JETI Journal of Emerging Technologies and Innovations

JETI | Vol. 1 No. 1 (2024)

PAPER

Performance Analysis of YOLOv4-Based Multi-Object Tracker Using SORT

Jelynelle G. Bastasa, Daryl I. Cerina, Lester C. Tubo, and *John A. Bacus

College of Engineering Education University of Mindanao, Philippines

*johnbacus001@gmail.com

ABSTRACT

Object tracking has become fundamental in the technology that greatly contributed to and is widely used in security, health, and many fields in the industry. With computer vision, the technology develops and improves its performance. This study aims to develop an automated system that tracks multiple people simultaneously utilizing the combination of YOLOv4 and SORT algorithms. Moreover, the resulting data were gathered to conclude that the network size of the algorithm used in testing is directly proportional to the accuracy and indirectly proportional to the frame rate. Using NVIDIA GTX 1650 GPU, the system attains 6.67 FPS at 608 network size with Multi-Object Tracking Accuracy (MOTA) and Higher-Order Tracking Accuracy (HOTA) of 27.22% and 22.71%, respectively. It is also expected to improve performance when utilizing a more powerful device and algorithm.

KEYWORDS

artificial intelligence, computer vision, human tracking, multi-object tracking, YOLO

1 INTRODUCTION

In computer vision, visual object tracking is widely used for video surveillance, robotics, human-computer interaction, vehicle navigation, and other applications. In the past two to three decades, visual object-tracking technology has made significant progress and achieved satisfactory results, making object-tracking technology a breakthrough[1]. Object tracking is also widely used in monitoring people, which made value in the industry, security, motion analysis, and other aspects using artificial intelligence to analyze human behavior[2].

Static cameras or cameras attached to a moving object were a solution for using object tracking before. Now, a computer vision with a based algorithm that utilizes the available infrastructure is one of the most anticipated solutions[3]. Moreover, some other object tracking research uses selective algorithms and fixed settings, which results in outperforming other methods. Other studies use the Kalman Filter method to predict the location of each multiple object movement from frame to frame[4]. Others applied deep neural networks for increased accuracy in visual object tracking; however, this resulted in slower and less efficient performance of the tracker[5]. Different algorithms are developed for object tracking. One of them is Simple Online and Real-time Tracking (SORT) algorithm. SORT implements the object tracking by detection method requiring an object detector that supplies detections. It then analyzes the detections and predicts each identity using Kalman Filter and Hungarian algorithm [6]. With Faster-RCNN as an object detector, SORT achieves a Multi-Object Tracking Accuracy (MOTA) of 42.7 and a Higher-Order Tracking Accuracy (HOTA) of 36.1 using the MOT20 dataset from MOTChallenge [7]. Faster-RCNN is an object detection algorithm that utilizes region proposal networks for prediction. This algorithm attains a mean Average Precision (mAP) of 70.4% on a Pascal VOC dataset with an average speed of 5 frames per second (FPS) on the Tesla K40 GPU [8].

Systems of previous studies pair the object tracker with an object detector that utilizes region proposal networks which consumes a significant amount of time due to a selective search of features and individually predicts each proposal [9]. In addition, it requires a very powerful device to have a reasonable frame rate. Moreover, it will become useless in real-time applications, especially when used in a much less powerful device. To overcome the limitation, this study attempts to pair the SORT algorithm with a one-stage detector such as You Only Look Once (YOLO), specifically YOLOv4. One-stage detectors only require a single pass through the neural network which means much faster. YOLOv4 obtained an mAP of 65.7% with 62 FPS on the COCO dataset using Tesla V100 GPU [10]. YOLOv4 is selected for its balance between accuracy and real-time performance, and advanced data augmentation to enhance detection robustness while maintaining high FPS, making it well-suited for integration with SORT [11].

The general objective of this study is to develop an automated system that tracks multiple people at once using the combination of SORT and YOLO. The specific objectives of this study are the following: (1) to evaluate the performance of the combination; (2) to determine the frame rate of each configuration; (3) to determine the relationship of network size to the accuracy and frame rate of the system.

By developing an automated system that tracks multiple people at once using SORT and YOLO, the study can significantly enhance the efficiency and effectiveness of surveillance systems. This can lead to better monitoring and security in public spaces such as airports, train stations, and shopping malls. Evaluating the performance of the combination of the SORT tracker and YOLO detector provides a benchmark for future researchers and developers. This can help identify the strengths and weaknesses of the current approaches, paving the way for improvements and innovations in multi-object tracking. Determining the frame rate of each configuration is crucial for applications that require real-time processing, such as autonomous driving and robotics.

