Mix and Match: ByteTrack with DETR

Sureen Heer
Stanford University
sureen@stanford.edu

Simba Xu
Stanford University
simbaxu@stanford.edu

Alice Ku
Stanford University
aliceku@stanford.edu

Abstract

Multi-object tracking (MOT) is an important task in computer vision, with applications in number of fields from surveillance to analyzing hospital interactions. MOT involves simultaneous detection and tracking of objects in video sequences. There are many challenges in MOT including occlusion, varying movements and presence of objects, and inter-object interactions. From our research, several models, like ByteTRACK, use YOLOX, a Faster R-CNN architecture to do object detection and then apply their tracking algorithm to associate the objects between frames. For our project, we aim to understand the impact of applying transformers to MOT. We have three models: the baseline ByteTrack model with the YOLOX detector, the ByteTrack model with a DETR detector, and the Trackformer with Deformable DETR that uses the transformer architecture end to end, for both detection and tracking. We hoped to see improvement in performance from the ByteTrack with YOLOX to ByteTrack with DETR, but our results showed that ByteTrack with YOLOX, with a MOTA score of 76.4%, outperformed the other two models, with ByteTrack with DETR achieving 68.2% and Trackformer with Deformable DETR achieving 74.1%.

1. Introduction

Multi-Object Tracking (MOT) is a crucial task in computer vision that involves tracking multiple objects as they move through a series of frames in a video. This problem is important because it has a wide range of applications, including autonomous driving, surveillance, and sports analytics. Our motivation for pursuing this problem stems from the growing demand for more robust and accurate MOT systems. In our approach, we explore three different models for MOT. The first model is the ByteTrack baseline model, known for its sophisticated association algorithm and handling of low-confidence detections. The second model enhances the ByteTrack baseline by replacing its YOLOX detector with the DETR (DEtection TRansformer) detector, with hope of leveraging the benefits of transformer-based detection for better performance. The third model is the Trackformer model with Deformable DETR, which uses transformers to performs detection and tracking. The input to our algorithms consists of video sequences where each frame contains multiple objects to be tracked. For the ByteTrack baseline and ByteTrack with DETR models, the input is processed using their respective detectors (YOLOX or DETR) to identify and localize objects in each frame. These detections are then fed into the ByteTrack algorithm for tracking. For Trackformer, the input video is directly processed by the transformer-based model which is then fed to a transformer-based tracker. The output of our algorithms is a set of trajectories for each tracked object across the video frames.

2. Related Work

2.1. Object Detection

Object detection is the backbone to multi-object tracking. There are two-stage detectors such as Faster R-CNN [8] that predict bounding boxes with respect to regional proposals and single-stage detectors like YOLOX [6] and CenterNet [12] that make predictions with respect to anchor boxes or possible object centers (anchor-free methods). YOLO models are suitable for real-time applications, balancing speed and accuracy. YOLOX is different from the earlier YOLO models as it detects in an anchor-free manner and uses advanced detection techniques such as decoupled heads and SimOTA for label assignment [6]. DETR, Detection Transformer, turns object detection into a direct set prediction problem, by using self-attention mechanism in transformer encoders and decoders to capture global dependencies and eliminate the need for heuristics such as anchor-boxes and Non-Maximum Supression (NMS) [3].

2.2. Tracking Algorithms

Tracking algorithms take in as input the detected bounding boxes and associated information like class labels and confidence scores from the detectors described above. A track is initialized for each object when it is detected for the first time. For each track, the tracking algorithm, tra-

ditionally SORT or DeepSORT, predicts the object's next position in the next frame, typically using a Kalman filter [2, 10]. These predictions are then matched with the new detections provided by the detectors, generally using metrics like Intersection over Union (IOU). The Hungarian algorithm is often used to ensure that the detection is assigned to the closest track prediction [11]. After data association, the tracks are updating, correcting predictions based on new detections. Tracks that are not associated with detection due to occlusion, for example, are kept for a few frames for re-identification (to reduce identity switches) but if they remain unassociated for several consecutive frames, then those tracks are terminated, with the assumption that the object has left the scene or is no longer detectable. Compared to traditional tracking algorithms like SORT and DeepSORT, ByteTrack utilizes hierarchical data association to consider both high confidence and low confidence detections to improve the ability to track objects that are intermittently detected with lower confidence [11].

