MDPI

Article

Investigating Performance of Object Detectors in Multi-Object Tracking

Dheeraj Varma Chittari [1,2], Khurram Azeem Hashmi [1,2,3], Muhammad Zeshan Afzal [1,2,3] and Didier Stricker [1,3]

[1] Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; khurram_azeem.hashmi@dfki.de (K.A.H.); macharav@rhrk.uni-kl.de; didier.stricker@dfki.de (D.S.)
[2] Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; muhammad_zeshan.afzal@dfki.de
[3] German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany

Abstract: Object detection and tracking plays a crucial role in many real-world applications such as activity recognition, self-driving cars, pedestrian surveillance etc. Current state-of-the-art models employ deep learning based approaches to tackle the problem of detecting and tracking objects especially humans from a video sequence. In this paper, we discuss several object detectors and multi-object trackers then we evaluate the tracking performance of widely used multi-object tracking algorithms. Later, we compare the object detections results of two best performing object detectors on standard multi-object tracking datasets MOT17, MOT20 to show that Sparse R-CNN outperforms Faster R-CNN in terms of object detections. Thus we conclude by encouraging to adapt Sparse R-CNN as an underlying object detector for tracking scenarios in order to obtain improved performance and better results.

Keywords: object detector; multi-object tracker; Sparse R-CNN; Faster R-CNN; deep learning; MOT17; MOT20; pedestrian detection and tracking.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Journal Not Specified* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

1. Introduction

In the last couple of decades, there have been significant contributions and advances in the field of computer vision and image processing. This is mainly due to the development of state-of-the-art neural network deep learning models. With increased computational efficiency and parallel processing the design and research towards very deep learning is very active than ever before. The algorithms trained on very deep learning models have outperformed the traditional once, one such network is a convolution neural networks [1].

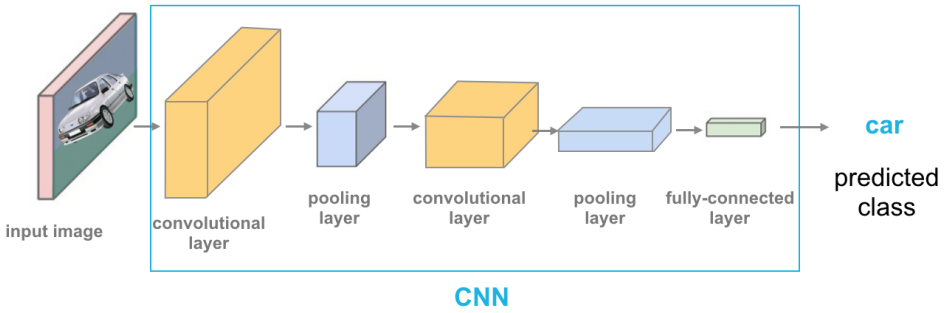

Figure 1. Convolution and pooling layers form the basic building blocks of object detectors in CNNs. Here each image is passed into the blocks of convolution neural network and it outputs the classification and object detection scores. Figure reference[2].

Many real-world applications such as optical character recognition, image recognition, image analysis in the medical domain rely on CNN based deep learning models[3,4]

for superior results and accuracy. CNNs have become reference models in the domain of object detection. Detecting objects from a given image involves two tasks. First correctly identify the object class and then localize the object in a bounding box. Current state-of-the-art object detectors adopts very deep learning paradigm and they can be classified into two categories: one-stage[6,7] and two-stage[3,5] object detectors. Generally, two-stage detectors yield accurate results, but have higher computation time than one-stage detectors. However, this notion depends on the chosen backbone network, hyperparameters[8].

Several computer vision tasks need to understand the scene and context from a video. In most of these tasks, humans are the key objects and we want to track all the pedestrians in a video sequence which give rise to the technique of multi-object tracking. Most of the multi-object tracking algorithms employ tracking by detection paradigm which simplifies the task into two fundamental steps. First we need to detect object locations in each frame and then form tracks by linking corresponding detections across time[9]. Therefore to obtain high accuracy of any multi-object tracking algorithm, the underlying object detector plays a crucial role in identifying all the objects accurately in all the frames. Therefore the detector with higher object detection accuracy also improves the results of object tracking.