The study's findings can help optimize these systems to operate smoothly and efficiently under various conditions. Understanding the relationship between network size, accuracy, and frame rate allows for a better balance between computational load and detection performance. This is particularly important for deploying tracking systems on resource-constrained devices like drones and mobile robots. The study contributes to the field of computer vision by advancing the development of robust and efficient multi-object tracking algorithms. This can inspire further research and development in areas like human-computer interaction, augmented reality, and smart city technologies.

2 MATERIALS AND METHODS

2.1 Research Design

The study made use of the applied research approach. The research aims to evaluate the performance of a multi-object tracking system by applying it to real-world video input and using practical evaluation metrics. This study uses a mid-range graphics card for evaluating the system's performance, NVIDIA's GTX 1650. The frame rate may vary when used in a more or less powerful device. Lastly, the network sizes of YOLOv4 used for this study are 320, 416, 512, 608, 704, and 800. The selection of YOLOv4 network sizes (320 to 800) and SORT parameters was based on prior research, experimental optimization, and hardware constraints. Lower resolutions provided higher FPS but lower accuracy, while higher resolutions improved detection at the cost of speed, allowing for a trade-off analysis between MOTA, HOTA, and FPS. SORT parameters were tuned based on existing studies and empirical testing to minimize ID switches and false negatives, ensuring real-time performance within computational limits.

2.2 Conceptual Framework

This system begins with receiving video input or image sequence, as shown in Figure 1. Every frame will undergo processing such as object detection, non-maximum suppression, and object tracking. The object detector used is You Only Look Once version 4 (YOLOv4), and Simple, Online, and Real-time Tracking for object tracking. Detections are filtered in every frame using Soft Non-Maximum Suppression (Soft-NMS)[12]. The system's output will be a video stream showing the detections with their IDs. The logs of tracks are also recorded because they will be used for evaluation. The average frame rate is also registered as additional data

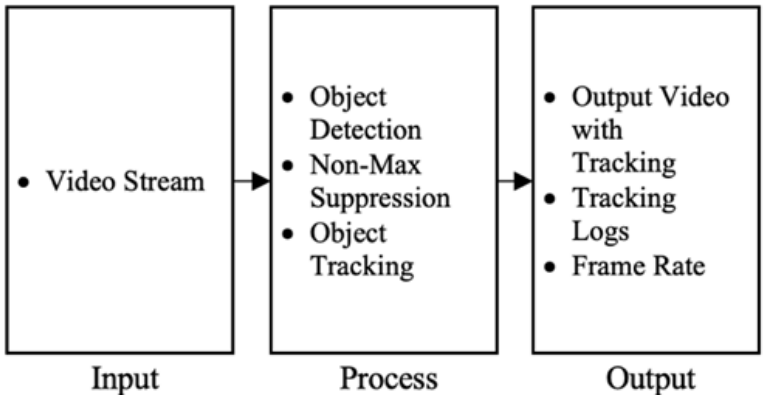

Figure 1. Conceptual framework of the multi-object tracking system.

2.3 Device Used for Testing and Evaluation

This study runs the system in NVIDIA's GTX 1650 GPU with CUDA enabled. TechPowerUp in 2022 [13] mentioned that GTX 1650 is a mid-range graphics card that uses a TU117 processor with 896 cores and a 4 GB memory size. It is ranked 116th out of 685 GPUs based on benchmarks according to UserBenchMark[14].

2.4 Multi-Object Tracking Evaluation

Dataset. This study evaluates the performance of the combination of SORT and YOLOv4 using TrackEval (Luiten, 2020) with MOT20 from MOTChallenge as the dataset[15]. The MOT20 dataset from the MOTChallenge was utilized for evaluating the performance of the multi-object tracking system. This dataset consists of eight challenging video sequences, with four sequences designated for training and four for testing. The sequences are recorded in unconstrained environments, such as crowded train stations and town squares, which present significant challenges for tracking algorithms.

Preprocessing: The video sequences from the MOT20 dataset were preprocessed to match the input requirements of the YOLOv4 detector. Each frame of the video sequences was resized to the network size specified for the YOLOv4 model (ranging from 320x320 to 800x800 pixels). The preprocessing steps included normalization of pixel values and converting the images to the appropriate color space.

Testing Environment: The testing was conducted on an NVIDIA GTX 1650 GPU, which provided a balance between computational power and accessibility. The YOLOv4 model was configured with different network sizes to analyze the trade-off between accuracy and frame rate.

