People tracking accuracy improvement in video by matching relevant trackers and YOLO family detectors

H. Quan [1], G. Ma [2], Y. Weichen [2], R. Bohush [3], F. Zuo [4], S. Ablameyko [1,5]
[1] Belarusian State University, 220030, Minsk, Republic of Belarus, Nezavisimosti av. 4;
[2] EarthView Image Inc., 313200, China Deqing County, Zhejiang, Keyuan Road 11;
[3] Polotsk State University, 211440, Novopolotsk, Republic of Belarus, Blokhin str. 29;
[4] Henan International Joint Laboratory of Theories and Key Technologies on Intelligence Networks,
Henan University, 450046, China, Kaifeng;
[5] United Institute of Informatics Problems, National Academy of Sciences of Belarus,
220012, Minsk, Republic of Belarus, Surganov str. 6

Abstract

The tracking-by-detection paradigm is widely used for people multi-object tracking tasks. Up to now, there exist many detectors and trackers, many evaluation benchmarks, which necessitates the use of relatively uniform estimation methods and metrics. It leads to necessity to choose better combined models of detectors and trackers. To solve this task, we developed a comprehensive performance evaluation methodology for estimation of people tracking accuracy and real-time by using different detectors and trackers. We conducted experiments by choosing the official pre-trained models of YOLOv5, YOLOv6, YOLOv7, YOLOv8 with representative BoTSORT, ByteTrack, DeepOCSORT, OCSORT, StrongSORT trackers under two benchmarks of MOT17 and MOT20. Detailed metrics in terms of error and speed such as higher order tracking accuracy and frames per second were analyzed for the combinations of detectors and trackers. It is concluded that the OCSORT+YOLOv6l model has the best comprehensive performance and the combination of OCSORT and YOLOv7 has the best average performance under MOT17 and MOT20.

Keywords: YOLO family detectors, tracking-by-detection, multi-object tracking, scoring function, comprehensive performance, video surveillance.

Citation: Quan H, Ma G, Weichen Y, Bohush R, Zuo F, Ablameyko S. People tracking accuracy improvement in video by matching relevant trackers and YOLO family detectors. Computer Optics 2024; 48(5): 734-744. DOI: 10.18287/2412-6179-CO-1422.

Introduction

Multi-object tracking (MOT) is a very important research area in computer vision, which is widely used in video surveillance, intelligent security and autonomous driving. Currently the tracking objects are mainly people or vehicles, and people are typically non-rigid objects. They are more difficult to track than rigid objects, therefore have greater research value.

These practical applications require real-time and accurate people tracking models, which makes the Tracking-by-Detection (TBD) paradigm a good fit for such needs. Tracking-by-Detection model usually consists of a detection module and a tracking module, which address the three main tasks of detection, localization, and association. Recently, many high-performance detectors, trackers and specialized people re-identification models have emerged.

However, existing tracking models often use only one type of detector, or only give the evaluation and comparison of each metric under some benchmark. But in real applications, the models are used in complex scenarios. Therefore, we would like to know which combination of models is more general or which combination has better general comprehensive performance under different benchmarks.

Current tracking models with excellent performance include ByteTrack [1], BoTSORT [2], StrongSORT [3] and others. StrongSORT is the easier of these models to use and deploy. Just as StrongSORT does, many researchers combine YOLO detectors with these trackers and will even integrate Re-ID[4] models to improve tracking performance.

YOLO has a large family of detectors. The YOLO [5] series detectors have lightweight, fast and accurate performance and are very easy to integrate into the tracking model. In addition to the consistent naming from the first version of YOLO to YOLOv8, there are various named variants, as well as some variants that have been improved accordingly, but are not named. At the same time, different trackers have their own different architectures and inconsistent performance under different evaluation benchmarks. Naturally, we would like to know exactly what effect these detector models have on the trackers. In [6], the combination of YOLOv5 with DeepSORT [7] is replaced with YOLOv7 [8]. In [9], we find better algorithms using different combined models of YOLO and StrongSORT. The experimental results showed that the multi-object tracking accuracy was improved. Although, whether the improved detector performance had a positive effect on all the metrics of the model seems to be not fully determined.

Therefore, this paper focus on different combinations of YOLO detectors and trackers for tracking people in videos, selecting metrics such as Higher Order Tracking Accuracy (HOTA) [10] and evaluating them in MOT benchmarks in order to find a better solution in terms of accuracy and speed. The YOLOv5, YOLOv6 [11], YOLOv7, and YOLOv8 detectors, are combined with BoTSORT, ByteTrack, DeepOCSORT [12], OCSORT [13] and StrongSORT trackers, and their performance is evaluated separately. We have chosen two metrics, frames per second (FPS) and HOTA, selected MOT17 [14] and MOT20 [15] benchmarks based on realistic application requirements. We proposed a cross-benchmark comprehensive performance evaluation method, which scores different combined models and helps us to select the appropriate combined algorithm for tracking people in videos. This may also provide an evaluation reference for proposing universal tracking models in the future.

1. Real-time YOLO object detection models

YOLOv5, YOLOv6, YOLOv7 and YOLOv8 have been chosen as detector models.

YOLOv5's architecture consists of three parts, namely Backbone, Neck and Head. Backbone is mainly used for extracting features. It consists of a series of ConvModule and CSPLayer. Then uses SPPFBottleneck. Neck uses a PAFPN that can shorten the path of lower and topmost feature information to achieve efficient fusion of features. Head part is used for regression prediction and calculation of CIoU and BCE Loss. Although the author only provides the code in the repository (https://github.com/ultralytics/yolov5).

YOLOv6 is proposed by Meituan, and the basic architecture is the same as YOLOv5. Likewise, it has been continuously optimized and improved, and has been updated to version 4. However, important architectural changes are presented in YOLOv6 v3.0 [16]. The main ones are, using BiC module in Neck to improve the localization accuracy and SimCSPSPPF module to improve the speed. The advantages of anchor-based and anchor-free paradigms are obtained using the AAT training strategy. The DLD self-distillation technique is also used in small models for enhanced performance [11].

YOLOv7 is based on YOLOv5 and YOLOR for improvement. The overall architecture is still consistent with YOLOv5. The main innovations proposed in this model are Extended efficient layer aggregation networks, Model scaling for concatenation-based models and planned re-parameterized convolution. In addition, the batch normalization layer is directly connected to the convolution layer in the conv-bn-activation topology. EMA is used as the final inference model. Its most prominent feature is the use of YOLOR's implicit knowledge technique, which combines implicit knowledge with convolutional feature maps in addition and multiplication manner [9]. This makes the model much more accurate, but the speed still suffers somewhat. Perhaps integration into our tracker might lead to new discoveries.

YOLOv8 is positioned as a unified framework integrating detection, tracking, segmentation, classification and pose estimation. Compared to YOLOv5, it uses the C2f module to replace the original C3 module. The anchor-free paradigm is adopted, using decoupled head design, each decoupled head consists of Bbox and class loss respectively. The sample matching uses the matching method of the Task-Aligned Assigner. As with YOLOv5, their repository is in https://github.com/ultralytics/ultralytics. However, the series of improvements led to a high speed and accuracy improvement of the model, which attracted our interest.

