MOT16-VisualTracker

Linghang Kong, Xingyin Xu, Hanyuan Zhang

***Abstract*—Multi-object tracking (MOT) is critical in applications such as autonomous driving, robotics, and surveillance. Despite advancements, MOT systems face persistent challenges, including occlusions and rapid movements, which compromise the accuracy of object identification and tracking. Addressing the need for improved evaluative tools, we introduce an interactive data visualization platform (MOT16-VisualTracker). This tool aids developers in understanding and enhancing the performance of their MOT models by providing real-time visual comparisons with ground truth data. Our platform, built using Streamlit, facilitates an insightful examination of a model's deficiencies, enabling developers to make informed improvements. By offering visual and quantitative feedback on model performance, the tool fosters a deeper understanding of typical failure points in tracking applications. For more details and access to our codebase, visit our GitHub repository (https://github.com/hanyuanz2000/MOT16-VisualTracker).**

***Index Terms*—MOT, Multiple Object Tracking, Object Tracking, Computer Vision, Data Visualization.**

I. Introduction

The evolution of multi-object tracking (MOT) systems has been pivotal in advancing numerous applications, from autonomous vehicles to surveillance. Despite significant progress, accurately tracking multiple objects in dynamic environments remains a challenging problem, especially under varying conditions like occlusions [1] [2], id switch [3], and environmental factors such as lighting and shadows. These difficulties underscore the ongoing need for refined approaches in MOT systems to better understand and overcome their limitations.

Challenges in visualizing model performance: To the best of our knowledge, there are few if any interactive visualization tools available for the MOT16 dataset [4]. Researchers typically rely on two main methods: evaluating each frame by human eyes or relying on quantitative evaluation metrics. These practices, often implemented separately, involve either laboriously checking each frame visually or relying solely on numerical metrics. This bifurcated approach is inefficient and time-consuming, lacking a unified method that combines both qualitative and quantitative assessments.

Our Approach: Our research introduces an innovative interactive tool designed to enhance understanding of model performance. By leveraging interactive visualization techniques, our tool contrasts real-time data with ground truth to pinpoint performance discrepancies and weaknesses. This dynamic visualization facilitates a dual analysis approach: User will be able to use bar charts of quantitative metrics to validate visual observations, and comparing abnormal changes in these metrics with direct detailed, frame-by-frame manual inspections. Our MOT16-VisualTracker allows users not only to identify performance anomalies within specific time frames but also to focus their attention strategically on critical segments where deviations from expected model behavior occur.

This paper was produced by the New York University Data Science students.

Manuscript received May 9, 2024.

Main contributions: Our primary contributions are encapsulated in the development and implementation of the MOT16-VisualTracker, an interactive tool that enhances the comprehension of model performance on the MOT16 dataset through detailed visual and quantitative analysis:

- Interactive Visualization Features: The MOT16-VisualTracker presents bounding boxes on both the initial and final frames selected of a video sequence, alongside five dynamic evaluation metrics. Users can manipulate time frame sliders to observe changes in bounding boxes and metric outcomes, enabling a direct comparison between model outputs and ground truth. This interaction not only clarifies model performance over time but also highlights specific frames where discrepancies are most pronounced.

- We have extended the functionality of the TrackEval [5] backend to support evaluation on user-specified frames. This modification enables the MOT16-VisualTracker to perform targeted analyses on selected frames, rather than processing entire video sequences. This capability is crucial for detailed examinations of specific incidents or anomalies within a sequence, such as occlusions, rapid movements, or environmental effects like lighting and shadows.

- Real-World Application and Model Refinement: Through practical case studies, we demonstrate how the MOT16-VisualTracker identifies weaknesses within MOT models. These insights are pivotal for iterative model improvement, ensuring that enhancements are based on both observed anomalies and quantitative data.

Paper Organization:

- **Section II: Related Works** - Discusses the MOT16 dataset, reviews MOT models, and explores the role of visualization in evaluating MOT performance.
- **Section III: Background** - Provides background and contextual information about the MOT16 dataset and the development of the MOT16 Visual Tracker, setting the stage for a detailed discussion of our tool.
- **Section IV: Methods** - Details the architecture and

functionality of the MOT16 Visual Tracker, focusing on its data handling and visualization capabilities.
- **Section V: Evaluation** - Conducts case studies to demonstrate the benefits of our tool over traditional evaluation methods.
- **Section VI: Conclusion and Future Works** - Summarizes findings and outlines future enhancements for MOT evaluation tools.

