Multiple Object Tracking and Segmentation using BDD100k Dataset

Kathakoli Sengupta,
Boston University,
Boston,
ksg25@bu.edu

Harsh Sharma,
Boston University,
Boston
hsharma@bu.edu

Abstract

Vehicle and person re-identification without facial or gait features has been a surging research topic. The main aim of this project is to build a robust model that can track vehicles and people in video data. The model will perform instance segmentation on the objects(vehicles, persons) in each frame. It will take cues to the particular class to be tracked at a point and use optical flow and Kalman filter to predict its track of motion which localizes the area of search. However, it will not give the exact position, compensated by a deep neural network-based re-id model that will identify the object, plot a particular track, and reidentify it even if it gets out of the frame. Hence, this model will be robust to handling noise related to mask overlapping(after the use of instance segmentation), outlier features such as noise from other cars(due to the deep neural network-based feature extractor), viewpoint errors, and specific abrupt motion change(using motion history based re-id) and will still be able to track the object. The model will be able to deal with all possible shortcomings in multi-object tracking since it will be tested on CVPR 2023 BDD100K Challenge and is expected to beat their baseline. Video Link: https://drive.google.com/file/d/19wnIs3wUwLb7Nv6aer5NhKJXEDVIU9c4/view?usp=share_link

1. Introduction

Multiple object tracking and segmentation is a fundamental yet challenging task for autonomous driving. The BDD100K MOT and MOTS datasets [15] was created to address this challenge, providing diverse driving scenarios with high-quality instance segmentation masks under complicated occlusions and reappearing patterns. The BDD100K dataset presents an excellent testbed for developing tracking and segmentation algorithms in natural scenes.

This paper proposes an approach for solving the multiple object tracking and segmentation problem using the BDD100K dataset. Our approach addresses the challenges of developing accurate tracking and segmentation algorithms that work in diverse driving scenarios, even in complicated occlusions and reappearing patterns.

The major re-identification problems faced in previous research include uncertainty in object tracking in cases of overlapping bounding regions or masks and detecting a new object when it comes back into view after going out of the frame. To address these challenges, we propose a multi-object tracking and segmentation model with the following key components:

- Training an instance segmentation model on the input images to create a segmentation mask for every object.
- Incorporating optical flow to limit the search region in the next frame, improving the efficiency of the tracking algorithm.
- Modifying the re-identification model with a generalized approach to enhance the accuracy of object association across frames.
- Implementing a tracking algorithm that can deal with occlusions and sudden object disappearance from the frame and still perform successful person re-identification on them.

This paper presents an approach to solving the multiple object tracking and segmentation problem using the BDD100K dataset. We aim to develop accurate tracking and segmentation algorithm in diverse driving scenarios by implementing our proposed approach. Section 2 reviews the related works, while Section 3 discusses our proposed methodology. In Section 4, we provide details of the Dataset used in our experiments; in Section 5, we give detailed results of our research; in Section 6, we provide a discussion and conclusions drawn from our algorithm.

2. Related Works

The field of multi-object tracking involves several stages, including object detection and segmentation, object re-

identification, and tracking. In recent years, many researchers have developed approaches to address these challenges.

Segmentation is an essential task in computer vision, including instance, semantic, and panoptic segmentation. Instance segmentation predicts a mask and its corresponding category for each object instance, while semantic segmentation classifies each pixel, including the background, into different semantic categories. Panoptic segmentation unifies the instance and semantic segmentation tasks and predicts a mask for each object instance or background segment.

Several specialized architectures have been developed for the three tasks of segmentation. Mask-RCNN [4] and HTC [2] can only deal with instance segmentation because they predict the mask of each instance based on its box prediction. On the other hand, FCN [10] and U-Net [13] can only perform semantic segmentation since they predict one segmentation map based on pixel-wise classification.

Object re-identification is an essential step in multi-object tracking. Several CNN-based models, such as VehicleNet [17], have been proposed to extract features from images and project them to a low-dimensional space for similarity measurement. However, attention-based methods, such as transreid [6], have been introduced to investigate long-range dependencies.