2. Multi-object tracking models

There known quite many various trackers. We selected the better-performing models in the benchmarks suitable for people tracking.

ByteTrack proposes a new data association method BYTE. It exploits the similarity between detection frames and tracking trajectories to reduce missed detections and improve track coherence by removing the background from low-scoring detection results while retaining high-scoring detection results and digging out the real objects (obscured, blurred, and other difficult samples). It uses YOLOX as a detector. Only the Kalman filter is used to predict the position of the tracking track of the current frame in the next frame, and the IoU between the predicted frame and the actual detected frame is used as the similarity between the two matches, which is done by the Hungarian algorithm. However, not using ReID features to calculate the appearance similarity also achieved good results [1].

BoTSORT model is based on the ByteTrack improvement, which is available in versions with and without ReID. It has three main improvements. For Kalman filtering, an eight-tuple state vector is chosen to directly estimate the width and height of the bounding box. And accordingly, the process noise covariance Qk and the measurement noise covariance matrix Rk are modified. For Camera Motion Compensation, the global motion compensation technique is used. The image key points are first extracted, and then feature tracking is performed with translation-based local anomaly suppression using sparse optical flow. The affine transformation matrix is calculated using RANSAC and the predicted bounding box is transformed from the coordinate system of previous frame to the next frame, and then corrected and updated at KF. For the fusion mechanism of IOU+ReID, ResNest50 from the FastReID library is used as the backbone and BoT (SBS) is used as the baseline to train the ReID network. The classical TripletLoss is used for the loss function. Exponential Moving Average is used to update the trajectory state of the i-th prediction frame in the k-th frame. The matching output is performed using new appearance cost [2].

OCSORT stands for Observation-Centric SORT, which emphasizes the motion model without using appearance features. The authors argue that the SORT model has three limitations, being sensitive to state noise in high frame rate videos, amplifying temporal errors, and being estimation-centric, ignoring the role of observation. Observation-centric Online Smoothing (OOS), Observation-Centric Momentum (OCM), and Observation-Centric Recovery (OCR) strategies for the above limitations are proposed for robust tracking under occlusion and nonlinear motion. The OOS strategy performs online smoothing of parameters by observed virtual trajectories to fix cumulative errors in time intervals caused by untracked trajectory re-association. The OCM method reduces errors by adding velocity consistency terms to the cost matrix and using the observations associated with the track for the direction calculation. The OCR idea is that once a track remains untracked after the normal association phase, an attempt will be made to associate the last observation of the track with the observation of the newly emerged time step, facilitating the handling of occlusion situations [13].

DeepOCSORT is improved based on the OCSORT model that does not use object appearance features for matching. It is combined with OCSORT's OOS, OCM, and CMC modules, respectively, and applies CMC updates before the Kalman extrapolation step so that the prediction phase comes from the CMC correction state. A modified Exponential Moving Average is applied to dynamically incorporate the appearance information into the model. Adopting an adaptive increase in the weight of appearance features based on the discriminative power of appearance embedding [12].

StrongSORT uses YOLOX as the detection module. The main framework is two branches and a matching. One branch consists of ECC and NSA Kalman to compute the motion gating matrix. The other branch consists of BoT and EMA, fusing both motion and appearance cues to compute the cost matrix. Finally, Vanilla Matching is used. Two lightweight and easy-to-use algorithms are proposed. They are the appearance-free linking model for solving missing associations and Gaussian smooth interpolation for solving missing detections, respectively [3]. It has a high HOTA metric, but a relatively low speed.

3. The proposed methodology for people tracking accuracy improvement

The aim of our methodology is to choose the combination of corresponding detectors and trackers to have the best tracking characteristics. The proposed methodology is shown in Fig. 1.

We input video sequences with different benchmarks into the combined model, change different detection modules and tracking modules, and get tracking data. Then the tracking data is input into the evaluation model to derive the corresponding selected metric values. In the next step, the scoring formula is used for the corresponding metric values to derive the scores and rankings for each combination model. Finally, our optimal combined model algorithm is obtained based on the composite score.

Among the trackers used, BoTSORT, DeepOCSORT and StrongSORT use ReID pre-trained weights with the weight file “osnet_x1_0_msmt17.pt” [4]. ByteTrack and OCSORT do not use the ReID model that relies on performing appearance features for extraction and matching. The confidence threshold and the intersection over union threshold of the detector affect the detection results of the model. Setting the confidence threshold too small increases false detections and reduces precision, and also reduces the speed of the model.

Setting it too high will result in missed detections and lower recall.

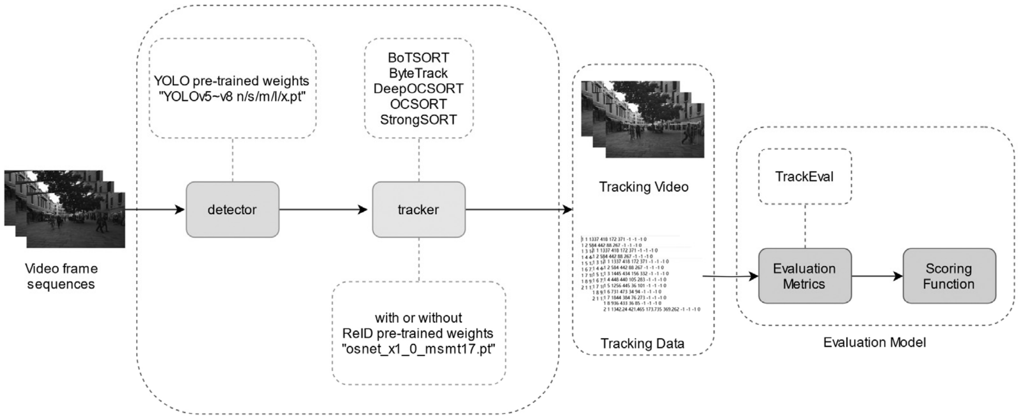

Fig. 1. Basic flowchart of the methodology

Setting the intersection over union threshold too small is prone to leakage detection, and setting it too large is prone to false detection. In practical application scenarios, there is a high degree of target overlap, many small targets, and different parameters of the pre-trained models, which are not conducive to a relatively objective comparison of each combined model. Therefore, the confidence threshold and the intersection over union threshold are usually set to 0.5 is a better choice to achieve a balance. So, we used a detection confidence threshold set to 0.5 and an intersection over union threshold set to 0.5. The pre-trained weight files for all detectors are the official files.