II. Related Works

A. MOT16 Dataset

The MOT16 dataset [4] serves as the foundation for our project due to its comprehensive annotations and suitability for evaluating multiple object tracking (MOT) algorithms. Building upon the earlier MOT15 framework, MOT16 introduces enhancements including more reliable ground truth data, a more balanced distribution of crowd densities, and a better-suited training set. The dataset comprises 14 sequences, each split into training and testing clips, and includes object detections in frames. Its inclusion of real-world challenges like occlusions and variable crowd densities makes it an ideal platform for testing the robustness of MOT algorithms in complex urban environments. Our project utilizes the MOT16 dataset to develop visualization tools that not only display tracking performance discrepancies effectively but also provide intuitive insights into temporal sequences, spatial relationships, and performance metrics. These enhancements are critical for understanding and improving the robustness and accuracy of MOT systems.

B. MOT Models

Prior studies in the field of data visualization and machine learning have explored various aspects of MOT performance, with a focus on enhancing tracking accuracy and robustness. Researchers have developed sophisticated models that integrate deep learning techniques to improve detection and tracking under partial occlusions or in crowded scenes. Zhou *et al.* introduced the concept of omni-scale feature learning and implemented it in OSNet [6]. Aharon *et al.* added improvements such as camera motion compensation-based features tracker and cosine-distance fusion methods for robustness to the traditional SORT model and created BoT-SORT [7]. BoT-SORT combines the benefits of object tracking and re-identification technologies, emphasizing robust tracking across various scenes and conditions and prioritizing high accuracy in identity preservation. Zhang *et al.* introduced ByteTrack, a simple, effective and generic association method, tracking by associating almost every detection box instead of only the high-score ones [8]. ByteTrack emphasizes on simplicity and speed, with a focus on efficient tracking by leveraging every detection box and minimizing identity switches. Wojke *et al.* developed DeepSORT by improving from the baseline SORT model and integrating appearance information [9]. Recently, Du *et al.* revisited DeepSORT and developed StrongSORT, which mainly served as a baseline for comparison between tracking methods [10]. StrongSORT focuses on balancing speed and accuracy through efficient algorithms like AFLink and GSI, addressing missing associations and detections.

C. Data Visualization in MOT Models

There remains a gap in comprehensively understanding how these models fail, particularly through visual analysis. Alikhanov *et al.* compared the state-of-the-art MoTs' performance on the surveillance dataset, utilizing barplot, dot-plot, qualitative comparisons, and line-plots [11]. However, they focused more on the influence of confidence levels on different MoTs. To our best knowledge, existing works primarily focus on quantitative improvements over benchmarks without deep dives into qualitative analysis of failure modes. Our research seeks to fill this gap by leveraging data visualization to provide a nuanced understanding of MOT model failures, offering a bridge between raw performance metrics and actionable insights.

D. Official Evaluation Kit and Metrics in MOT Models

The standard evaluation metrics used in the MOTChallenge Official Evaluation Kit [5] encompass five categories, including HOTA (Higher Order Tracking Accuracy), CLEAR MOT Metrics, Identity Metrics, VACE Metrics, and COUNT Metrics. Each category targets specific aspects of tracking performance:

- **HOTA Metrics:** The primary metric is HOTA (Higher Order Tracking Accuracy), which assesses a tracker's ability to accurately detect and associate objects across frames. Defined in [12], HOTA measures "how well the trajectories of matching detections align, averaging this over all matching detections, while also penalizing detections that do not match." This metric effectively balances detection precision with association accuracy, providing a comprehensive overview of tracking performance.

  HOTA is calculated across various detection thresholds, making it sensitive to both detection quality and association accuracy. The overall detection accuracy, $DetA$, is given by:

$$DetA = \frac{|TP|}{|TP| + |FN| + |FP|}$$

  where TP represents true positives, FN false negatives, and FP false positives. The association accuracy, $AssA$, is calculated as:

$$AssA = \frac{1}{|TP|} \sum_{c \in TP} \frac{|TPA(c)|}{|TPA(c)| + |FNA(c)| + |FPA(c)|}$$

  Here, $TPA(c)$ denotes true positive associations, $FNA(c)$ false negative associations, and $FPA(c)$ false positive associations. Combining these, the $HOTA$ score at each threshold α is:

$$HOTA_\alpha = \sqrt{DetA_\alpha \cdot AssA_\alpha} \tag{1}$$

  The overall HOTA score is then calculated as the integral over all thresholds:

$$HOTA = \int_{0<\alpha\leq 1} HOTA_\alpha \, d\alpha \tag{2}$$

- **CLEAR MOT Metrics:** These metrics quantify the accuracy and precision of object detection and tracking. These metrics, such as MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision), evaluate the overall effectiveness of the tracking system by considering factors like false positives, missed detections, and identity switches [13].

- **Identity Metrics:** These metrics assess the ability of a tracking system to maintain consistent object identities across different frames. A key metric in this category is IDF1, which evaluates the accuracy of identity labels assigned by the tracker over sequences, reflecting its ability to accurately preserve identities [14].

- **VACE Metrics:** Metrics such as Segment-level False Discovery and Assignment (SFDA) and Average Tracking Accuracy (ATA), are used to evaluate the consistency and accuracy of tracking assignments on a segment-by-segment basis [15].