Other approaches, such as the global-local feature fusion and channel attention mechanism used in [12], have been developed to enhance accuracy by addressing factors such as orientation variations, illumination changes, occlusion, low resolution, rapid vehicle movement, and similar vehicle models. Simple Online and Realtime Tracking with a Deep Association (DeepSort) [14] used a motion model with Mahalanobish distance measurement and an appearance extractor model with cosine distance to efficiently create the track of multiple vehicles. Optical flow estimation with Kalman filter [9] was also used to detect the next state for a particular tracking pixel.

The 1st Place Solution for ECCV2022 SSLAD BDD100K MOT/MOTS/SSMOT/ SSMOTS Challenges used object bounding boxes to detect and a segmentation head to extract binary masks within each detected box. A ReID model extracts features from the bounding boxes, and a tracker matches the object ID in the image sequence.

Our proposed approach builds on these existing works by incorporating a segmentation mask for every object, optical flow to limit the search region in the next frame, and a generalized re-identification model for accurate object association across frames. By addressing the limitations of previous approaches and combining them into a comprehensive approach, we aim to develop accurate tracking and segmentation algorithms that work in diverse driving scenarios.

3. Methodology

This section will discuss each component of our proposed multi-object tracking and segmentation model in detail. Fig. 1 provides the algorithm's flow. We will explain how we train the instance segmentation model on the input images to create a segmentation mask for every object. Next, we will discuss how we incorporate optical flow to limit the region of search in the next frame, improving the efficiency of the tracking algorithm. We will then explain the tracking algorithms we used to accomplish our goals and how we used MaskDINO based segmentation masks as the detection module for tracking algorithms like DeepOCSort, StrongSort and OCSort.

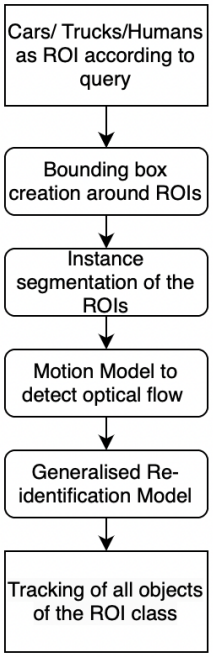

Figure 1. Block diagram.

3.1. Segmentation Module

The detection model will generate instance segmentation masks for each object in each video frame using mask DINO [8], a unified object detection and segmentation framework. Fig.2 shows the model's architecture. The model extends DINO [16] (DETR with Improved Denoising Anchor Boxes) by adding a mask prediction branch that supports all image segmentation tasks, with a ResNet50 [5] backbone.

The segmentation branch in Mask DINO works by performing mask classification to generate multi-class segmentation masks. It adopts a key idea from Mask2Former and constructs a pixel embedding map obtained from the backbone and Transformer encoder features. This map is then

dot-producted with content query embeddings to obtain an output mask. The segmentation head M performs mask classification for all segmentation tasks. The model uses a different head for the bounding box creation of the same network.

The equation for the segmentation branch is as follows:

$$m = q_c \odot M(T(C_b) + F(C_e)) \tag{1}$$

Here, q_c represents the content query embedding, $\odot$ denotes the dot product operation, M is the segmentation head, T is a convolutional layer that maps the channel dimension to the transformer hidden dimension, C_b and C_e are the 1/4 and upsampled 1/8 resolution feature maps from the backbone and Transformer encoder, respectively, and F is a simple interpolation function that performs 2x upsampling of C_e. The content query embeddings are selected based on the confidence scores obtained from the three prediction heads in the encoder output for classification, detection, and segmentation. These prediction heads improve the query selection scheme for segmentation tasks.

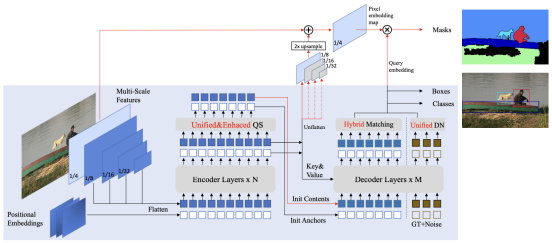

Figure 2. The framework of Mask DINO, which is based on DINO (the blue-shaded part) with extensions (the red part) for segmentation tasks.' QS and DN are short for query selection and denoising training. [8]

In summary, the detection module in our proposed approach will use Mask DINO with a ResNet50-like backbone to generate instance segmentation masks for each object in each video frame. The segmentation branch performs mask classification to generate multi-class segmentation masks using a content query embedding scheme based on three prediction heads in the encoder output.