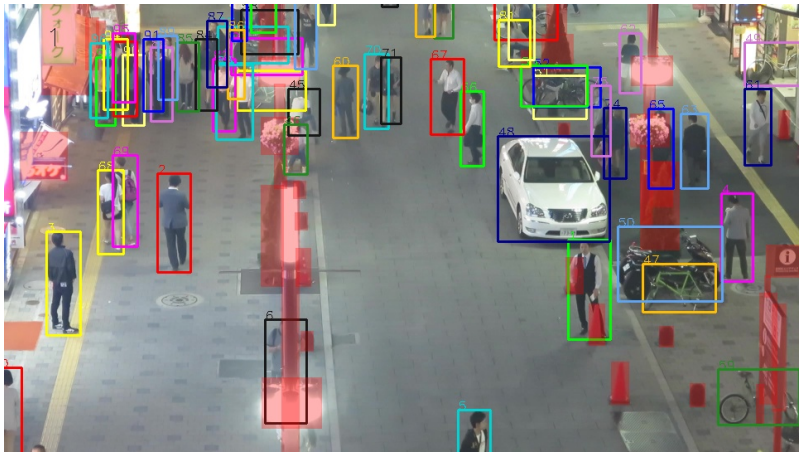

Figure 2. Detecting multiple persons in a single image from a MOT challenge dataset[10].

In this paper, we compare the detection results of two state-of-the-art object detectors, Faster R-CNN[3] and Sparse R-CNN[11] and encourage to use the latter for multi-object tracking algorithms instead of Faster R-CNN for better results. Since Sparse R-CNN has given better detection results on standard MOT challenge datasets[12].

2. Related Work

Multi-object tracking involves object detection and forming a link between the detected objects over all the frames in a video. Object detector estimates the bounding boxes and the links between objects are achieved by data association.

2.1 Detectors in multi-object tracking

i) Two-stage object detectors

For each potential object in an image, first certain proposals are obtained and then these proposals are fed into a neural network to get classification and localization of the object proposals. These proposals are also known as regions of interest. R-CNN[1] uses a selective search algorithm to obtain these proposals. Faster R-CNN[3] uses region proposal network to generate bounding box proposals then the region of interest pooling is applied over each proposal to obtain feature maps and these are passed to

regression and classification heads. The regression head refines the bounding boxes and the classification head gives an object prediction score.

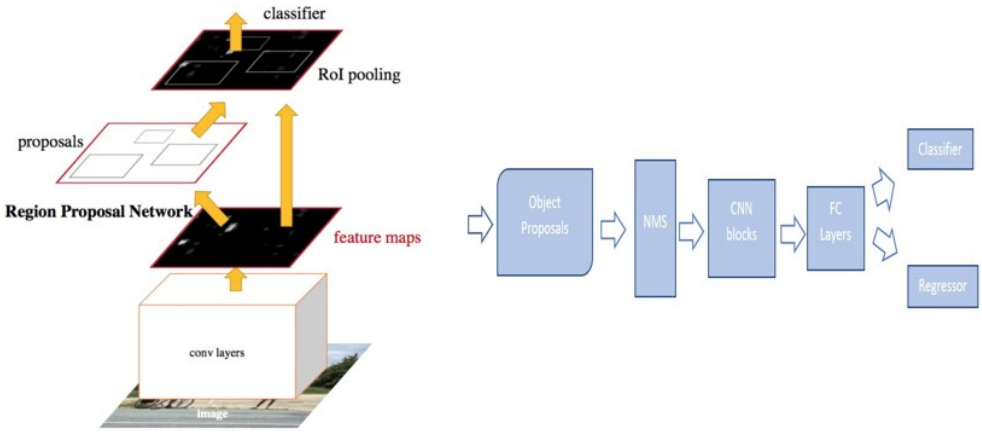

Figure 3. Faster R-CNN with region proposal network with end-to-end training. One the right is the process for region proposal network with sliding window and k anchor boxes to extarct the object proposals.

Faster R-CNN is the base algorithm that has inspired many state-of- the-art object detectors. One such detector is Sparse R-CNN[11] which uses two-stage object detection framework. Sparse RCNN works on a small number of fixed object proposals. These proposals are learned via backpropagation. This method is computationally efficient and accurate because of the small set of learnable object proposals.