- **COUNT Metrics:** These metrics provide numerical summaries of the tracking process, counting detections, ground-truth detections, tracker-assigned identities, and true object identities, helping in assessing the scale and complexity the tracker can manage [4].

Together, these metrics provide a multifaceted evaluation of a tracker's performance, from basic detection to complex aspects like identity maintenance and tracking accuracy over time.

III. Background

The current methods for evaluating MOT systems quantitatively, as facilitated by the MOTChallenge Official Evaluation Kit [5], require running metrics across entire sequences and subsequently manually reviewing output to identify model failures. This process is often cumbersome, time-intensive, and not conducive to iterative adjustments during the model development phase due to its lack of immediate, actionable insights. Moreover, the existing evaluation practices do not support the detailed, frame-by-frame analysis necessary for pinpointing specific issues such as occlusions, rapid movements, and variations due to environmental factors like lighting and shadows.

Recognizing these limitations, our project introduces a novel visualization tool, the MOT16-VisualTracker, which significantly enhances the process of evaluating and refining MOT algorithms by providing a user-friendly, interactive interface. This tool allows users to:

- Conduct both broad and segmented analyses of tracking performance, enabling a nuanced understanding of model behavior across different conditions within the MOT16 dataset.
- Visualize discrepancies in real-time between model predictions and ground truth, facilitating a more intuitive and efficient refinement process.

By integrating qualitative and quantitative evaluation methods, our tool offers a comprehensive overview of model performance. It features dynamic visualization capabilities that include displaying bounding boxes and evaluation metrics across selected video frames.

This approach not only highlights where discrepancies occur but also provides insights into the nature and severity of these deviations. Our tool aims to bridge the gap between traditional evaluation methods and the need for more detailed, immediate feedback, thereby accelerating the iterative process of model improvement and helping researchers and developers achieve higher accuracy and reliability in their MOT systems.

IV. Methods

A. System Architecture

The architecture of our visualization tool comprises two main components: the backend processing module and the frontend visualization interface. The backend is built on Python and utilizes the MOTChallenge Official Evaluation Kit, which we have modified to support incremental evaluation over user-specified frame ranges. The frontend is developed using Streamlit, a Python library that enables rapid development of interactive web applications. Our visualization aims to answer several key questions: How does your model perform and compare with ground truth? What insights can we gather about the model's strengths and weaknesses?

B. Data Handling

Our tool processes input in the form of .txt files containing bounding box data as specified by the MOTChallenge format. Each line in the file represents a detected object in a frame and includes the frame number ($frame_{id}$), object ID, bounding box coordinates, and confidence scores, etc. Coordinates are represented in a format of $(x_{min}, y_{min}, x_{max}, y_{max})$, where (x_{min}, y_{min}) represents the coordinates of the top-left corner of the bounding box, and (x_{max}, y_{max}) represents the coordinates of the bottom-right corner of the bounding box. The backend parses this data to filter frames according to user input and prepares it for both visualization and performance evaluation.

C. Bounding Box Visualization

In the bounding box visualization module, users have the capability to upload the output file from their model and select specific initial and end frames using an interactive slider. This functionality allows users to dynamically choose any frame or range of frames they wish to analyze. As the user adjusts the slider, the tool instantaneously renders the bounding boxes on the corresponding video frames. This real-time update facilitates an intuitive understanding of how the model's predictions evolve over time, compared to the ground truth annotations. The module supports seamless interactive navigation through video sequences, enhancing user engagement and analytical precision.

D. Zooming Feature

Alongside the basic visualization of bounding boxes on individual frames, the tool also offers a zooming feature that provides users with a clear, expanded view of selected frames. This capability is particularly beneficial when dealing with small bounding boxes or scenarios where multiple bounding boxes overlap, making it challenging to discern details with the naked eye. The zoom functionality is implemented using Viewer.js, a powerful JavaScript library. In our application, images are converted to base64 and embedded within a customizable HTML container, which then utilizes Viewer.js to enable dynamic, interactive image zooming and manipulation directly in the user interface. This feature enhances detailed inspection and analysis of tracking accuracy under complex visual conditions.

E. Evaluation Metrics Visualization

The evaluation metrics visualization module dynamically calculates and presents a variety of metrics for selected frames upon user initiation over their chosen video segment. The results are displayed in the form of bar charts and detailed logs, aiding in identifying performance fluctuations and understanding the behavior of the tracking algorithm across different scenarios within the same video sequence. We support the visualization of five types of metrics: HOTA, CLEAR, Identity, VACE, and Count Chart.

- **HOTA Metrics Graph** It contains metrics including Higher Order Tracking Accuracy (HOTA), Detection Accuracy (DetA), Detection Recall (DetRe), Detection Precision (DetPr), Association Accuracy (AssA), Association Recall (AssRe), Association Precision (AssPr), Localization Accuracy (LOCA), Overall Weighted Tracking Accuracy (OWTA), Higher Order Tracking Accuracy with Localization (HOTA(LOCA)), and Localization Accuracy (LOCA(O)).