3.2. Motion Tracking Module

The motion is detected using the optical flow information [9]. Lucas and Kanade's algorithm estimates the optical flow of each object to be tracked to calculate the motion. This motion data is stored for tracking, which requires us to store all the Optical Flow data of all the pixels in each frame. Next, the Kalman filter is integrated with the system that only needs the last nearest one-frame motion data. A standard Kalman filter tracks objects with linear observation with bounding coordinates as input and constant velocity motion. The track's age is counted and those exceeding a maximum age are deleted. New track hypotheses are created for unassociated detections and classified as tentative for the first three frames, after which they are deleted if no successful association is made [14]. The Hungarian algorithm solves the association problem between predicted Kalman states and new measurements. Motion and appearance information are integrated using two metrics. The Mahalanobis distance incorporates motion information, and unlikely associations are excluded by thresholding the distance at a 95% confidence interval. The Mahalanobis distance is given by Eq. 2.

$$d_{i,j}^{(1)} = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{2}$$

We denote the projection of the i-th track distribution into measurement space by (y_i, S_i) and the j-th bounding box detection by d_j.

$$b^{(1)} i, j = 1[d^{(1)}(i, j) \leq t^{(1)}] \tag{3}$$

is used to measure the admissibility based on both tracks and t as the threshold parameter. The approximate estimation of the track of the concerned vehicle will be made more certain with a Transformer based Re-identification model on top of it.

3.3. Appearance Descriptor Module

The appearance descriptor module is a CNN trained on many person re-identification datasets. The feature embeddings in the appearance model work as a successful re-identification technique and help to make effective nearest neighbor queries to the motion module.

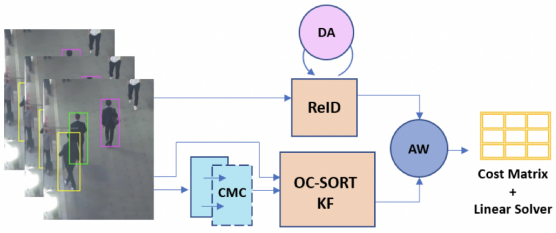

Figure 3. The DeepOCSort framework that shows the best results among the tracking algorithm used in this paper [11]

This research explores three tracking algorithms based on this motion and appearance modules. The StrongSort [3] uses a very deep feature representation network trained on large-scale datasets, allowing it to capture more complex and discriminative features better suited for tracking individuals. It has a multi-stage detector, which enables the algorithm to recover from occlusions more effectively. It

introduces a mechanism for handling re-identification failures by temporarily storing the identity of a lost object until it is re-identified. This reduces the likelihood of misidentifying objects and improves the overall tracking accuracy.

The OCSort [1] can handle the occlusions, if any, in the tracking method. It uses the motion history of the object to predict its position and update its state vector. To prevent the incorrect association of objects during occlusion, OC-SORT employs an adaptive gating strategy, where the gating threshold is adjusted based on the number of unassociated objects in the frame. This ensures that only objects within a certain distance of each other are considered for association.

Deep OC-SORT [11] extends the observation-centric approach of OC-SORT by incorporating a deep neural network for person re-identification and introduces an adaptive learning strategy for updating the deep neural network, which handles the limitation that OCSort may struggle with accurate pedestrian re-identification in challenging scenarios with its gating strategy. The detailed architecture is shown in Fig.3

3.4. Prototypical Cross attention Network

One recent approach to improving segmentation predictions is cross-attention, which uses past spatiotemporal information encoded in memory to attend to relevant memories using separate key and value feature vectors. However, the standard attention operation suffers from poor computational and memory scaling properties. Prototypical cross-attention (PCAN) [7] has been introduced to address these limitations. PCAN employs a clustered memory consisting of prototypes generated by fitting a Gaussian Mixture Model to the memory's keys, allowing a soft cluster assignment to be computed. The corresponding value prototypes are then retrieved using key cluster assignment probabilities. For attending to the clustered memory, PCAN computes the average over the value prototypes, weighted with the cluster assignment probabilities. The final attention operation is similar to the original dot-product cross attention but attends to a reduced set of prototypes. PCAN addresses the limitations of standard cross-attention and provides a more efficient and robust approach for utilizing past spatiotemporal information to improve segmentation predictions.