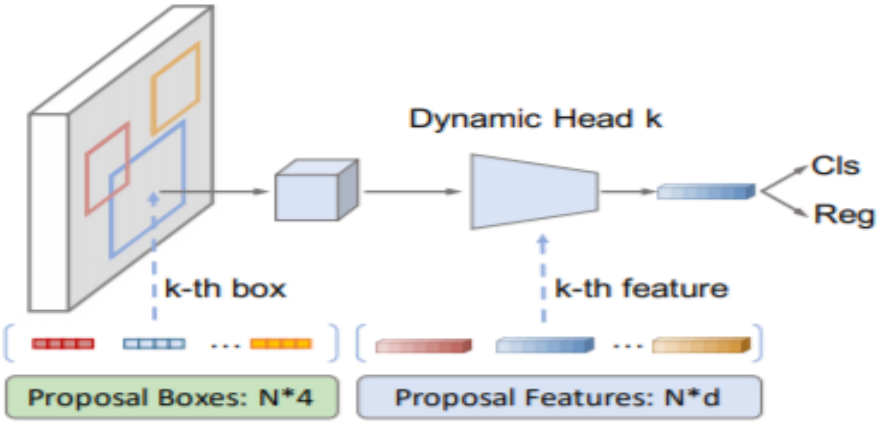

Figure 4. This is the complete Sparse R-CNN object detection pipeline. The proposal boxes and proposal features are learnable parameters. These are passed into dynamic head to obtain localization and classification.The above figure is referenced from original Sparse R-CNN paper[11].

ii) Single-stage object detectors

Single-stage object detectors have a feed-forward CNN that are fully connected which provide object classification and bounding boxes. YOLO[6] (You Only Look Once) and SSD[7] (Single Shot MultiBox) are a couple of algorithms that proposed an architecture without the pre-proposal region computations. YOLO detectors divides the image into a grid of s by s cells where each cell predicts fixed number of bounding boxes and classification probabilities. YOLO detectors have proved to perform faster but not as accurately as two-stage object detectors.

2.2 Multi Object Tracking and Data Association

In general, multi-object tracking methods[15–17] will store the high score detection boxes obtained from object detectors. Low score boxes are ignored because they may contain backgrounds and false positives which may reduce the overall tracking accuracy. These high-score boxes are then used to perform data linking of objects (say pedestrians) between each frames. However, some recent methods such as Bytetrack[18] reject the idea of ignoring the low-score detection boxes and the algorithm considers low-score detection boxes to achieve higher accuracy.

Data association is the key element in multi-object tracking, which matches the detection boxes and predicted tracklets by computing the similarity scores. These algorithms[17,19,20] adopts the combination of position, motions and appearance of an object for tracking because of their efficiency. SORT[21] (Simple Online and Realtime Tracking) technique combines position and motion for tracking. Kalman Filter[22] is used to identify the location of the tracklets in the next frame. The similarity score between predicted and detected boxes is determined by using IoU formula. Intersection of Union[25] computes the intersecting areas between the ground truth and predicted regions.

For short-range tracking, position and motion are highly helpful. For long-range tracking, that is identifying the object after it has been occluded for a long time, appearances similarity is accurate. DeepSORT[23] algorithm uses re-identification model on detection boxes to extract appearance features. Several other tracking algorithms [16,24] achieve more robust results by designing the networks to learn object motions.

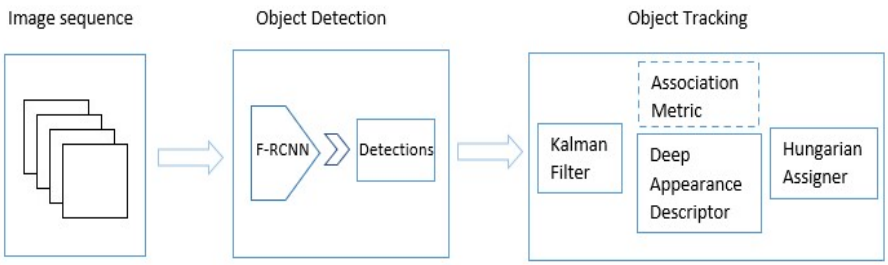

Figure 5. Architecture of DeepSORT Tracking algorithm[14].

Tracking without bells and whistles [9] also known as Tracktor is a distinct tracking technique. Unlike other tracking algorithms[18,23] tracktor does not use position or motion similarity in order to predict the position of an object in the next frame. It instead relies on the bounding box regression of a detector assuming that the target has moved slightly between frames which is achievable through a high frame rate. The principle element is regression based detector. Tracktor predicts the position of an object in the new frame by calculating the bounding box regression, thus forming a trajectory.

Bytetrack is another state-of-the-art object tracker. It uses YOLOX[13] object detector to obtain the detection boxes. The detection boxes with classification scores greater than 0.5 are stored as high-score and scores between 0.2 - 0.5 are stored as low-score detection boxes. The rest are discarded. Then the Kalman filter[22] motion model is employed to predict the objects location in the new frame. It does not use any ReID or attention mechanisms [20,40]. For each frame, first match the tracklets (predicted boxes) to the high score detection boxes based on motion similarity. Then, perform the second matching between the unmatched tracklets and the low score detection boxes to recover true negatives. The motion similarity is computed by the Intersection of Union of the predicted box and the detection box and the Hungarian assigner[41] is used to assign identities to objects between frames.