- **CLEAR MOT Metrics Graph** It contains metrics including Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), Multi-Object Detection Accuracy (MODA), Detection Recall (DetRe), Detection Precision (DetPr), Association Accuracy (AssA), Association Recall (AssRe), Association Precision (AssPr), Localization Accuracy (LOCA), Overall Weighted Tracking Accuracy (OWTA), Higher Order Tracking Accuracy with Localization (HOTA(LOCA)), Localization Accuracy (Overlap) (LOCA(O)), Mostly Tracked (MT), Mostly Lost (ML), Partially Tracked (PT), False Positives (FP), False Negatives (FN), Identity Switches (IDSW), Fragmentation (Frag), and Standard Multi-Object Tracking Accuracy (sMOTA).

- **Identity Metrics Graph** It contains metrics including Identity F1 Score (IDF1), Identity Recall (IDR), Identity Precision (IDP), Identity True Positives (IDTP), Identity False Negatives (IDFN), Identity False Positives (IDFP).

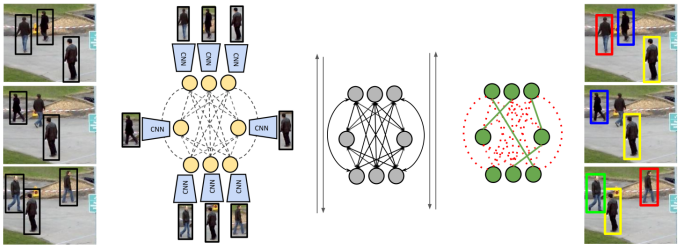


Fig. 1: MPNTrack Method Overview

- **VACE Metrics Graph** It contains metrics including Semi-Final Data Alignment (SFDA), Advanced Tracking Accuracy (ATA).

- **COUNT Metrics Graph** It contains metrics including Detected Objects (Des), Ground Truth Detections (GT_Dets), Ground Truth Identities (GT_IDs).

To accommodate evaluations that begin in the middle of video sequences, we have implemented a method in the backend code that involves creating a temporary folder to store the selected frames from the tracker and ground truth data. The object class from the original eval toolkit [5] has been modified to handle evaluations starting from a user-defined frame.

V. Evaluation

In evaluating the effectiveness of the MOT16-VisualTracker, we conducted a comparative analysis against traditional evaluation metrics. Traditional methods, which typically rely on isolated quantitative metrics or manual inspection of model outputs, often fall short in providing a comprehensive assessment of a model's overall performance. By integrating visual analytics, our MOT16-VisualTracker offers a more robust and holistic evaluation framework, enabling detailed scrutiny of tracking accuracy and identification of specific performance issues.

A. Case Studies

In our case studies, we focus on the MPNTrack model [16], a notable departure from traditional multi-object tracking (MOT) models that primarily focus on feature extraction. MPNTrack utilizes a fully differentiable Message Passing Network (MPN) framework that prioritizes data association, as illustrated in Figure 1. Although recent state-of-the-art (SOTA) models like Bytetrack and SORT-based models have outperformed MPNTrack, its robust performance and potential for further improvement make it an excellent candidate for demonstrating the capabilities of our visualization tool. We will showcase three case studies using the sequence MPN16-02.

The Baseline Traditionally, analyzing a customized MOT model requires counting and comparing the number of detections and IDs visually in densely populated scenes. As

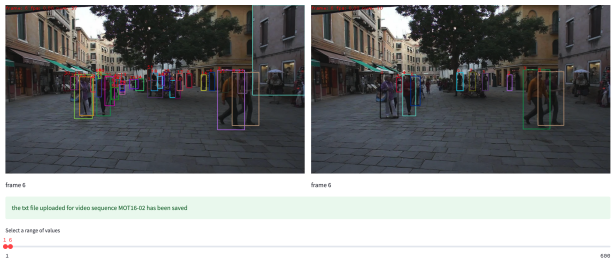


Fig. 2: Basecase

we can see from figure 2, bounding boxes may overlap, some boxes are too small, and the colors of the bounding boxes that belong to the same objects may be different for the model and ground truth. Unless there is significantly less bounding boxes labeled or abnormally located bounding boxes, it would be almost impossible for naked eyes to conduct any meaningful analysis or inferences.