2.3. Transformer Models in Computer Vision

Transformers, initially introduced in the field of natural language processing, has revolutionized language modeling through self-attention mechanisms and positional encodings, capturing long-range dependencies with efficient parallelization [9]. Inspired by this success, researchers have extended transformers to computer vision tasks, giving rise to Vision Transformers (ViTs), which apply transformer architectures directly to image or video data for tasks like image classification and object detection [5]. Models like DETR [3] and Trackformer [7] use the transformer architecture for object detection and tracking, demonstrating their effectiveness in end-to-end approaches to multi-object tracking tasks. DETR pioneered direct object detection by predicting object queries and their corresponding bounding boxes. Deformable DETR [13] further improved DETR by only attending to a small set of sampling points around a reference to mitigate issues such as slow convergence and limited feature spatial resolution. Trackformer handles both object detection and tracking within a single unified transformer framework.

2.4. Comparative Analysis

Previous work in MOT have focused on improving object detection or tracking algorithms. There has been exploration of combining advanced detection models with robust tracking algorithms, the specific combination of using DETR's powerful detection capabilities with ByteTrack's tracking algorithm has not been explored. We use as baseline the ByteTrack model with YOLOX. Compared to ByteTrack with YOLOX, the detections will hopefully be more accurate improving the input for tracking. The important objective is, however, to see how varying level of incorporation of transformer architecture into MOT task affects the accuracy. Therefore, we also run the Trackformer model with Deformable DETR.

3. Data

3.1. MOT17

The MOT17 dataset is a popular benchmark in multi-object tracking and comes from the Multiple Object Tracking Challenge, which is a benchmark for MOT algorithms. The dataset is publicly available at the MOTChallenge website [1]. The dataset consists of video sequences capturing different real-world scenarios such as crowded streets and occluded objects. Each video sequence contains hundreds to thousands of frames and each is labeled with bounding box annotations specifying the position, size and ID of each object in each frame. There are 14 video sequences, 7 for training and 7 for testing. For each set, there are 6 videos that have resolution 1920x1080 and 1 that has resolution 640x480. We use this dataset to train and test all three models. We pre-processed the dataset by converting the data into COCO format, as this is the format accepted by the detection models we are using. Figure 1 displays the ground truth annotations for a single frame of a training sample.

3.2. CrowdHuman

CrowdHuman is a benchmark dataset for evaluating detectors in crowded scenarios and is publicly available at the CrowdHuman website [4]. It contains 15000 training images, 4370 validation images and 5000 test images. There is a total of 470K human instances and 23 persons per image, accounting for various types of occlusions. Each instance is annotated with head and full-body bounding-boxes. The typical resolution of images is 1920x1080. This dataset is used for the pretrained DETR and YOLOX detectors that we use in our tracking models. The dataset is pre-processed to be in the COCO format as well. Figure 2 displays the ground truth annotations for an sample image.

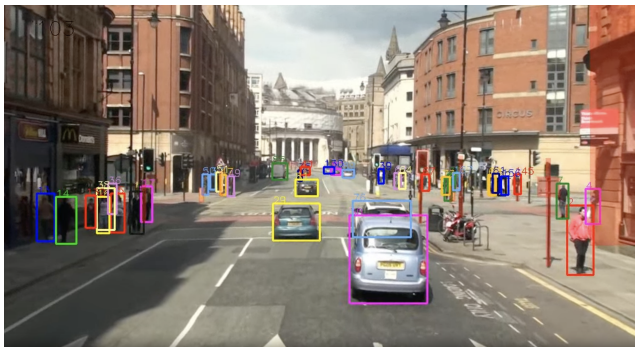

Figure 1. Ground truth annotations for frame 103 of video sequence MOT17-13-FRCNN.