3. Materials and Method

In this section, we discuss the architectures of several feature extractors such as Feature Pyramid Network, Resnet and Darknet. Later we explain the detailed training procedure that we have followed to carry out our experiments.

3.1. Feature Pyramid Network

FPN[26] is a feature extractor that creates a pyramid of features by generating several layers of activation maps. The effectiveness of hierarchy of object features increases from low-level to high-level layers. Lateral connections in FPN aid the object detectors to predict the bounding box location accurately.

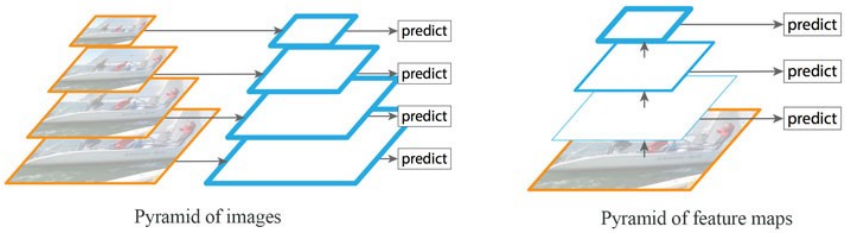

Figure 6. The left figure indicates multiple image scales and from each scale the object regions are extracted. The right figure shows that one image scale is necessary to predict various size objects and thick line indicates that high-level object feature concentration[26].

FPN bottom-up structure is the convolutional network for feature extraction. It has several modules and convolutional layers. The top-down structure applies 1x1 filters for dimensionality reduction and then several 3x3 convolutions are applied followed by upsampling. Finally, we get feature pyramid maps with 256 dimensions outputs. Current state-of-the-art object detectors [3,4,11] uses FPN to generate a pyramid of feature maps then from these maps Regions of Interest are obtained.

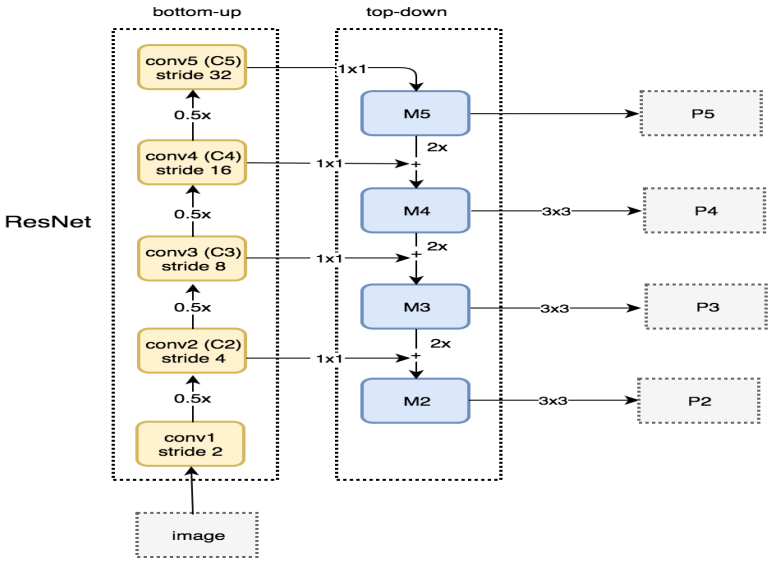

Figure 7. The image show the bottom-up and top-down operations present in FPN given an input image the framework outputs feature pyramids of different scale[P2, P3, P4, P5][27].

3.2. Resnet

Resnet stands for Residual Network. It is a deep neural network introduced to overcome the optimizing problem and vanishing gradients present in its predecessors[34, 35]. The core idea of residual blocks is to connect the input to the output of a layer by skipping a few connections.

In the below figure 8, the input of a layer x is directly connected to output by skipping few connections. Therefore the output will be F(x) + x, where F(x) is the result of the identity function. The dimensions of x and F(x) may not be always equal in order to solve this padding is added input x with weights to have same dimensions as F(x).