The metrics used for evaluation are CLEAR MOT [16] and HOTA [17]. As shown in Figure 2, the tracking logs output of the system is sent to TrackEval for evaluation and returns the Multi-Object Tracking Accuracy (MOTA) and Higher-Order Tracking Accuracy (HOTA) as a result.

The combination is evaluated with different YOLOv4 network sizes set for comparison. The dimensions used are 320 to 800, with a step size of 96. Network size refers to the resolution of the network used in the neural network of YOLO. The score threshold for detection is set to 20%, which means that any person detected with a score of 20% below is automatically discarded.

As for the SORT configuration, maximum age and minimum hits are set to 30 and 3, respectively. The maximum age refers to the maximum number of frames the track can be missing before it is removed. While minimum hits, on the other hand, refer to the minimum number of associated detections required before tracking can be initialized for that detection.

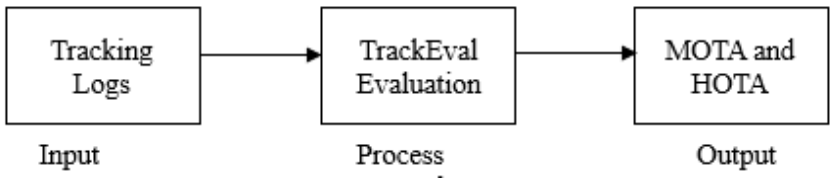

Figure 2. Evaluation process using TrackEval.

3 RESULTS AND DISCUSSIONS

3.1 SORT + YOLOv4 Sample Output

Figure 3 shows the successful implementation of the SORT tracker with YOLOv4 as the detector. A pink bounding box surrounds each person detected and displays the ID number. The ID number corresponds to the assigned track ID showing its unique identity against other detection. ID switching and reassigning usually happen, especially when the detection skips the frame due to occlusions. The yellow and green boxes shows which bounding box was zoomed.

Fig. 1 Sample Output of the YOLOv4+SORT Tracking System: (a) Full-screen output. (b) Magnified yellow box. (c) Magnified green box.

3.2 Multi-Object Tracking Evaluation Result

Table 1 summarizes the system's evaluation in TrackEval using the different configurations in the GTX 1650 device. Since Faster-RCNN is the default pair of SORT, its performance is also evaluated in the device for comparison.

Table 1. Evaluation Result of YOLOv4 anf Faster-RCNN Trackers

Tracker		MOTA (%)	HOTA (%)	FPS
YOLOv4	320	11.267	8.9117	12.332
	416	16.845	16.781	10.021
	512	22.651	20.1	8.6015
	608	27.218	22.706	6.6711
	704	30.814	24.828	5.6243
	800	32.348	25.876	4.838
Faster-RCNN		42.7	36.1	3.6919

Figure 4 visualizes the evaluation result of YOLOv4 using different network sizes. HOTA and MOTA increase when the network size increases but in exchange for the frame rate. This depicts that network size is directly proportional to HOTA and MOTA but indirectly proportional to the frame rate. The frame rate is expected to increase when used in a more powerful device.

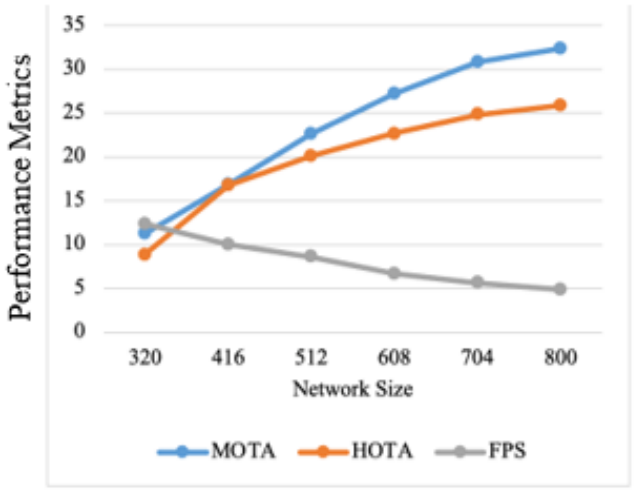

Fig. 4 Evaluation results of YOLOv4 with varying network sizes.

Faster-RCNN outperforms YOLOv4 regarding HOTA and MOTA, but its frame rate is lower. Therefore, YOLOv4 is better when used in a system that requires results immediately, but the device is low to average class.