Since different MOT models have different parameters, architectures, training methods, data sets, etc., it is difficult to compare their performance without a unified standard. And researchers have contributed many open-source benchmarks in the field of multi-object tracking alone, such as MOT20, DanceTrack [17], SportsMOT [18], HiEve [19], etc. The benchmarks are used for different purposes. For example, MOTS [20] is mainly used for segmentation. We choose MOT17 and MOT20 benchmarks from four perspectives for our experiments. First, the tracking object is chosen to be people benchmark. Secondly, the benchmark with newer release is selected in time. Then, the benchmark with more complex background is selected for the application scenario. Finally, the benchmark that is commonly used by researchers is selected.

The selection of evaluation metrics is also very important for evaluating the performance of the tracker. We consider both the accuracy of the model and its speed from the perspective of practical applications. The main accuracy evaluation metrics are TrackmAP [21], VACE [22], Identity [23], CLEARMOT [24]. For the purpose of analyzing the performance of each module of the combined model, we choose detection accuracy (DetA), localisation accuracy (LocA) and association accuracy (AssA) [10]. DetA is detection Jaccard index averaged over localization thresholds. LocA is average localization similarity averaged over all matching detections and averaged over localization thresholds. AssA is association Jaccard index averaged over all matching detections and then averaged over localization thresholds.

For the purpose of considering the comprehensive performance of the combined model, we choose HOTA, that is geometric mean of detection accuracy and association accuracy averaged across localization thresholds. We decided to select Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP) [24] and IDF1 [23] with the corresponding DetA, LocA and IDF1 metrics to complement and validate each other. MOTA accounts for all object configuration errors made by the tracker, false positives, misses, mismatches, over all frames. MOTP measures the average overlap between correctly matched hypotheses and their respective objects, providing a measure of localization precision in multi-object tracking. IDF1 is the ratio of correctly identified detections over the average number of ground-truth and computed detections.

In general, we choose HOTA, LocA, DetA, AssA, MOTP, MOTA and IDF1 to evaluate the accuracy of the model. Then, FPS is chosen as the speed metric:

$$FPS = \frac{F}{Pt + Dt + Nt + Tt}, \tag{1}$$

where F is the total number of frames of a video, Pt, Dt, Nt and Tt are the pre-processing, detection, non-maximum suppression and tracking times of a video.

4. Experimental results

4.1. YOLOv5 detector and various trackers

Detailed evaluation results of YOLOv5 with different trackers under MOT17 is shown in Tab. 1. From this table we can see OCSORT+YOLOv5x has the highest DetA. But OCSORT+YOLOv5m has the highest MOTA. BoTSORT+YOLOv5x has the highest LocA and MOTP. StrongSORT+YOLOv5l has the highest HOTA and AssA. But StrongSORT+YOLOv5m has the highest IDF1.

Although the highest DetA and MOTA, HOTA and IDF1, do not belong to the same model, the difference between the metric values is not very large, probably due to the different calculation formulas. For DetA and LocA, the combination of each tracker with different YOLOv5 detectors is consistent with YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, except for the combination of ByteTrack with YOLOv5. The AssA of the combined model is not regular under the YOLO detection module using the same parameter size. The HOTA metrics, on the other hand, are in the order of StrongSORT, OCSORT, BoTSORT, DeepOCSORT, ByteTrack from largest to smallest.

In Tab.1 the order of FPS from largest to smallest for each tracker with different parameter sizes of the YOLO model is YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. The FPS is much higher with the combination of ByteTrack and OCSORT trackers than with the combination of the other 3 trackers.

Detailed evaluation results of YOLOv5 with different trackers under MOT20 is shown in Tab. 2. From Tab. 2 we can see that DeepOCSORT+YOLOv5x has the highest DetA. BoTSORT+YOLOv5l has the highest LocA. ByteTrack+YOLOv5l has the highest AssA. OCSORT+YOLOv5x has the highest HOTA. For DetA, the combination of each tracker with different YOLOv5 detectors is consistent with YOLOv5x, YOLOv5m, YOLOv5l, LOv5s. For LocA, the combination of each tracker with different YOLOv5 detectors is consistent with YOLOv5l, YOLOv5x, YOLOv5m, YOLOv5s. The AssA of the combined model is not regular under the YOLO detection module using the same parameter size. The HOTA metrics, on the other hand, are in the order of OCSORT, StrongSORT, BoTSORT, DeepOCSORT, ByteTrack from largest to smallest.

In Tab. 2, the largest FPS is ByteTrack+YOLOv5s. The order of FPS from largest to smallest for each tracker with different parameter sizes of the YOLO model is different. This shows that the speed of the combined model does not depend entirely on the size of the detector parameters. The FPS is much higher with the combination of ByteTrack and OCSORT trackers than with the combination of the other 3 trackers. This indicates that the speed of the combined model is strongly influenced by the integrated ReID model, while the combination without the ReID model is significantly faster.

Comparing Tab. 1 and Tab. 2, the metrics in MOT17 are higher than in MOT20, and we know from the description of the benchmark that this is because the scenarios in MOT20 are more complex. We know from the laws of DetA and LocA that the detection and localization performance of the combined model does not depend entirely on the parameter size of the YOLOv5 model. In terms of the comprehensive metric HOTA, it is difficult to determine which combination model has better accuracy. In terms of FPS, ByteTrack is the fastest with various combinations of YOLOv5 models.

4.2. YOLOv6 detector and various trackers

Detailed evaluation results of YOLOv6 with different trackers under MOT17 is shown in Tab. 3. From this table, we can observe that the highest DetA, LocA, and AssA are OCSORT+YOLOv6l, BoTSORT+YOLOv6m, and StrongSORT+YOLOv6l, respectively. As can be observed in Tab. 3, the highest HOTA is StrongSORT+YOLOv6l. In comparison, it is difficult to know which detector or which tracker shows better performance.

From Tab. 4, DeepOCSORT+YOLOv6l has the highest DetA. ByteTrack+YOLOv6m has the highest LocA and AssA. OCSORT+YOLOv6l has the highest HOTA. OCSORT+YOLOv6s has the highest FPS. With the same tracker, the LocA metrics in descending order, the performance of each YOLOv6 detector is YOLOv6m, YOLOv6s, YOLOv6l. This indicates that the integrated model using YOLOv6m as the detection module has better localization performance under the MOT20 benchmark. With the same tracker, the HOTA metrics in descending order, the performance of each YOLOv6 detector is YOLOv6l, YOLOv6m, YOLOv6s. This indicates that the integrated model using YOLOv6l as the detection module has better comprehensive performance under the MOT20.

4.3. YOLOv7 detector and various trackers

Detailed evaluation results of YOLOv7 with different trackers under MOT17 is shown in Tab. 5. From this table, we see that DetA, LocA, AssA, HOTA and FPS are the highest for OCSORT+YOLOv7x, BotSORT+YOLOv7x, StrongSORT+YOLOv7l, StrongSORT+YOLOv7x, ByteTrack+YOLOv7l, respectively. Since there are only two YOLOv7 pre-trained models available for comparison and their parameters differ by more than a factor of two, it is not meaningful to compare which detector performs better. With the same YOLOv7 pre-trained model, the HOTA metrics of the combined model in descending order, the performance of each tracker is StrongSORT, OCSORT, BoTSORT, ByteTrack, DeepOCSORT. This shows that the combined model using StrongSORT has better accuracy.