Figure 2. Ground truth annotations for an image from CrowdHuman dataset.

4. Methods

We will now describe the three models we will be running, with increasing incorporation of Transformer architecture.

4.1. ByteTrack with YOLOX detection

The main objective of ByteTrack is to secure the information from bounding boxes with low confidence scores, especially the bounding boxes of occluded objects. After detection, ByteTrack sort bounding boxes to high-confidence boxes, low-confidence boxes, and background boxes. Background boxes are abandoned immediately, but both high-confidence and low-confidence boxes will be kept. Keeping low-confidence boxes improves tracking consistence since occluded objects, although having lower confidence scores, still contain information that could help the association process between previous and next frames [11].

BYTE, the detection association of ByteTrack takes a video sequence with an object detector as input, and outputs the tracks of the video and the bounding boxes and identity of the detected objects.

With a detection score threshold, BYTE sorts the bounding boxes to high or low confidence according to the scores obtained through the detector. Then, they apply Kalman filter to predict the new locations of each track. Afterwards, there are two association processes. The first similarity is computed between high-confidence detection and the predicted boxes of tracks. The second similarity is computed between low-confidence boxes and the unmatched tracks from the previous step. The unmatched tracks will be preserved, and if they exit more than a certain number of frames, they will be deleted. Lastly, they output the bounding boxes and the identities of the tracks in current frames.

In the original ByteTrack paper, ByteTrack: Multi-Object Tracking by Associating Every Detection Box, the authors adopted YOLOX as there detector [6]. This version of framework, evaluated on the MOT17 dataset, is the baseline in our project.

YOLOX is an improved version of the YOLO detector. As seen from Figure 3, it replaced coupled head with decoupled head, which improves the converging speed and helps with the end-to-end architecture. They also simplifies the training and decoding processes by removing anchors.

4.2. ByteTrack with DETR detection

DEtection TRansformer (DETR) is a detection framework based on a conventional CNN and encoder-decoder [3]. It simplifies its detection process by dropping components that encode prior knowledge. It also detects objects in sets and predicts all objects at once. Specifically, DETR utilizes bipartite matching loss and transformer parallel decoding, which increase its efficiency. Figure 4 shows the framework of DETR.

DETR calculates its loss through three steps. First, it computes bipartite matching between the predicted and ground truth objects, i.e., the optimal assignment. It does so by searching for a permutation of N predictions. 1 is the optimal assignment, in which $\mathcal{L}_{\text{match}}$ is the pair-wise matching cost between the ground truth y_i and the prediction $\hat{y}_{\sigma(i)}$.

$$\hat{\sigma} = \underset{\sigma \in N}{\operatorname{argmin}} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \quad (1)$$

Then, the Hungarian loss is calculated for the pairs matched in 1, which is a linear combination of the negative log-likelihood for class prediction and box loss. To simplify the implementation, they make box predictions directly, while other researchers make predictions according to initial guesses. The box loss is a linear combination of the IoU loss and l_1.

4.3. Tracking with Transformer with Deformable DETR Detection

This method replaces both Detection task and Tracking task with transformer architecture, implemented with modification upon TrackFormer [7]. This method follows the following structure: Frame-Level Feature Extraction: Using Deformable DETR Resnet 50 to extract Feature per frame in the video sequence. Frame Feature Encoding: Applying self-attention in a Transformer encoder. Query Decoding: Utilizing self- and encoder-decoder attention in a Transformer decoder. Mapping Queries: Converting queries to box and class predictions. Objects are implicitly represented in the decoder queries, which act as embeddings that the decoder uses to output bounding box coordinates and class predictions. The decoder employs two types of attention mechanisms: Self-Attention: Applied over all queries, this mechanism enables joint reasoning about the objects within a scene; Encoder-Decoder Attention: This provides

