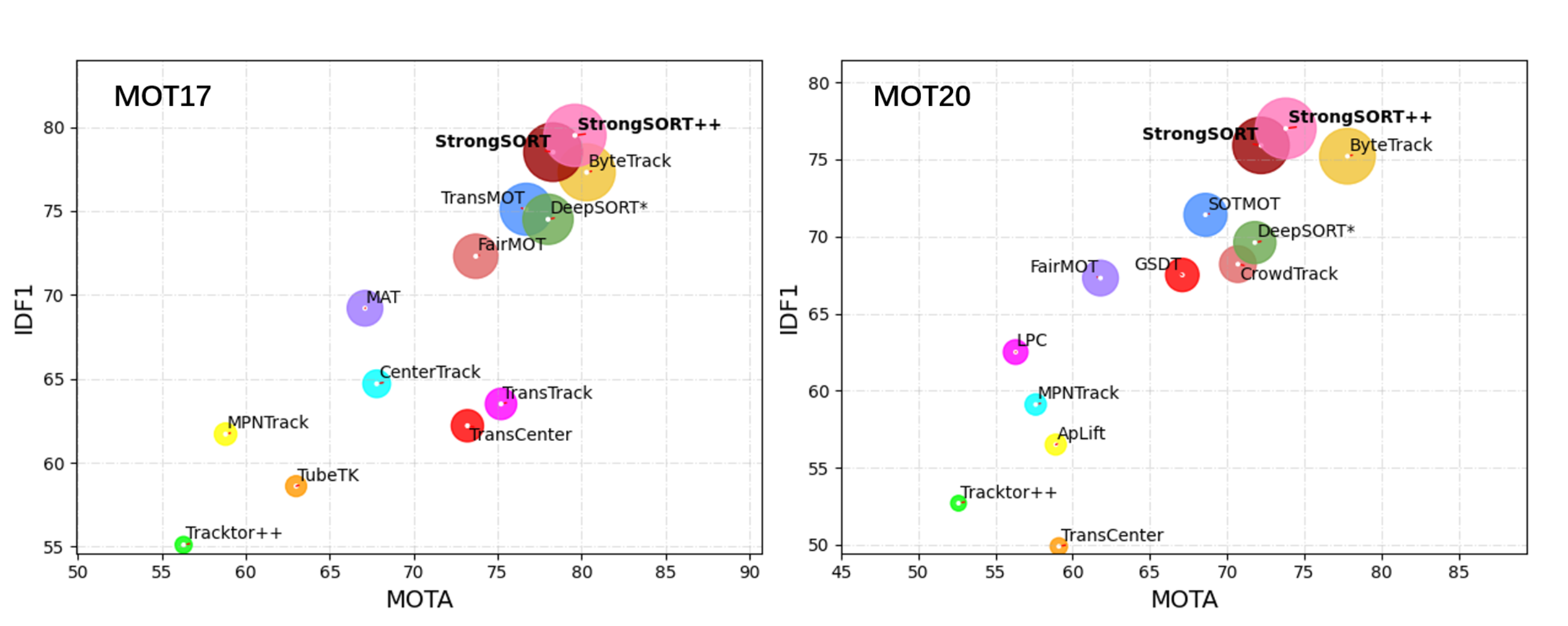
Associates short tracklets into complete trajectories at high computational complexity, proposes an appearance-free link model (AFLink) to perform global association without appearance information, and achieves a good balance between speed and accuracy

#### OC-Sort

Current motion models in MOT typically assume that the object motion is linear in a small time window and needs continuous observations, so these methods are sensitive to occlusions and non-linear motion and require high frame-rate videos. A simple motion model can obtain state-of-the-art tracking performance without other cues like appearance. This emphasize the role of “observation” when re-covering tracks from being lost and reducing the error accumulated by linear motion models during the lost period.

## Trackers comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **MOT 17** | | **MOT 20** | |
| **HOTA** | **MOTA** | **HOTA** | **MOTA** |
| **Stong Sort** | **Feb-23** | **63.5** | 78.9 | **61.5** | **71.2** |
| **Byte Sort** | **Apr-22** | 63.1 | **80.3** | 61.3 | 77.8 |
| **OC-Sort** | **Mar-22** | 63.2 | 78 | 62.1 | 75.5 |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Pros** | **Cons** | **Complexity** | **Accuracy** | **Speed** |
| Bytetrack | * High accuracy due to object-aware tracking * Robustness to occlusions and lighting changes * Supports multi-object tracking | * Requires object detection for initialization * Can be computationally expensive * Limited scalability for large-scale tracking | High | High | Medium |
| StrongSort | * High accuracy with reliable object tracking * Scalable to large-scale tracking * Effective handling of occlusions and re-identification of lost objects | * Requires object detection for initialization * Can be sensitive to camera motion and lighting changes * Limited performance with long-term tracking | Medium | High | Medium |
| OCSort | * Low computational cost * Efficient tracking of small objects * Robustness to occlusions and lighting changes | * Limited performance with long-term tracking * Requires fine-tuning to achieve optimal performance | Low | Medium | High |

# Literature references

* Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR. <https://arxiv.org/abs/1905.11946>

Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781-10790). <https://arxiv.org/abs/1911.09070>

* Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*. <https://arxiv.org/abs/2207.02696>
* Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., ... & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII* (pp. 1-21). Cham: Springer Nature Switzerland. <https://arxiv.org/abs/2110.06864>
* Du, Y., Song, Y., Yang, B., & Zhao, Y. (2022). Strongsort: Make deepsort great again. *arXiv preprint arXiv:2202.13514*. <https://arxiv.org/abs/2202.13514>
* Cao, J., Weng, X., Khirodkar, R., Pang, J., & Kitani, K. (2022). Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*. <https://arxiv.org/abs/2203.14360>