In Tab. 6, the highest DetA, LocA, AssA, HOTA and FPS are DeepOCSORT+YOLOv7l, BotSORT+YOLOv7l, ByteTrack+YOLOv7x, OCSORT+YOLOv7l, ByteTrack+YOLOv7l, respectively. With the same YOLOv7 pre-trained model, the HOTA metrics of the combined model in descending order, the performance of each tracker is OCSORT, StrongSORT, BoTSORT, DeepOCSORT, ByteTrack. This suggests that the combined model using OCSORT has better accuracy in this case. With the same YOLOv7 pre-trained model, the FPS metrics of the combined model in descending order, the speed of each tracker is ByteTrack, OCSORT, BoTSORT, StrongSORT, DeepOCSORT. This shows that in this case, the combined model using ByteTrack is more suitable for real-time applications.

4.4. YOLOv8 detector and various trackers

Detailed evaluation results of YOLOv8 with different trackers under MOT17 is shown in Tab. 7. From this table, we see that DetA, LocA, AssA, HOTA and FPS are the highest for DeepOCSORT+YOLOv8m, OCSORT+YOLOv8l, StrongSORT+YOLOv8l, OCSORT+YOLOv8m, ByteTrack+YOLOv8s, respectively. In MOT17, it is difficult to see which detector or which tracker is more accurate or faster.

In Tab. 8, we see that DetA, LocA, AssA, HOTA and FPS are the highest for DeepOCSORT+YOLOv8n, BoTSORT+YOLOv8l, ByteTrack+YOLOv8x, OCSORT+YOLOv8n, ByteTrack+YOLOv8s, respectively. The same tracker is combined with different YOLOv8 detection modules, DetA in descending order, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8x, YOLOv8l. The same tracker is combined with different YOLOv8 detection modules, and LocA is YOLOv8l, YOLOv8x, YOLOv8m, YOLOv8s, YOLOv8n, in descending order. We get a strange conclusion that the detection performance and localization performance are the best using the combined model of YOLOv8n and YOLOv8l, respectively, in the MOT20 environment. We speculate that the possible reason is that the detection performance is too low, generating too many errors such as missed detection and false detection, which can affect the accuracy performance of the whole model.

5. Analysis of results

We have to define what combination will have better result. So, we first compare all the combined models under the same benchmark.

Let us compare Tab. 1, Tab. 3, Tab. 5 and Tab. 7. We find that the highest DetA, LocA, AssA were OCSORT+YOLOv7X, BoTSORT+YOLOv8l, StrongSORT+YOLOv7, respectively. The StrongSORT+YOLOv7x has the highest HOTA value of 39.04, the six metrics such as AssA are basically higher than the average of all combinations except FPS. While ByteTrack+YOLOv8s combination model is the fastest with FPS of 62.02, but the other 7 metrics such as HOTA are lower than the average of all combinations. Overall, the LocA of all combinations are not very different. While the models with ByteTrack and OCSORT combinations are better in speed, with the combinations with YOLOv5 and YOLOv8 being faster. HOTA, IDF1, MOTA, DetA, LocA, MOTP are below the mean value of the corresponding metrics in most combinations of ByteTrack. While most of the combinations of DeepOCSORT, HOTA, AssA are below the mean value of the corresponding metrics.

Let us compare Tab. 2, Tab. 4, Tab. 6 and Tab. 8. We find that the highest DetA, LocA, AssA, HOTA and FPS are DeepOCSORT+YOLOv6l, BoTSORT+YOLOv8l, ByteTrack+YOLOv8x, OCSORT+YOLOv6l, ByteTrack+YOLOv5s respectively. Overall, the LocA of all combinations are not very different. And the model speed is better with ByteTrack combination, especially YOLOv5 and YOLOv8. HOTA, DetA are below the mean value of the corresponding metrics in most combinations of ByteTrack.

In the above analysis, we have explored the performance under MOT17 and MOT20 benchmarks respectively. With the same benchmark, it is still difficult to choose which combination of models we can use to achieve a balance between accuracy and speed. Under different benchmarks, we do not know which combination model will have good applicability. Moreover, we give all qualitative analysis, is there a quantitative analysis method that is more convenient to choose?

Inspired by the attention mechanism [25], which tells us how to choose what is important, that is the scoring method we propose in the following. From the formula in [10], we know that DetA, LocA and AssA can be represented by HOTA synthetically, and HOTA can also be decomposed into these three metrics. Therefore, it is sufficient to use HOTA to measure the comprehensive performance of accuracy. And HOTA has no relationship with FPS. Therefore, the two metrics HOTA and FPS are chosen to evaluate the comprehensive performance of the model adequately because they reflect the accuracy and speed of the model, respectively.

We consider that the model may be applied in different scenarios, so for different benchmarks, we will assign the same weights. As for the application of multi-object tracking in realistic video, a frame rate of 30 FPS is generally sufficient. But in our experimental data, some models can reach 60FPS, So, we will reduce its weight by half. We assign weights of $1-\alpha$, α to HOTA and FPS, respectively, under the same benchmark. based on the experimental data, we set α as 0.25.

$$score(X) = Xw/b, \qquad (2)$$

where X is the matrix of metric values for $c\times 2b$, w is the column vector of metric weights for $2b\times 1$, c is the number of combined models, and b is the number of benchmarks.

We first minimax normalized the metric values under each benchmark, and then used equation (2) to perform the calculation. Finally, we rank the scores from largest to smallest and find the combined model with the best overall performance under the two benchmarks. Based on the above steps, we obtain the results in Tab. 9. We can conclude that the comprehensive performance of OCSORT+YOLOv6l is the best under MOT17 and MOT20 benchmarks, with scores exceeding the second combination OCSORT+YOLOv7 by 10.61 points.

To know which type of combination (detector plus tracker) has the best overall performance, we have to calculate the average of the scores of each type. Further, averaging the scores for each YOLO version with different sizes of models is given in Tab. 10. From Tab. 10, we see that the highest average scores were obtained for each combination of models with YOLOv7 using the OCSORT tracker, exceeding the mean scores of 14.31, 5.22, and 12.19 for each combination with YOLOv5, YOLOv6, and YOLOv8, respectively.

We obtained average scores of 62.61, 50.79, 43.65, 42.56, 35.06 for each tracker OCSORT, StrongSORT, BoTSORT, ByteTrack, and DeepOCSORT, respectively. In the average score of the tracker and all YOLO CNN combined models, OCSORT performed the best, it outperformed the second place with 11.82 points. Averaging scores for each tracker are given in Tab.10. Each tracker using YOLOv7 has the best average performance, it surpassed the second place by 8.46 points.

6. Discussion

In response to the above analysis, we make the following conclusions.