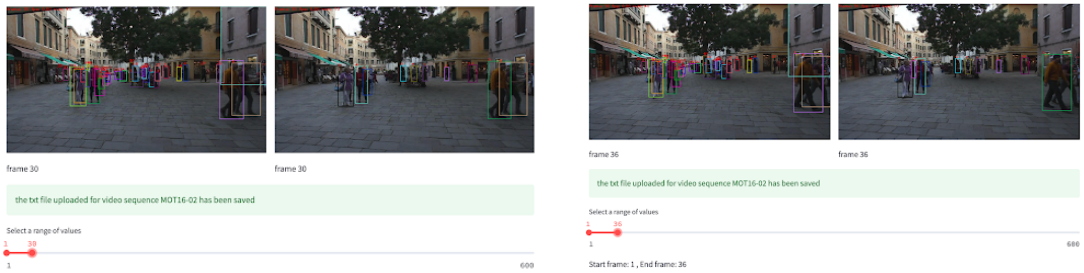
Case Study 1: the left column of Figure 3 shows ground truth, and right column of Figure 3 represents MPNTrack outcome. Comparing the MPNTrack outcome between frames 30 and 36, we find that MPNTrack made an error when tracking people on the right bottom of the frame. In frame 30, MPNTrack correctly identified pedestrian id 3 and 8, while in frame 36, it fails to track pedestrian id=8. This is a typical ”lost tracking” phenomenon in MOT models. Our tool not only shows the bounding boxes of the frames dynamically but also calculated and visualized the corresponding metrics for reference. In this case study, we calculated the metrics between frame 1 to frame 30, and frame 1 to frame 36. As shown in Figure 4, the HOTA scores of frame 30 are all higher than the HOTA scores of frame 36 except Association Recall (AssRe), with a significant 3.3% decrease in Overall Weighted Tracking Accuracy (OWTA). The decrease in OWTA indicates the inconsistent model MPNTrack performance. In Figure 5, the Mostly Lost (ML) increases by one, which corresponds to the losing track of id 3. The rest of the metrics in Figure 5 remain approximately the same. By Figure 6, the metrics overall indicate that frame 30 has a slightly better performance compared to frame 36, with a 20.45% increase in Identity False Negatives (IDFN) while 20.2% increase in Identity True Positives (IDTP). Figure 7 shows a significant 5.74% decrease in Advanced Tracking Accuracy (ATA), indicating a notable decline in tracking accuracy. This can be caused by ”lost tracking”. Overall by evaluating the metrics, we can find the insight that frame 36 has a worse performance than frame 30, and some metrics align and indicate the existence of lost tracking. This case successfully revealed a drawback of the MPNTrack algorithm. As mentioned above, MPNTrack uses a message passing network that relies heavily on node and edge encoding. For our case 1, as an object is exiting toward the edge of the image, the active edges in the graph representation usually decrease, and existing edges are more likely to be classified as non-active, thus losing track of the object in motion.

Case Study 2: Similarly, we compare the MPNTrack outcomes between frames 46 - 220 and 46 - 230. It is challenging to discern how the model performs, but with the help of the evaluation metrics plots, users can quickly identify the model’s weaknesses. In this case, the most obvious underperformance is the model’s failure to detect 10 expected objects. Besides the significant gap between the number of objects identified by the model and the ground truth—which suggests the model’s general performance is not impressive—further insights can be gleaned. We find that the pedestrian with id=4 in MPNTrack model notation loses track by frame 230. Figure 10 shows an overall decrease in HOTA score, indicating worsened performance in frame 230. Specifically, Association Recall (AssRe), Detection Recall (DetRe), and Association Precision (AssPr) decreased by 1.41%, 1.31%, and 1.09% respectively. This decrease highlights the missing objects it previously detected or a failure to maintain their identities consistently. The CLEAR score shows an increase in fragmentation, indicating the incidence of interrupted tracking that is subsequently resumed (see Figure 11). Figure 12 suggests a significant rise in IDFP, indicating potential issues with precision in identity management. Figure 13 shows a slight 3.48% decrease in Advanced Tracking Accuracy (ATA), pointing to a reduction in the model’s ability to maintain accurate and consistent tracking. This scenario of bounding box overlapping, a common challenge for MOT models, can severely impact model performance. Through this case study, we can infer that MPNTrack is not very robust against bounding box overlapping. This information can be conveyed to developers and non-professional stakeholders very efficiently through changes in evaluation metrics and straightforward visualizations.

Overall, we assessed the effectiveness of the MOT16-VisualTracker by comparing the analysis process facilitated by our tool to traditional manual methods through the two case studies presented above. We effectively addressed existing challenges in evaluation. The tool enhances quantitative assessments by automatically counting and displaying the number of detections and IDs across selected frames, allowing developers to swiftly obtain performance metrics without manual effort. The application of the MOT16-VisualTracker not only simplifies the evaluation process through automated metrics but also enriches the analysis by directing attention to critical aspects of model performance.