4 CONCLUSIONS AND FUTURE WORKS

This study successfully developed an automated system that utilizes computer vision to track multiple people in real-time using YOLOv4 for object detection and SORT for tracking. The results demonstrate that network size significantly impacts tracking performance, with larger sizes improving MOTA and HOTA at the expense of a lower frame rate. Using a GTX 1650, the system achieved 12.33 FPS at 320 resolution with a MOTA of 11.27% and a HOTA of 8.91%, while at 608 resolution, it attained 6.67 FPS with a MOTA of 27.22% and a HOTA of 22.71%. Comparatively, SORT with Faster R-CNN provided the highest accuracy (MOTA: 42.7%, HOTA: 36.1%) but at a significantly lower frame rate (3.69 FPS), making it unsuitable for real-time applications on resource-limited hardware. The selection of YOLOv4 network sizes (320 to 800) and SORT parameters was guided by prior research, experimental optimization, and hardware limitations. Lower resolutions were used to maintain high FPS, while larger resolutions were tested to enhance tracking accuracy, allowing for a trade-off analysis between speed and precision. SORT parameters were fine-tuned to reduce ID switches and false negatives, ensuring optimal tracking performance within computational constraints. Overall, the findings suggest that YOLOv4 is preferable when real-time performance is a priority, particularly on mid-range hardware, whereas Faster R-CNN is more suitable for scenarios where accuracy is more critical than speed. Future work can explore more advanced tracking algorithms, hardware acceleration techniques, or hybrid detection models to further enhance tracking efficiency and accuracy.

To further improve the system's performance, use a more accurate object detector algorithm without sacrificing the frame rate. Moreover, improving the object tracker might also boost the accuracy of the overall system. Lastly, testing the system on a wide number of devices could contribute to the development of the study.

5 REFERENCES

[1] Y. Zhang, Z. Chen, and B. Wei, "A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement," in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1312-1316, doi: 10.1109/ICCC51575.2020.9345010.
[2] Z. Feng, X. Zhu, L. Xu, and Y. Liu, "Research on Human Target Detection and Tracking Based on Artificial Intelligence Vision," in 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), 2021, pp. 1051-1054, doi: 10.1109/IPEC51340.2021.9421306.
[3] M. Yaghi, T. Basmaji, R. Salim, J. Yousaf, H. Zia, and M. Ghazal, "Real-time Contact Tracing During a Pandemic using Multi-camera Video Object Tracking," in 2020 International Conference on Decision Aid Sciences and Application (DASA), 2020, pp. 872-876, doi: 10.1109/DASA51403.2020.9317132.
[4]] H. Li, H. Lin, Z. Ruiqiang, L. Lei, D. Wang, and J. Liu, "Object Tracking in Video Sequence based on Kalman filter," in 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), 2020, pp. 106-110, doi: 10.1109/ICCEIC51584.2020.00029.
[5] J. Zhang, T. Chen, and Z. Shi, "A Real-Time Visual Tracking for Unmanned Aerial Vehicles with Dynamic Window," in 2020 China Semiconductor Technology International Conference (CSTIC), 2020, pp. 1-3, doi: 10.1109/CSTIC49141.2020.9282552.
[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and real-time tracking," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003.
[7] J. H. Luiten, "TrackEval," 2020. [Online]. Available: https://github.com/JonathonLuiten/TrackEval.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
[9] J. Hui, "Object detection: Speed and accuracy comparison (faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)," 2019.
[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv [cs.CV], 2020.
[11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.
[12] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS — Improving Object Detection with One Line of Code," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5562-5570, doi: 10.1109/ICCV.2017.593.
[13] "Nvidia GeForce GTX 1650 specs," TechPowerUp, 31-Aug-2022. [Online]. Available: https://www.techpowerup.com/gpu-specs/geforce-gtx-1650.c3366.
[14] "Nvidia GTX 1650," UserBenchmark. [Online]. Available: https://gpu.userbenchmark.com/Nvidia-GTX-1650/Rating/4039.
[15] P. Dendorfer et al., "MOT20: A benchmark for multi-object tracking in crowded scenes," ArXiv, vol. abs/2003.09003, 2020.
[16] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The C LEAR MOT Metrics," Journal of Image and Video Processing, vol. 2008, p. 246309, 2008, doi: 10.1155/2008/246309.
[17] J. Luiten, A. Ošep, P. Dendorfer et al., "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking," International Journal of Computer Vision, vol. 129, pp. 548-578, 2021, doi: 10.1007/s11263-020-01375-2.