First, under the MOT17 benchmark, we need to choose which of these two combinations is more suitable according to the needs of the actual application. If we pursue speed, we can choose ByteTrack+YOLOv8s, and if we pursue accuracy, we choose StrongSORT+YOLOv7x.

Second, under the MOT20 benchmark, the combination of ByteTrack+YOLOv5s is the fastest model, while the combination of OCSORT+YOLOv6l has the highest HOTA value. In this video environment, using OCSORT+YOLOv6l would be a better choice. This is because the other 7 metrics of ByteTrack+YOLOv5s are below the average of all combinations and the absolute value of HOTA is very small.

Finally, comparing the performance of each combination model with MOT17 and MOT20, most models have lower metrics under MOT20 than MOT17. Obviously, after using our proposed comprehensive evaluation methodology, we recommend OCSORT+YOLOv6l, which guarantees better accuracy and speed under MOT17 and MOT20.

Despite our analysis and discussion of the above, these analyses have certain shortcomings. For example, the parameters of the pre-trained models of the detectors we used differed relatively widely, as shown in Tab. 11, with the average parameters of YOLOv7 being nearly two times larger than the others. The reasonableness of our choice of scoring weights also needs further study and exploration. The given analysis is only applicable to the specified benchmark and the specified conditions, and its extensibility is still worth exploring.

Conclusion

In this paper, we considered a combination of various detectors and trackers for people multi-object tracking tasks. We proposed comprehensive performance evaluation method across benchmarks that can effectively evaluate the combined performance of the combined models. We conducted experiments and analysis by selecting the official pre-trained models of YOLOv5, YOLOv6, YOLOv7, YOLOv8 with representative BoTSORT, ByteTrack, DeepOCSORT, OCSORT, StrongSORT trackers under two benchmarks of MOT17 and MOT20. OCSORT+YOLOv6l model has the best comprehensive performance. The combination of OCSORT and YOLOv7 has the best average performance under two benchmarks, MOT17 and MOT20.

References

[1] Zhang Y, et al. Bytetrack: Multi-object tracking by associating every detection box. Proc 17th European Conf on Computer Vision (ECCV) 2022: 1-21.

[2] Aharon N, Orfaig R, Bobrovsky B-Z. BoT-SORT: Robust associations multi-pedestrian tracking. ArXiv Preprint. 2022. Source: <https://arxiv.org/abs/2206.14651>. DOI: 10.48550/arXiv.2206.14651.

[3] Du Y, et al. StrongSORT: Make DeepSORT great again. IEEE Trans Multimed 2023; 25: 8725-8737. DOI: 10.1109/TMM.2023.3240881.

[4] Zhou K, Xiang T. Torchreid: A library for deep learning person re-identification in pytorch. arXiv Preprint. 2019. Source: <https://arxiv.org/abs/1910.10093>. DOI: 10.48550/arXiv.1910.10093.

[5] Redmon J, Divvala SK, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 779-788. DOI: 10.1109/cvpr.2016.91.

[6] Fengxia Y, Xing Z, Boqi L. Video object tracking based on YOLOv7 and DeepSORT. arXiv Preprint. 2022. Source: <https://arxiv.org/abs/2207.12202>. DOI: 10.48550/arxiv.2207.12202.

[7] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. Proc IEEE Int Conf on Image Processing (ICIP) 2017: 3645-3649.

[8] Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv Preprint. 2022. Source: <https://arxiv.org/abs/2207.02696>. DOI: 10.48550/arXiv.2207.02696.

[9] Quan H, Bohush R, Ma G, Weichen Y, Ablameyko S. People detecting and tracking in video by CNN YOLO and StrongSORT combined algorithm. Nonlinear Phenom Complex Syst 2023; 26(1): 83-97. DOI: 10.33581/1561-4085-2023-26-1-83-97.

[10] Luiten J, et al. HOTA: A higher order metric for evaluating multi-object tracking. Int J Comput Vis 2021; 129: 548-578. DOI: 10.1007/s11263-020-01375-2.

[11] Li C, et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv Preprint. 2022. Source: <https://arxiv.org/abs/2209.02976>. DOI: 10.48550/arXiv.2209.02976.

[12] Maggiolino G, Ahmad A, Cao J, Kitani K. Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification. arXiv Preprint. 2023. Source: <https://arxiv.org/abs/2302.11813>. DOI: 10.48550/arXiv.2302.11813.

[13] Cao J, Weng X, Khirodkar R, Pang J, Kitani K. Observation-centric sort: Rethinking sort for robust multi-object tracking. arXiv Preprint. 2022. Source: <https://arxiv.org/abs/2203.14360>. DOI: 10.48550/arXiv.2203.14360.

[14] Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K. MOT16: A benchmark for multi-object tracking. arXiv Preprint. 2016. Source: <https://arxiv.org/abs/1603.00831>. DOI: 10.48550/arXiv.1603.00831.

[15] Dendorfer P, et al. MOT20: A benchmark for multi object tracking in crowded scenes. arXiv Preprint. 2020. Source: <https://arxiv.org/abs/2003.09003>. DOI: 10.48550/arXiv.2003.09003.

[16] Li C, et al. YOLOv6 v3.0: A full-scale reloading. arXiv Preprint. 2023. Source: <https://arxiv.org/abs/2301.05586>. DOI: 10.48550/arXiv.2301.05586.

[17] Sun P, et al. DanceTrack: Multi-object tracking in uniform appearance and diverse motion. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition 2022: 20993-21002. DOI: 10.1109/CVPR52688.2022.02032.

[18] Cui Y, et al. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. arXiv Preprint. 2023. Source: <https://arxiv.org/abs/2304.05170>. DOI: 10.48550/arXiv.2304.05170.

[19] Lin W, et al. Human in events: A large-scale benchmark for human-centric video analysis in complex events. arXiv Preprint. 2020. Source: <https://arxiv.org/abs/2005.04490>. DOI: 10.48550/arXiv.2005.04490.

[20] Voigtlaender P, et al. MOTS: Multi-object tracking and segmentation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition 2019: 7942-7951. DOI: 10.1109/CVPR.2019.00813.

[21] Yang L, Fan Y, Xu N. Video instance segmentation. Proc of the IEEE/CVF Int Conf on Computer Vision 2019: 5188-5197.

[22] Manohar V, et al. Performance evaluation of object detection and tracking in video. Proc 7th Asian Conf on Computer Vision 2006; Pt II: 151-161.

[23] Ristani E, Solera F, Zou R, Cucchiara R, Tomasi C. Performance measures and a data set for multi-target, multi-camera tracking. In Book: Hua G, Jégou H, eds. Computer vision – ECCV 2016 workshops. Pt II. Cham: Springer International Publishing Switzerland; 2016: 17-35.

[24] Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the clear mot metrics. Eurasip J Image Video Process 2008; 2008: 246309. DOI: 10.1155/2008/246309.

[25] Vaswani A, et al. Attention is all you need. arXiv Preprint. 2017. Source: <https://arxiv.org/abs/1706.03762>. DOI: 10.48550/arXiv.1706.03762.