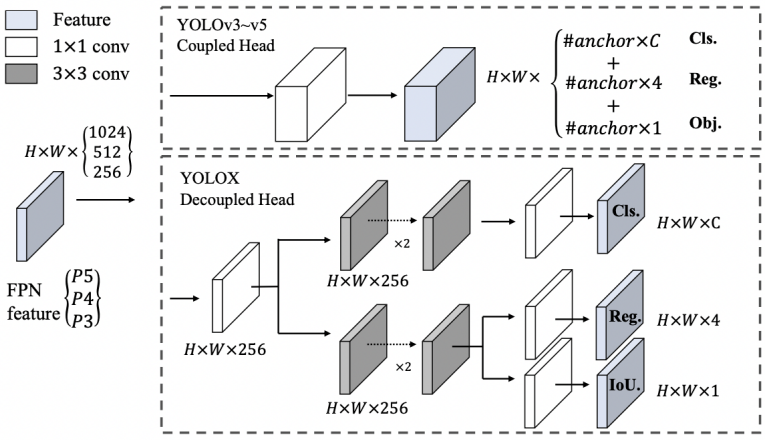

Figure 3. Comparison between coupled and decoupled head frameworks for YOLO.

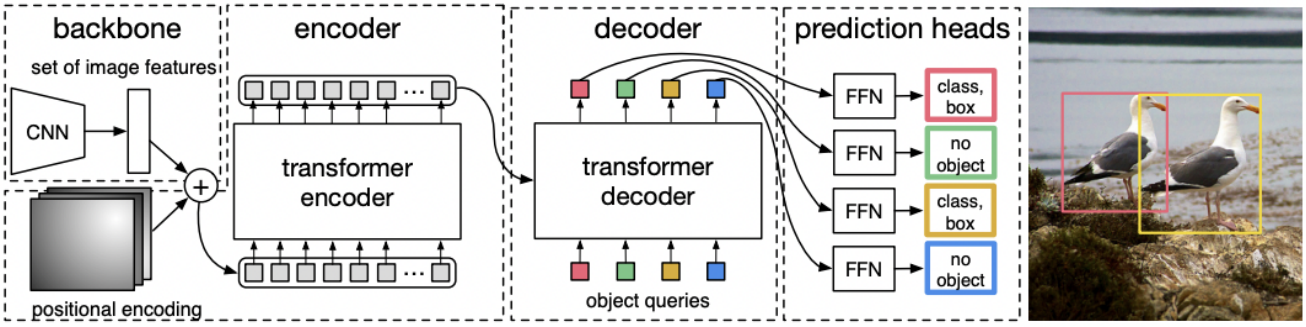

Figure 4. DETR framework. DETR uses CNN as backbone and uses the encoder-decoder method to predict parallely.

the queries with global access to the visual information encoded in the features.

Tracking with attention with queries

The set of output embeddings in this method is initialized with two types of query encodings: Static Object Queries: These queries allow the model to initialize tracks at any frame of the video. Autoregressive Track Queries: These queries are responsible for tracking objects across frames.

When new objects appear in the scene, they are detected by a fixed number of N_{object} output embeddings, each initialized with a static and learned object encoding, referred to as object queries. Each object query is designed to predict objects with specific spatial properties, such as bounding box size and position. The decoder's self-attention mechanism utilizes these object encodings to avoid duplicate detections and to understand the spatial and categorical relationships between objects.