4. Dataset

We use BDD100k [15] Dataset. The Multiple Object Tracking and Segmentation(MOTS) datasets sampled at 5Hz, which takes a total space of 6GB. There are 8 classes defined in this dataset namely, Pedestrian, Rider, Car, Truck, Bus, Train, Motorcycle, Bicycle. However for an evaluation parity with MaskDINO that supports only seven of this class's segmentation, the rider class was relabelled same as pedestrian in this research. It consists of 90 segmented multi-object tracking videos with 14000 frames of multiple cities, multiple weathers, multiple times of day and multiple scene types. The segmented data is split into 60 training videos, 10 validation videos, and 20 testing videos.

5. Results

We evaluated the performance of two object tracking models, PCAN and MaskDINO+DeepOCSort, on the MOTChallenge benchmark. Fig. [4] shows the segmentation result generated by MaskDINO for all 81 object classes, which we will further filter to only detect the 8 classes required by BDD100k.

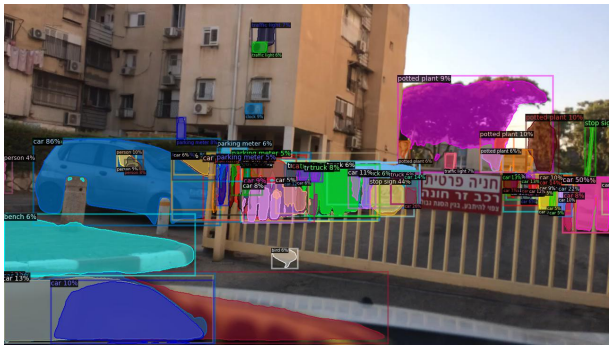

Figure 4. Segmentation result of MaskDINO

Figure 5. Images 7 frames apart showing tracking using the MaskDINO+DeepOCSort algorithm

Table [1] displays the performance metrics for PCAN and MaskDINO+DeepOCSort on the benchmark. We observed that MaskDINO+DeepOCSort outperformed PCAN in terms of MOTSP for all classes except Motorcycle, where PCAN scored slightly higher. Furthermore, MaskDINO+DeepOCSort achieved significantly higher MOTSP scores for classes such as Pedestrian+Rider and Car compared to PCAN. These results suggest that MaskDINO+DeepOCSort is a better model for object tracking, likely due to its ability to leverage more contextual information.

We provide the tracking results generated by MaskDINO+DeepOCSort in Fig. [5], which shows consecutive images from the same video, demonstrating that the algorithm can correctly tag objects to the same

	PCAN		MaskDINO+DeepOCSort	
Class	MOTSP	IDF1	MOTSP	IDF1
Ped+Rider	75.7	45.8%	81.1	44.0%
Car	85.1	73.2%	87.4	77.8%
Truck	86.5	55.6%	91.0	58.8%
Bus	87.5	62.2%	90.3	60.6%
Motorcycle	62.0	26.2%	51.8	25.4%
Bicycle	70.1	50.7%	82.0	44.2%
OVERALL	58.4	39.5%	60.4	40.2%

Table 1. Performance comparison of PCAN and MaskDINO+DeepOCSort on the MOTSChallenge benchmark.

Segmentation Metric	Tracking model	MOTSP	IDF1
PCAN		58.4	39.5%
YOLOv8-segm*	DeepOC SORT	55.3	37.6%
MaskDINO	DeepOC SORT	60.4	40.2%
	StrongSORT	58.7	39.4%
	OCSORT	57.7	35.3%

Table 2. Performance comparison of different methods tried on MOTSChallenge benchmark.

ID across multiple frames. In Table [2], we compared the performance of MaskDINO+DeepOCSort against other state-of-the-art object tracking methods. The results showed that MaskDINO+DeepOCSort achieved a higher MOTSP score than DeepOC SORT and YOLOv8-segmentation, and a similar score to StrongSORT. While MaskDINO+OCSort achieved a slightly lower MOTSP score than MaskDINO+DeepOCSort, it achieved a higher IDF1 score.