Appendix A

Tab. 1. Detailed evaluation results of YOLOv5 with different trackers under MOT17

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv5s	33.33	38.21	24.57	26.85	83.35	81.17	45.47	14.21
BoTSORT	YOLOv5m	34.85	39.55	26.77	28.48	84.31	82.41	45.74	13.53
BoTSORT	YOLOv5l	35.95	40.76	26.80	28.02	84.51	82.69	48.50	13.18
BoTSORT	YOLOv5x	35.42	39.89	26.88	27.88	84.61	82.88	46.90	12.35
ByteTrack	YOLOv5s	31.08	34.12	21.22	23.77	82.68	80.61	45.71	56.04
ByteTrack	YOLOv5m	33.65	37.11	23.94	26.23	83.37	81.40	47.78	53.05
ByteTrack	YOLOv5l	33.78	37.45	23.79	25.86	83.28	81.33	48.31	46.22
ByteTrack	YOLOv5x	33.24	36.73	23.98	25.96	83.30	81.37	46.36	35.71
DeepOCSORT	YOLOv5s	32.29	36.80	25.43	27.75	82.70	80.63	41.22	8.49
DeepOCSORT	YOLOv5m	34.24	38.39	27.84	29.54	83.69	81.82	42.49	7.77
DeepOCSORT	YOLOv5l	34.08	38.37	27.87	28.96	83.77	82.08	41.97	7.44
DeepOCSORT	YOLOv5x	33.84	38.18	27.96	28.95	83.82	82.25	41.18	7.15
OCSORT	YOLOv5s	34.17	39.32	25.41	28.26	82.92	80.70	46.19	49.34
OCSORT	YOLOv5m	36.90	41.91	27.78	30.12	83.82	81.89	49.35	46.71
OCSORT	YOLOv5l	36.19	41.29	27.80	29.72	83.97	82.19	47.34	41.42
OCSORT	YOLOv5x	36.39	41.51	28.02	29.64	84.11	82.33	47.46	33.29
StrongSORT	YOLOv5s	34.77	40.16	25.19	27.93	82.85	80.64	48.20	15.24
StrongSORT	YOLOv5m	37.15	42.62	27.48	29.60	83.80	81.85	50.57	13.95
StrongSORT	YOLOv5l	37.25	42.24	27.57	29.26	83.98	82.11	50.60	12.94
StrongSORT	YOLOv5x	37.14	42.15	27.72	29.11	84.06	82.27	50.00	12.09

Tab. 2. Detailed evaluation results of YOLOv5 with different trackers under MOT20

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv5s	9.09	7.73	3.82	4.31	81.46	78.54	21.81	15.60
BoTSORT	YOLOv5m	11.01	9.37	4.81	5.24	82.72	80.35	25.37	14.49
BoTSORT	YOLOv5l	10.69	8.48	4.43	4.72	83.25	81.07	25.99	14.17
BoTSORT	YOLOv5x	11.16	9.67	5.00	5.28	83.14	81.01	25.11	12.88
ByteTrack	YOLOv5s	7.55	5.46	2.46	2.93	80.71	77.51	23.33	57.33
ByteTrack	YOLOv5m	9.40	6.67	3.15	3.56	82.08	79.67	28.14	51.40
ByteTrack	YOLOv5l	9.58	6.42	2.97	3.37	82.47	80.16	30.97	48.22
ByteTrack	YOLOv5x	9.77	7.03	3.29	3.66	82.42	80.13	29.14	33.29
DeepOCSORT	YOLOv5s	9.05	8.04	4.05	4.64	81.15	78.24	20.38	9.31
DeepOCSORT	YOLOv5m	10.80	9.40	5.04	5.57	82.40	80.09	23.28	7.80
DeepOCSORT	YOLOv5l	10.43	8.62	4.63	5.01	82.94	80.80	23.63	7.99
DeepOCSORT	YOLOv5x	10.98	9.77	5.23	5.60	82.88	80.77	23.21	7.07
OCSORT	YOLOv5s	9.63	8.36	3.98	4.62	81.26	78.28	23.39	46.71
OCSORT	YOLOv5m	11.61	10.00	4.98	5.57	82.53	80.14	27.17	40.75
OCSORT	YOLOv5l	11.36	9.20	4.58	5.02	83.06	80.86	28.29	40.90
OCSORT	YOLOv5x	11.93	10.50	5.17	5.62	82.99	80.82	27.69	30.55
StrongSORT	YOLOv5s	9.42	8.12	3.99	4.59	81.04	77.95	22.37	14.93
StrongSORT	YOLOv5m	11.47	9.88	4.99	5.53	82.31	79.77	26.54	13.33
StrongSORT	YOLOv5l	11.33	9.13	4.59	4.99	82.86	80.51	28.10	13.35
StrongSORT	YOLOv5x	11.64	10.13	5.19	5.60	82.75	80.46	26.27	11.99

Tab. 3. Detailed evaluation results of YOLOv6 with different trackers under MOT17

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv6s	33.19	36.59	23.76	25.63	84.79	82.90	46.60	13.07
BoTSORT	YOLOv6m	35.89	40.29	26.94	28.45	85.07	83.42	48.04	9.00
BoTSORT	YOLOv6l	36.60	41.45	28.33	29.64	84.70	82.94	47.58	9.16
ByteTrack	YOLOv6s	29.98	30.84	19.08	21.17	83.61	81.66	47.41	37.98
ByteTrack	YOLOv6m	33.25	36.29	22.60	24.59	83.73	81.95	49.22	25.85
ByteTrack	YOLOv6l	33.54	36.92	24.14	26.01	83.32	81.47	46.96	33.93
DeepOCSORT	YOLOv6s	31.81	34.91	24.57	26.38	84.16	82.41	41.45	8.26
DeepOCSORT	YOLOv6m	33.24	36.76	27.90	29.50	84.42	82.89	39.83	6.65
DeepOCSORT	YOLOv6l	34.79	39.26	29.43	30.65	84.02	82.34	41.40	6.37
OCSORT	YOLOv6s	34.26	37.95	24.55	26.90	84.40	82.49	48.05	42.30
OCSORT	YOLOv6m	37.01	42.05	27.90	30.12	84.66	82.97	49.30	33.52
OCSORT	YOLOv6l	37.92	43.31	29.47	31.37	84.22	82.44	49.04	35.90
StrongSORT	YOLOv6s	34.71	38.52	24.38	26.63	84.33	82.41	49.64	14.09
StrongSORT	YOLOv6m	37.24	41.79	27.66	29.77	84.51	82.84	50.31	12.15
StrongSORT	YOLOv6l	38.36	43.48	29.10	30.84	84.15	82.35	50.81	11.48

Tab. 4. Detailed evaluation results of YOLOv6 with different trackers under MOT20