Track queries follow objects through a video sequence, retaining their identity information while adapting to their changing positions in an autoregressive manner. Each new object detection initializes a track query with the corresponding output embedding from the previous frame. The Transformer encoder-decoder performs attention on frame features and decoder queries, continuously updating the instance-specific representation of an object's identity and location within each track query embedding. Self-attention over both query types allows for detecting new objects while avoiding re-detection of already tracked ones. The Process is as follows:

- **Initialization:** In frame $t = 0$, initial detections spawn new track queries for corresponding objects, which follow these objects to subsequent frames.
- **Object Queries:** N_{object} object queries (white) are decoded to output embeddings for potential track initialization. Each valid object detection $\{b_0^0, b_1^0, \ldots\}$ with a classification score above σ_{object} (not predicting the background class) initializes a new track query embedding.
- **Track Queries:** At any frame $t > 0$, track queries initialize additional output embeddings associated with different identities (colored). The combined set of $N_{\text{object}} + N_{\text{track}}$ output embeddings is initialized by learned object queries and temporally adapted track queries.

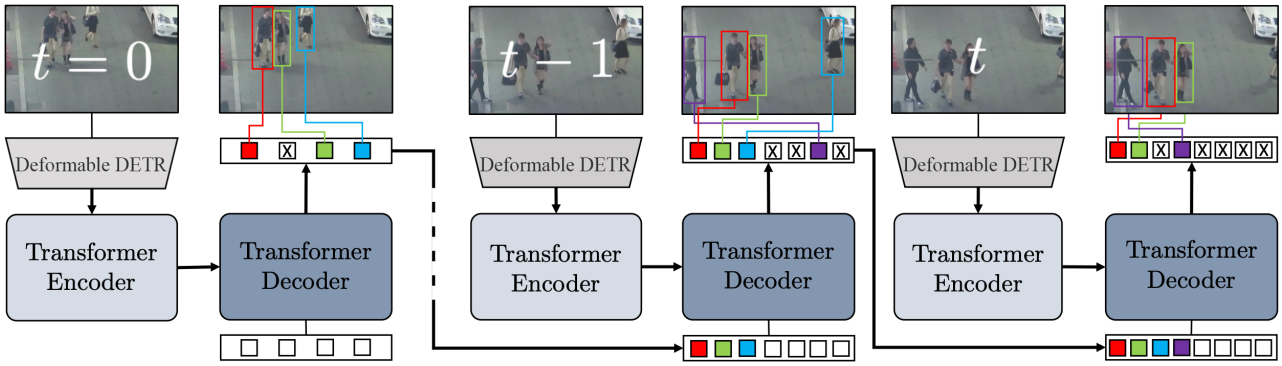

Figure 5. Transformer Tracking model adapted and modified from TrackFormer

Once again, non-maximum suppression (NMS) is used to remove duplicate bounding boxes.

Tracker enables short-term re-identification of track queries using an attention-based process. Removed track queries are kept for up to $T_{\text{track-reenter}}$ frames and can be re-activated if their classification score exceeds $\sigma_{\text{track-reenter}}$.

For feature extraction per frame, Deformable DETR is used for its flexibility and robustness which makes adapting it to the tracker easier as well as improved accuracy on small objects due to deformable attention, which allows the model to focus on a small, adaptive set of sampling points around a reference point. This is beneficial to large crowd public detection tasks.

5. Experiments, Results, and Discussion

5.1. Experiments & Evaluation Metrics

We used pretrained models from YOLOX and DETR as well as Deformable DETR for detection, since those rely on large training which would be computationally exhaustive to retrain. Deformable DETR methods are reimplemented with adjustments for tracking tasks described in tracking transformer model. DETR detections are reformulated and calculated with object confidence, class confidence, and class prediction. Since all datasets are preprocessed to align with COCO format, COCO class number of 92 is used with human label = 1. Similar to YOLOX post-processing, DETR detections are converted from

$$[x_{center}, y_{center}, w, h]$$

to

$$[x_0, y_0, x_1, y_1]$$

. Different from YOLOX post-processing, which follows a similar raw output format

$$[x_{center}, y_{center}, w, h]$$

but values are absolute pixel values as opposed to relative to image values for DETR. NMS is applied to both YOLOX and DETR outputs with nms_threshold =0.45 and objects are filtered out with confidence_threshold = 0.7. Deformable DETR does not need NMS applied. Detection outputs are then fed into trackers.