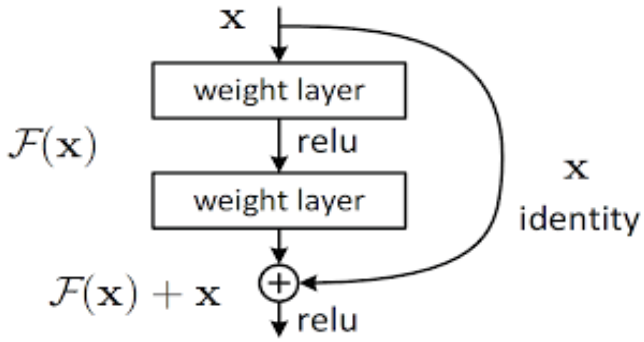

Figure 8. Showing the residual connections which are building blocks of Resnet architecture. The figure is referenced from [32].

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates state-of-the-art algorithms for object detection and image classification[33]. In the year 2015, when Resnet was first introduced it beat the predecessor[35] from the previous year by an accuracy of 46 percent. An ensemble of these residual nets achieves a 3.57 percent error on the ImageNet test set. Resnet has several versions resnet-51,resnet-101 etc the numbers indicate the depth of the layers in the model.

3.3. Darknet

YOLO series detectors[28,29] adapts Darknet network for feature extractors. Darknet-19[30] has 1 × 1, 3 × 3 convolutions with global average pooling and batch normalization. YOLOv3 uses Darknet-53[31] backbone and it has 53 layer convolutions thus much deeper network than Darknet-19. Darknet-53 employs residual and skip connections. The latest YOLOv4[29] uses CSPDarknet-53 which uses a Cross Stage Partial connections (CSP) network strategy to divide the feature map into two segments and concatenates them through a cross-stage hierarchy. The main reason to do split and merge is to allow more gradient flow in the network.

3.4. Training Procedure

We have used the MMDetection[47] and MMTracking[38] public repository extensively to perform our experiments. These repositories implement various object detection and object tracking algorithms using PyTorch deep learning framework. We selected the available models from the repositories to proceed with our experiments.

For the training procedure, we have used the batch size of 4 samples per GPU and we followed the default 1x learning rate schedule that is used in MMDetection[47]. Models are trained for 48 epochs with learning rate decays with a 0.1 factor at epochs 42 and 47. Based on the linear scaling rule[46] the initial learning rate is configured. We employed the SGD optimizer and default anchor configuration. The rest of the hyperparameters follow the values provided in the original papers of the models.

4. Experiments and Results

We introduce this section describing the various datasets used. Then we focus on the different accuracy metrics, their definitions and formulations. Finally, we present the results of our experiments in the tabular form and interpret the results.

4.1. Datasets

4.1.1. MOT Datasets

The multi-object tracking challenge benchmark captures challenging person trackings which have several occlusions and crowded scenes. The person trackings differ in their view angle, camera motion and frame rate. MOT Challenge has MOT17[36] and MOT20[37] separate datasets and their benchmark performance.

The MOT17 dataset has seven different outdoor and indoor scenes of public areas with pedestrians as main focus. Each video is divided into two clips, one for training and the other for testing. The MMTracking[38] library distributes the entire dataset into equal parts of training and validation sets.

Table 1. MOT17 dataset distribution of pedestrian instances in training and validation sets.

	Training	**Validation**	**Total**
Images	8000	7950	15950
Pedestrians	175220	161669	336890

The MOT20 dataset has eight video sequences of which four are train and the other are for test. It has taken from crowded places such as sports stadium, train stations, town squares.

Table 2. MOT20 dataset distribution of pedestrian instances in training and validation sets.

	Training	**Validation**	**Total**
Images	4467	4463	8900
Pedestrians	519476	615136	1134613

4.1.2. CrowdHuman Dataset

CrowdHuman[39] as the name suggests that it captures images from very crowded regions. They contain rich diversity and occlusions. It has a total of 15000 images for training and 4370 images for validation. On average each image contains 23 persons.

Table 3. CrowdHuman dataset distribution of pedestrian instances in training and validation sets.

	Training	**Validation**	**Total**
Images	15000	4370	19370
Pedestrians	438790	127710	566500

4.2. Formulations

The following measures and formulas are used during the experiment procedures.

4.2.1. Intersection over Union

Intersection over Union[43] calculates the intersection between the predicted and the ground truth regions. The formula is:

$$IoU = \frac{\text{Area of Overlapping Region}}{\text{Area of Union}}$$

4.2.2. Precision

The precision computes the ratio of true positive samples to the total predicted samples. It is calculated as follows:

$$Precision = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

4.2.3. Recall

The recall is calculated as the ratio of true positives over all the correct samples from the ground truth. It is defined as:

$$Recall = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

4.2.4. IFD1

IDF1[45] evaluates the identity preservation ability. It gives association performance.