VI. Conclusion and Future Works

We presented MOT16-VisualTracker as an interactive visualization tool for user to visualize the performance of their MOT model, discover the weaknesses, and strengthen the model. The MOT16-VisualTracker has demonstrated its efficacy through a series of case studies, revealing its ability to deliver deeper, actionable insights that are not easily obtained through traditional evaluation methods. By integrating quantitative metrics with per-frame visual data, our tool allows

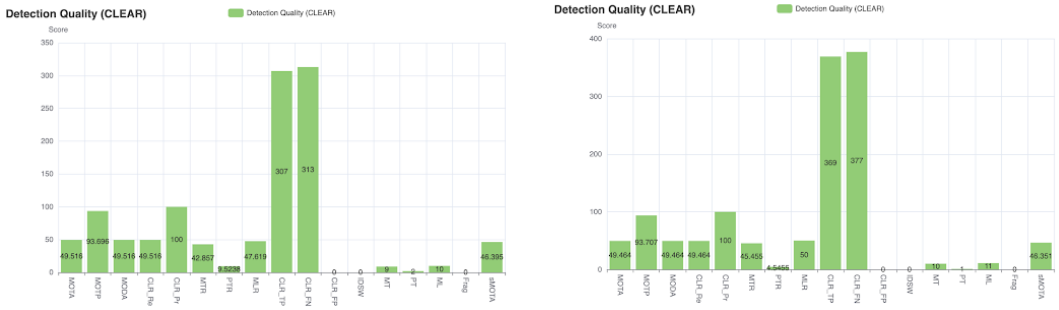


(a) Case 1. Losing track on a specific pedestrian (Frame 30) (b) Case 1. Losing track on a specific pedestrian (Frame 36)

Fig. 3: Case 1 (Frame 30 - 36)

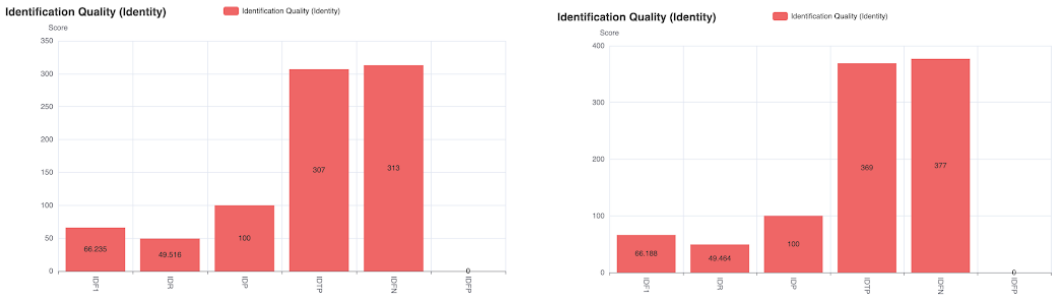
(a) Case 1. HOTA score (Frame 1 - Frame 30) (b) Case 1. HOTA score (Frame 1 - Frame 36)

Fig. 4: Case 1. HOTA scores



(a) Case 1. CLEAR score (Frame 1 - Frame 30) (b) Case 1. CLEAR score (Frame 1 - Frame 36)

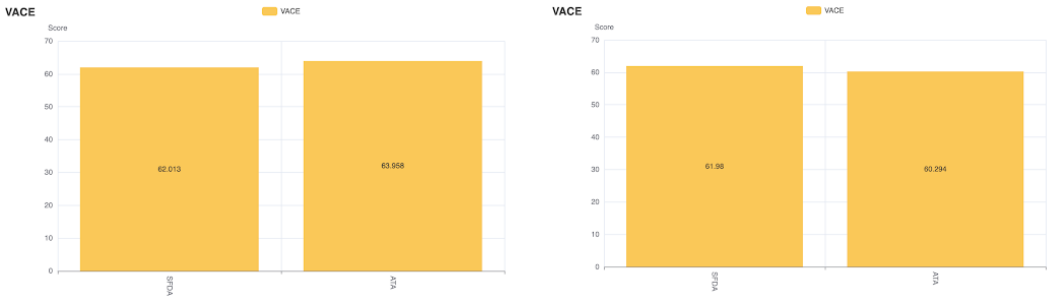
Fig. 5: Case 1. ClEAR scores



(a) Case 1. Identity score (Frame 1 - Frame 30) (b) Case 1. Identity score (Frame 1 - Frame 36)

Fig. 6: Case 1.Identity Scores

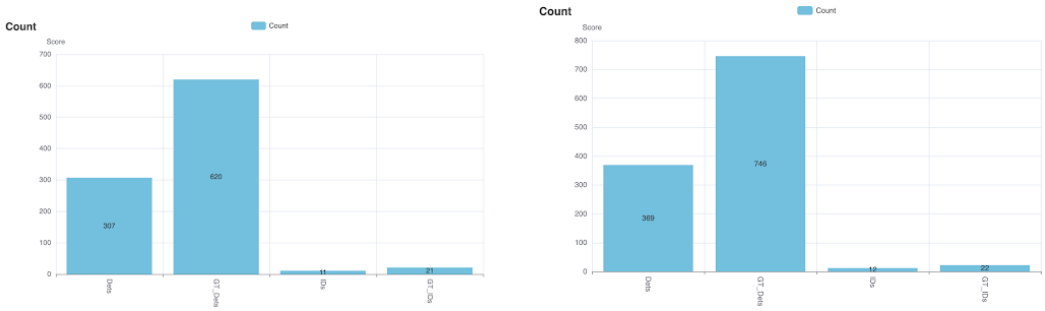
users to intuitively grasp model behavior and identify specific areas of improvement, all within a user-friendly interface that significantly reduces the complexity and time required for analysis.