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv6s	9.99	7.84	3.86	4.35	82.46	79.75	26.05	14.41
BoTSORT	YOLOv6m	11.89	11.14	5.83	6.26	82.86	80.57	24.45	9.56
BoTSORT	YOLOv6l	14.83	16.91	9.28	10.08	81.91	79.40	23.92	7.89
ByteTrack	YOLOv6s	7.46	4.47	2.06	2.41	82.77	80.19	27.10	42.20
ByteTrack	YOLOv6m	8.94	5.65	2.73	3.07	83.24	81.06	29.39	33.56
ByteTrack	YOLOv6l	11.26	9.79	4.59	5.10	81.78	79.34	27.80	33.91
DeepOCSORT	YOLOv6s	9.85	7.88	4.05	4.63	82.17	79.49	24.11	9.13
DeepOCSORT	YOLOv6m	11.52	11.04	6.10	6.63	82.60	80.32	21.94	5.89
DeepOCSORT	YOLOv6l	14.39	16.79	9.65	10.59	81.69	79.24	21.64	4.72
OCSORT	YOLOv6s	10.61	8.45	4.01	4.63	82.29	79.53	28.24	44.11
OCSORT	YOLOv6m	12.45	11.91	6.03	6.64	82.70	80.37	25.90	34.74
OCSORT	YOLOv6l	15.44	17.96	9.58	10.66	81.80	79.26	25.07	33.43
StrongSORT	YOLOv6s	10.51	8.31	4.01	4.60	82.07	79.19	27.69	15.44
StrongSORT	YOLOv6m	12.20	11.47	6.04	6.58	82.49	80.05	24.87	11.72
StrongSORT	YOLOv6l	15.15	17.22	9.61	10.59	81.56	78.91	24.11	8.67

Tab. 5. Detailed evaluation results of YOLOv7 with different trackers under MOT17

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv7l	36.38	41.91	28.77	30.17	84.49	82.60	46.29	12.54
BoTSORT	YOLOv7x	36.90	42.86	29.76	31.41	84.50	82.57	46.04	11.75
ByteTrack	YOLOv7l	35.40	40.39	26.02	28.48	83.14	81.13	48.50	42.70
ByteTrack	YOLOv7x	35.70	40.95	26.99	29.67	83.23	81.11	47.56	31.67
DeepOCSORT	YOLOv7l	34.96	40.11	30.01	31.39	83.72	81.95	41.05	7.30
DeepOCSORT	YOLOv7x	34.73	39.70	30.98	32.52	83.73	81.95	39.24	7.15
OCSORT	YOLOv7l	38.05	44.36	30.02	32.21	83.98	82.06	48.51	42.10
OCSORT	YOLOv7x	38.63	45.53	30.99	33.42	83.97	82.05	48.43	31.82
StrongSORT	YOLOv7l	38.85	45.03	29.68	31.68	83.92	81.95	51.12	11.99
StrongSORT	YOLOv7x	39.04	45.65	30.65	32.98	83.89	81.92	49.99	11.57

Tab. 6. Detailed evaluation results of YOLOv7 with different trackers under MOT20

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv7l	11.80	11.04	5.73	6.25	83.05	80.73	24.50	13.01
BoTSORT	YOLOv7x	11.57	10.31	5.28	5.64	83.03	80.76	25.56	12.22
ByteTrack	YOLOv7l	10.31	8.48	3.91	4.53	82.23	79.65	27.36	42.38
ByteTrack	YOLOv7x	10.03	7.47	3.46	3.88	82.33	79.90	29.21	30.71
DeepOCSORT	YOLOv7l	11.74	11.39	6.04	6.68	82.81	80.48	22.99	7.03
DeepOCSORT	YOLOv7x	11.15	10.23	5.53	5.96	82.75	80.54	22.65	7.30
OCSORT	YOLOv7l	12.36	11.94	5.97	6.68	82.89	80.54	25.77	40.30
OCSORT	YOLOv7x	12.04	10.97	5.47	5.99	82.89	80.59	26.66	30.18
StrongSORT	YOLOv7l	12.24	11.53	5.97	6.64	82.68	80.19	25.27	11.53
StrongSORT	YOLOv7x	11.95	10.78	5.49	5.96	82.65	80.22	26.21	11.78

Tab. 7. Detailed evaluation results of YOLOv8 with different trackers under MOT17

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv8n	32.95	36.85	22.66	24.90	84.56	82.45	48.12	14.25
BoTSORT	YOLOv8s	34.53	39.37	25.88	27.88	84.64	82.69	46.39	13.83
BoTSORT	YOLOv8m	35.41	39.78	26.99	28.60	85.15	83.37	46.68	13.28
BoTSORT	YOLOv8l	33.72	36.90	24.80	26.05	85.18	83.38	46.05	12.93
BoTSORT	YOLOv8x	35.18	39.28	26.46	27.91	85.09	83.27	47.00	12.05
ByteTrack	YOLOv8n	29.14	30.87	19.16	21.51	83.84	81.78	44.54	59.28
ByteTrack	YOLOv8s	31.22	33.83	21.45	23.52	83.70	81.76	45.84	62.02
ByteTrack	YOLOv8m	33.31	36.56	23.53	25.62	83.82	81.98	47.44	47.78
ByteTrack	YOLOv8l	32.55	34.74	22.03	23.95	83.69	81.71	48.32	43.28
ByteTrack	YOLOv8x	33.66	37.31	23.66	25.94	83.61	81.67	48.13	32.21
DeepOCSORT	YOLOv8n	30.99	34.49	23.37	25.61	83.99	82.00	41.34	9.02
DeepOCSORT	YOLOv8s	32.84	37.09	26.73	28.79	84.01	82.19	40.67	8.21
DeepOCSORT	YOLOv8m	33.70	37.81	28.08	29.61	84.52	82.80	40.74	7.61
DeepOCSORT	YOLOv8l	31.87	34.31	25.90	27.12	84.50	82.79	39.46	7.70
DeepOCSORT	YOLOv8x	33.69	37.47	27.53	29.04	84.35	82.67	41.45	7.24
OCSORT	YOLOv8n	33.52	37.57	23.33	26.03	84.20	82.09	48.36	50.20
OCSORT	YOLOv8s	35.77	41.32	26.80	29.36	84.24	82.28	48.01	50.02
OCSORT	YOLOv8m	37.41	42.69	27.99	30.19	84.69	82.94	50.25	41.77
OCSORT	YOLOv8l	35.45	39.57	25.84	27.74	84.69	82.85	48.83	38.99
OCSORT	YOLOv8x	36.90	42.21	27.56	29.67	84.61	82.78	49.60	30.39
StrongSORT	YOLOv8n	33.69	37.49	23.22	25.84	84.13	82.01	49.09	16.15
StrongSORT	YOLOv8s	35.87	40.90	26.62	29.08	84.14	82.15	48.62	14.59
StrongSORT	YOLOv8m	37.11	41.96	27.69	29.72	84.60	82.80	49.97	14.76
StrongSORT	YOLOv8l	36.01	39.99	25.55	27.25	84.63	82.81	50.92	14.56
StrongSORT	YOLOv8x	36.93	41.59	27.21	29.12	84.58	82.73	50.33	11.97