Multi-object tracking (MOT) performance is evaluated using a variety of metrics. Identification metrics like IDF1, IDP (Identity Precision), and IDR (Identity Recall) assess the accuracy of identity matching by combining precision and recall. Detection and tracking metrics include Recall (Rcll) and Precision (Prcn), which measure the ability to detect all objects and the accuracy of these detections, respectively. Ground truth and detection metrics such as GT (Ground Truth), MT (Mostly Tracked), PT (Partially Tracked), and ML (Mostly Lost) quantify the completeness of tracking trajectories. Error metrics like FP (False Positives), FN (False Negatives), IDs (ID Switches), and FM (Fragmentations) indicate various tracking errors. Overall metrics like MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision) provide comprehensive performance evaluations, with MOTA considering false positives, false negatives, and ID switches, and MOTP measuring the precision of object positions. Identity-based metrics such as IDt (ID Transfer), IDa (ID Ambiguity), and IDm (ID Matching) further assess the consistency and accuracy of object identity assignments. High values in metrics like IDF1, IDP, IDR, Rcll, Prcn, MT, and MOTA indicate better tracking performance, while low values in FP, FN, IDs, and FM are desirable.

$$\text{MOTA} = 1 - \frac{\text{FP} + \text{FN} + \text{IDs}}{\text{GT}} \tag{2}$$

where FP is the number of False Positives, FN is the number of False Negatives, IDs is the number of identity switches, and GT is the number of ground truth objects.

5.2. Results

Please refer to Table 1 and Figure 6.

Table 1. Methods MOTA evaluated on MOT17 Test video sequences for public detection

	ByteTrack w/ YOLOX	**ByteTrack w/ Detr**	**Transformer Tracking w/ Deformable Detr**
MOT17-13-FRCNN	77.4%	45.7%	60.3%
MOT17-10-FRCNN	70.0%	59.0%	67.0%
MOT17-02-FRCNN	87.6%	62.8%	89.7%
MOT17-05-FRCNN	76.7%	79.0%	71.3%
MOT17-11-FRCNN	53.4%	63.3%	64.7%
MOT17-09-FRCNN	69.8%	63.2%	75.5%
MOT17-04-FRCNN	83.2%	83.0%	83.6%
OVERALL	76.4%	68.2%	74.1%

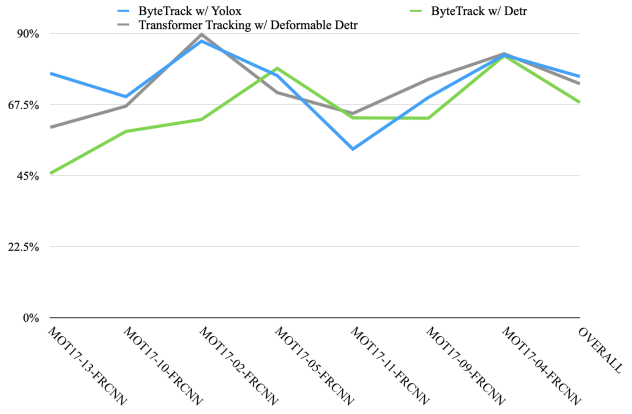

Figure 6. Methods MOTA Evaluation on Various MOT17 Test Sets

5.3. Discussion

In this work, we evaluated the performances of three MOT models with the MOT17 dataset. Overall, the MOTA score of ByteTrack with YOLOX outperforms those of ByteTrack with DETR and Transformer Tracking with Deformable DETR, while the later has a better performance. The performances of ByteTrack with YOLOX went down a bit compared to the original paper, which had a MOTA score of 80.3 on the MOT17 leaderboard.

One key could be the issue of compatibility between YOLOX and the ByteTrack tracker, which results in a high overall performance. This suggests that YOLOX's detection capabilities align well with ByteTrack's tracking approach, leading to effective object tracking.