Our results indicate that MaskDINO+DeepOCSort is an effective model for object tracking, outperforming PCAN on the majority of object classes and competing with other state-of-the-art methods. The ability to leverage contextual information appears to be a key factor in the superior performance of MaskDINO+DeepOCSort, and its tracking results demonstrate its potential in real-world applications.

6. Discussions and Conclusions

Our motive in this research was to develop an algorithm that beats PCAN, the current best multi-object tracking and segmentation model for BDD100K dataset. We attempted a modular approach for the same where we choose MaskDINO as a segmentation model and experiment with DeepOCSort, StrongSORT and OCSORT as our tracking algorithms. We expect this best of both worlds to perform better than any previous end-to-end or modular approaches tried, given the individual efficiency of each approach.

Our results align with our expectations. Initially, we had to handle the bottleneck that our segmentation model MaskDINO did not consider rider as a separate class while BDD100k has it. Hence, in the evaluation phase PCAN was succesfully classifying Rider while MaskDINO was failing to do so which affected our overall score. To resolve this issue, we merged the classes Rider and Pedestrian and considered it same as the person class in MaskDINO. This resulted in considerable decrease in the PCAN performance whereas MaskDINO+Sort algorithms performed similar. This change mainly accounts to all false positives for Rider and Pedestrian now getting considered as true positives in the combined class. However, this provides a comparable evaluation for our method with PCAN as discussed in the result section. It is also quite evident from our results that our methods clearly outperforms PCAN and Yolov8-segmentation for detection of most of the classes. The ID switches for our model is almost equal to that of PCAN.

Further, handling the ID switches more effectively while any object moves out of the frame is still a concern. One approach to solve this can be to develop better appearance model based on ViT transformers for the tracking algorithm that handles any occlusion in the image and also any object moving out of the frame and can re-identify them successfully. Overall, this research also proves the effectiveness of modular approaches over end to end ones for this kind of combined objectives like Multi-Object Tracking and Segmentation.

7. Contributions

Harsh Sharma:(Segmentation and Evaluation) Implementation of MaskDINO for segmentation and modification of the dataset labels for PCAN and evaluation of PCAN with modified classes

Kathakoli Sengupta:(Multi-object Tracking and Evaluation) Extraction of MaskDINO labels as the detection module to tracking algorithms and modifications of the labels for our method and evaluation of MaskDINO+DeepOCSort, StrongSort and OCSort

References

[1] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking, 2023. 4

[2] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation, 2019. 2

[3] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again, 2023. 3

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[6] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification, 2021. 2

[7] Lei Ke, Xia Li, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Prototypical cross-attention networks for multiple object tracking and segmentation. In *Advances in Neural Information Processing Systems*, 2021. 4

[8] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, 2022. 2, 3

[9] Shuo Liu. Object trajectory estimation using optical flow. all graduate theses and dissertations. 462, 2009. 2, 3

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015. 2

[11] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification, 2023. 3, 4

[12] Leilei Rong, Yan Xu, Xiaolei Zhou, Lisu Han, Linghui Li, and Xuguang Pan. A vehicle re-identification framework based on the improved multi-branch feature fusion network. *Scientific Reports*, 11(1):20210, 2021. 2

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[14] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017. 2, 3

[15] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 4

[16] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 2

[17] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. VehicleNet: Learning robust visual representation for vehicle re-identification. *IEEE Transactions on Multimedia*, 23:2683–2693, 2021. 2

A. Appendix A

The section lists the repositories used in this project that has the architectures and libraries used other than the ones listed in the zip file.

PCAN: https://github.com/SysCV/pcan

This repository contains the PCAN scripts used for training a PCAN model on BDD100k dataset and evaluation scripts for getting test scores for all individual classes.

Yolov8tracking: https://github.com/mikel-brostrom/yolov8_tracking

This repository contains Yolov8 based detection and DeepOCSort, OCSort and Strongsort based tracking algorithm scripts.

MaskDINO: https://github.com/IDEA-Research/MaskDINO

This repository contains MaskDINO architecture and its implementation scripts.

Modified scripts are added in the zip file.