(a) Case 1. VACE score (Frame 1 - Frame 30)

(b) Case 1. VACE score (Frame 1 - Frame 36)

Fig. 7: Case 1. VACE scores



(a) Case 1. Count of objects (Frame 1 - Frame 30)

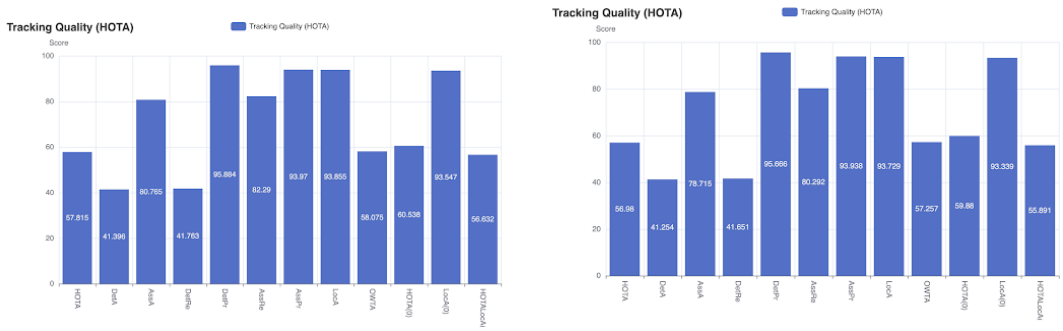
(b) Case 1. Count of objects (Frame 1 - Frame 36)

Fig. 8: Case 1. Counts of objects

(a) Case 2. Losing track on a specific pedestrian (Frame 220)

(b) Case 2. Losing track on a specific pedestrian (Frame 230)

Fig. 9: Case 2 (Frame 220 - 230)

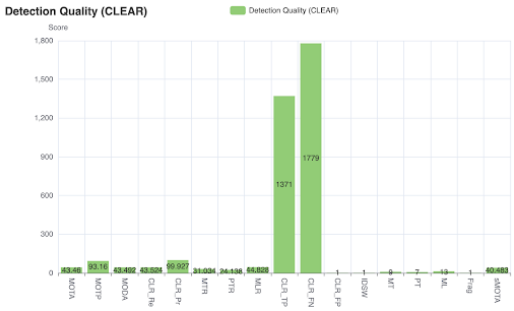


(a) Case 2. HOTA score (Frame 46 - 220)

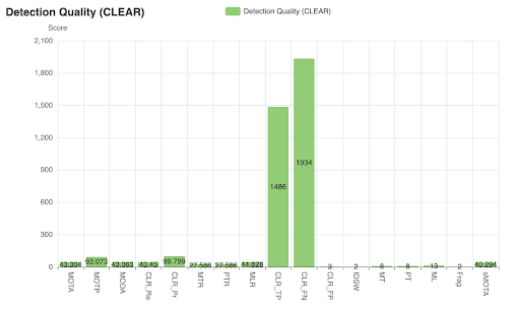
(b) Case 2. HOTA score (Frame 46 - 230)

Fig. 10: Case 2. HOTA scores

Despite its strengths, the current version of the MOT16-VisualTracker focuses primarily on the training dataset due to the availability of ground truth data. To enhance the tool's applicability and utility, developers can create a validation

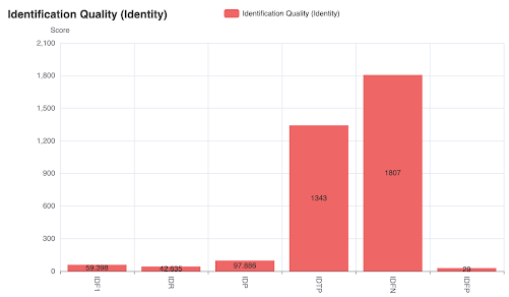


(a) Case 2. CLEAR score (Frame 46 - 220)

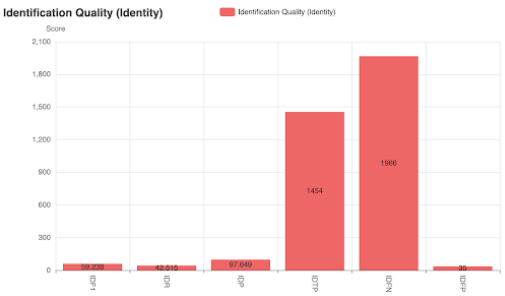


(b) Case 2. CLEAR score (Frame 46 - 230)