In contrast, replacing YOLOX with DETR in ByteTrack leads to a notable decline in performance. This drop could indicate potential compatibility issues between DETR's detection outputs and ByteTrack's tracking algorithm, since DETR still performs well on Recall and Precision metrics, which indicates decent detection. Despite DETR's robust detection capabilities, its representation might not integrate seamlessly with the ByteTrack tracker. With better fine-tuning or combined training, it is possible that accuracy improves.

Improvement is observed when both detection and tracking are handled by transformer-based methods in the Transformer Tracking configuration. This approach leverages the strengths of Deformable DETR for detection and a transformer tracker, resulting in a more cohesive and effective framework. The performance of this configuration suggests that transformers offer a robust and compatible solution for both detection and tracking, enhancing overall tracking accuracy. The deformable nature of DETR likely contributes to its robustness, handling various detection scenarios more effectively. However, the baseline ByteTrack still leads in terms of MOTA performance.

6. Conclusion and Future Work

In this report, we evaluated three different MOT models: the ByteTrack baseline model, the ByteTrack model with the DETR detector, and Transformer Tracking. Our goal was to enhance MOT performance by integrating advanced detection models and tracking algorithms, addressing the limitations of traditional methods in dealing with occlusions and maintaining track consistency in crowded scenes. Our experiments found that the ByteTrack model with the YOLOX detector outperformed the other two models. This model's success can be attributed to YOLOX's strong performance in object detection, combined with ByteTrack's robust tracking capabilities, resulted in better accuracy and consistency in tracking multiple objects across video frames. Surprisingly, the ByteTrack model with the DETR detector did not perform as well as expected. Despite DETR's promise in providing accurate and contextual object detection through its transformer-based architecture, the integration with ByteTrack did not yield the anticipated improvements. This could be due to several factors, including possible mismatches in detection and tracking strategies, or the need for further tuning and optimization of the combined system. Trackformer demonstrated competitive performance but still fell short of the ByteTrack with YOLOX model as it may require additional refinement and

optimization. For future work, given more time, team members, and computational resources, we would like to investigate the integration of DETR with ByteTrack more thoroughly, focusing on optimizing the interaction between detection and tracking stages, incorporate data augmentation for challenges like occlusions and varying lighting conditions, use different evaluation datasets for more rigorous testing. In conclusion, our study highlights the potential of combining advanced detection models with robust tracking algorithms to improve MOT performance. While the ByteTrack model with YOLOX emerged as the best performer, further research and optimization could unlock additional improvements, paving the way for more accurate and reliable MOT systems in various applications.

7. Contributions

Sureen: Formulated idea of integrating DETR with ByteTrack and worked on all sections of the report, especially Abstract, Introduction, Related Work, Data, Evaluation Metrics, Conclusion/Future Work and References.
Simba: Training and experimenting with ByteTrack to obtain baseline results, implemented ByteTrack with DETR, implemented Transformer Tracking with deformable DETR, conduct experiment and obtained results. Worked on the following sections in the report: Methods and Experiments.
Alice: Conducted literature review for milestone, debugged models, and worked on the following sections in the report: Methods.

References

[1] MOTChallenge: Multiple Object Tracking Benchmark. https://motchallenge.net/. 2

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, Sept. 2016. 2

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers, 2020. 1, 2, 3

[4] CrowdHuman Authors. CrowdHuman: A Benchmark for Detecting Human in the Crowd. https://www.crowdhuman.org/. 2

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[6] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. Yolox: Exceeding yolo series in 2021, 2021. 1, 3

[7] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer. Trackformer: Multi-object tracking with transformers, 2022. 2, 3

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 1

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. 2

[10] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric, 2017. 2

[11] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang. Bytetrack: Multi-object tracking by associating every detection box, 2022. 2, 3

[12] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points, 2019. 1

[13] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection, 2021. 2