Tab. 8. Detailed evaluation results of YOLOv8 with different trackers under MOT20

Tracker	Detector	HOTA	IDF1	DetA	MOTA	LocA	MOTP	AssA	FPS
BoTSORT	YOLOv8n	10.61	9.94	5.03	5.60	81.34	78.35	22.61	14.68
BoTSORT	YOLOv8s	10.52	8.86	4.46	4.97	82.29	79.64	25.03	15.09
BoTSORT	YOLOv8m	10.58	7.81	4.00	4.35	83.51	81.31	28.16	14.46
BoTSORT	YOLOv8l	9.86	6.59	3.34	3.55	84.52	82.62	29.26	14.39
BoTSORT	YOLOv8x	10.33	7.09	3.69	3.88	84.00	82.04	29.07	12.81
ByteTrack	YOLOv8n	8.60	6.57	3.01	3.56	81.27	78.31	24.70	52.90
ByteTrack	YOLOv8s	8.64	6.05	2.75	3.24	81.75	78.93	27.29	55.17
ByteTrack	YOLOv8m	9.19	5.63	2.67	3.09	82.95	80.66	31.73	49.34
ByteTrack	YOLOv8l	8.26	4.72	2.24	2.55	83.69	81.58	30.50	42.28
ByteTrack	YOLOv8x	9.33	5.62	2.65	2.99	83.18	81.07	32.88	30.80
DeepOCSORT	YOLOv8n	10.76	10.37	5.33	6.04	81.08	78.12	21.92	7.94
DeepOCSORT	YOLOv8s	10.39	9.11	4.71	5.33	82.05	79.42	23.03	8.36
DeepOCSORT	YOLOv8m	10.22	7.66	4.18	4.62	83.23	81.05	25.10	8.46
DeepOCSORT	YOLOv8l	9.49	6.46	3.48	3.74	84.24	82.39	26.04	9.24
DeepOCSORT	YOLOv8x	10.00	6.99	3.86	4.11	83.60	81.73	26.03	8.24
OCSORT	YOLOv8n	11.16	10.74	5.25	6.01	81.15	78.15	23.88	38.53
OCSORT	YOLOv8s	10.99	9.51	4.65	5.32	82.13	79.46	26.13	41.58
OCSORT	YOLOv8m	11.03	8.31	4.14	4.63	83.34	81.10	29.56	39.53
OCSORT	YOLOv8l	10.28	7.02	3.44	3.75	84.39	82.43	30.81	37.44
OCSORT	YOLOv8x	10.95	7.67	3.82	4.14	83.81	81.81	31.52	28.78
StrongSORT	YOLOv8n	11.06	10.36	5.26	5.96	80.95	77.78	23.49	12.85
StrongSORT	YOLOv8s	10.94	9.40	4.64	5.29	81.89	79.06	25.97	13.91
StrongSORT	YOLOv8m	10.96	8.29	4.15	4.60	83.10	80.76	29.16	17.01
StrongSORT	YOLOv8l	10.19	7.01	3.45	3.74	84.14	82.09	30.20	16.34
StrongSORT	YOLOv8x	10.85	7.61	3.81	4.09	83.61	81.50	31.06	13.28

Tab. 9. TOP Score ranking of the combined models

Tracker	Detector	Score	Rank	Tracker	Detector	Score	Rank
OCSORT	YOLOv6l	84.99	1	OCSORT	YOLOv6m	68.32	6
OCSORT	YOLOv7l	74.38	2	OCSORT	YOLOv8m	66.08	7
StrongSORT	YOLOv6l	73.66	3	BoTSORT	YOLOv6l	66.04	8
OCSORT	YOLOv7x	70.00	4	OCSORT	YOLOv5l	64.05	9
OCSORT	YOLOv5m	68.54	5	OCSORT	YOLOv8s	63.16	10

Tab. 10. Average performance score ranking of different trackers with the same YOLO version

Tracker	Detector	Score	Rank	Tracker	Detector	Score	Rank
OCSORT	YOLOv7	72.19	1	StrongSORT	YOLOv5	45.57	11
OCSORT	YOLOv6	66.97	2	DeepOCSORT	YOLOv7	44.58	12
StrongSORT	YOLOv7	62.33	3	ByteTrack	YOLOv8	42.28	13
OCSORT	YOLOv8	60.00	4	ByteTrack	YOLOv5	42.18	14
OCSORT	YOLOv5	57.88	5	BoTSORT	YOLOv8	41.27	15
ByteTrack	YOLOv7	54.66	6	BoTSORT	YOLOv5	39.10	16
StrongSORT	YOLOv6	54.55	7	DeepOCSORT	YOLOv6	38.90	17
BoTSORT	YOLOv7	53.63	8	ByteTrack	YOLOv6	37.35	18
StrongSORT	YOLOv8	48.37	9	DeepOCSORT	YOLOv8	32.35	19
BoTSORT	YOLOv6	47.34	10	DeepOCSORT	YOLOv5	30.89	20

Tab. 11. Comprehensive performance scores for each detector

Detector	Average parameters/M	Score
YOLOv7	68	57.48
YOLOv6	29.425	49.02
YOLOv8	30.44	44.85
YOLOv5	32.7	43.12

Authors' information

Hongxu Quan (b. 1988), graduated from Belarusian State University with a Master's degree (2023) in Mathematics and Computer Science. Currently, he works as an associate professor in Qiandongnan Nationalities Polytechnic. Research interests: computer vision, object detection, object tracking, BIM technology, intelligent construction.

Guangdi Ma (b.1985). Graduated from Chinese Academy of Surveying and Mapping in 2011. Chief Engineer of EarthView Image Inc. His scientific interests are: image analysis, photogrammetry, point cloud and oblique photography aided real 3D reconstruction.

Yang Weichen (b. 1979). Graduated from Jilin University, China in 2001. General manager of EarthView Image Inc. His scientific interests are: image analysis, photogrammetry, geographical information systems. Pioneered the business service mode of remote sensing target recognition to assist refined social governance in China.

Rykhard Bohush (b. 1974). Graduated from Polotsk State University in 1997. In 2022, he received his Doctor of Sciences degree. Head of Computer Systems and Networks department of Polotsk State University. His scientific interests include image and video processing, intelligent systems, and machine learning.

Fang Zuo (b.1981). Received B.S. and M.S. degrees in Computer Science and Applied Mathematics from Henan University, Ph.D. degree in Computer Science from East China Normal University. An associate professor at Henan University. His research interests are in pattern recognition, distributed multimedia systems, intelligence algorithm, and mathematical optimization theory.

Sergey Ablameyko (b. 1956). Received DipMath in 1978, PhD in 1984, DSc in 1990, Prof in 1992. Professor of the Belarusian State University. His scientific interests are artificial intelligence, computer vision, knowledge based systems, geographical information systems, medical imaging.

Received September 12, 2023. The final version – January 15, 2024.