Fig. 11: Case 2. CLEAR scores

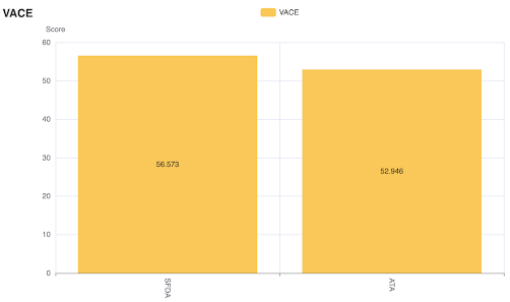


(a) Case 2. Identity score (Frame 46 - 220)

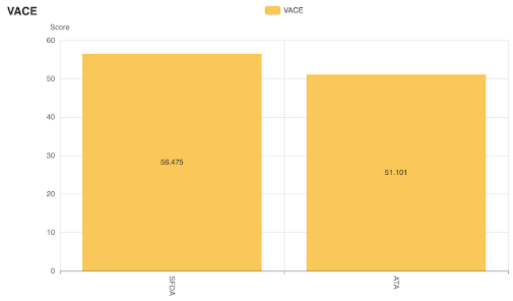


(b) Case 2. Identity score (Frame 46 - 230)

Fig. 12: Case 2.Identity Scores

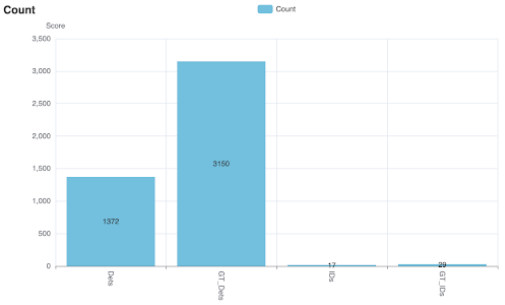


(a) Case 2. VACE score at Frame 220

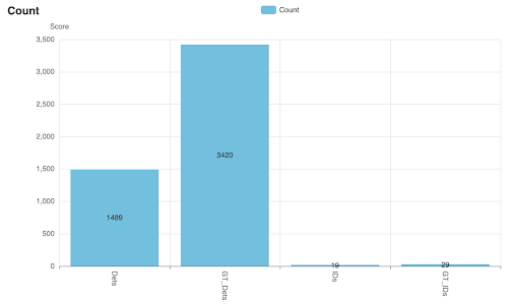


(b) Case 2. VACE score at Frame 230

Fig. 13: Case 2. VACE scores



(a) Case 2. Count of objects (Frame 46 - 220)



(b) Case 2. Count of objects (Frame 46 - 230)

Fig. 14: Case 2. Counts of objects

dataset that allows for rigorous testing outside of the training environment. This addition will provide a more robust evaluation framework, catering to a wider range of scenarios and further improving model reliability and performance.

ACKNOWLEDGMENTS

We would like to extend our gratitude to the developers and contributors of the following projects, whose work has significantly enhanced the functionality of our MOT16-VisualTracker:

1) **[TrackEval]** Our code base is a fork of Jonathon Luiten's TrackEval [5]. We have utilized tools within it extensively to acquire various evaluation metrics. The robust evaluation capabilities provided by TrackEval have been integral in offering comprehensive metric assessments within our tool.

Our project builds upon these outstanding open-source contributions to offer a seamless and powerful tool for MOT model developers. We appreciate the opportunity to utilize such high-quality resources.

REFERENCES

[1] M. Bashar, S. Islam, K. K. Hussain, M. B. Hasan, A. B. M. A. Rahman, and M. H. Kabir, "Multiple object tracking in recent times: A literature review," 2022.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370220301958

[3] Y. Park, L. M. Dang, S. Lee, D. Han, and H. Moon, "Multiple object tracking in deep learning approaches: A survey," *Electronics*, vol. 10, no. 19, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/19/2406

[4] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016.

[5] A. H. Jonathon Luiten, "Trackeval," https://github.com/JonathonLuiten/TrackEval, 2020.

[6] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," 2019.

[7] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," 2022.

[8] 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.

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

[10] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," 2023.

[11] J. Alikhanov and H. Kim, "Online action detection in surveillance scenarios: A comprehensive review and comparative study of state-of-the-art multi-object tracking methods," *IEEE Access*, vol. 11, pp. 68 079–68 092, 2023.

[12] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[13] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[14] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*. Springer, 2016, pp. 17–35.

[15] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, M. Boonstra, and V. Korzhova, "Performance evaluation protocol for text, face, hands, person and vehicle detection & tracking in video analysis and content extraction (vace-ii)," *Protocol Document*, 2005.

[16] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6247–6257.