4.2.5. MOTA

The Multi Object Tracking Accuracy[44] (MOTA) is a standard metric to calculate tracking performance. It can be calculated as follows:

$$MOTA = 1 - \frac{\sum(FN + FP + MME)}{\text{Total number of objects}}$$

where FN - False Negatives, FP - False Positives, MME - Total mismatches.

4.3. Results

4.3.1. Multi object trackers performance

Here we evaluate the different multi-object trackers performance over various tracking metrics. To perform this we have used resnet-101 as a pre-trained model and Faster-RCNN as object detector for SORT, DeepSORT, Tracktor and YOLOX for ByteTrack with four Quadro RTX 6000[42] GPUs.

Table 4. Performance of multi-object trackers on various evaluation metrics over MOT17 dataset.

Method	MOTA	IFD1	FP	FN	IDSw
SORT	61.0	55.4	17243	47090	5780
DeepSORT	63.4	67.9	15530	44060	3409
Tracktor	64.0	66.2	11560	49088	1375
Bytetrack	74.1	71.8	14237	23657	872

4.3.2. Detection results of Faster-RCNN and Sparse-RCNN

In this section, we compare the object (here pedestrian) detection results of two prominent object detection methods: Faster-RCNN and Sparse-RCNN. Both uses resnet-101 as a pretrained model. The models are trained on evaluated on both MOT17 and MOT20 datasets.

Table 5. Comparison of Faster-RCNN and Sparse-RCNN detection results over MOT17 dataset.

Method	AP_s	AP_m	AP_l	AP_{50}	AP_{75}	AR_s	AR_m	AR_l
Faster R-CNN	0.63	0.61	0.80	0.81	0.69	0.71	0.75	0.85
Sparse R-CNN	0.68	0.63	0.82	0.90	0.77	0.79	0.77	0.86

Table 6. Comparison of Faster-RCNN and Sparse-RCNN detection results over MOT20 dataset.

Method	AP_s	AP_m	AP_l	AP_{50}	AP_{75}	AR_s	AR_m	AR_l
Faster R-CNN	0.30	0.48	0.67	0.61	0.60	0.31	0.52	0.70
Sparse R-CNN	0.32	0.73	0.80	0.95	0.89	0.42	0.80	0.86

Generally, object detectors performance is measured in average precision and recall of detected objects to the ground truth objects present in each frame. Here the subscripts s(small), m(medium), l(large) indicates the relative size of objects and the subscripts 50, 75 represents the Intersection of Union[43] ratio. From the above tables 5, 6 it is clear that the Sparse R-CNN has generated better results in all the small, medium, large object categories. Overall Sparse R-CNN has outperformed Faster R-CNN on every accuracy metric thus making it the superior object detector even on training with the multi-object tracking datasets MOT17, MOT20.

5. Conclusion

We present an evaluation study comparing the performance of several state-of-the-art deep learning based object detection algorithms in the context of multi-object tracking. In the related work section, we provided a concise review of the object detectors and object trackers that are currently employed by the community. Next, we evaluated several aspects of feature extractors based on convolutions for images involving multiple objects. Then, we analysed the performance of several popular multi-object trackers such as SORT[21], DeepSORT[23], Tracktor[9], Bytetrack[18] over different tracking metrics (MOTA, IFD1, FP,FN) on MOT17 dataset.

The performance of an object tracking system depends on the accuracy of the underlying object detector. The more accurate object detection are passed to the object tracker the higher the tracking results will be. Since the detected bounding box results are compared with the predicted boxes using a similarity metric. Therefore we performed an experiment to compare the detection results of current state-of-the-art object detector Sparse R-CNN[11] with Faster R-CNN[3] and analysed the results in above tables 5, 6. The reason to choose Sparse R-CNN with Faster R-CNN is that the former has learnable object proposals which gives better detections while the latter is employed as a base detector in many object tracking algorithms. From this experiment, we show that the Sparse R-CNN provides better detection results than Faster R-CNN on multi-object tracking datasets such as MOT17, MOT20. As future work, we propose that employing Sparse R-CNN in top performing object trackers can improve their tracking accuracy significantly.

References

1. LeCun Y, Bengio Y, Hinton G. Deep Learning Nature 2015, 521, 436–444.
2. Cezanne Camacho (accessed on 28.02.2022) https://cezannec.github.io/Convolutional_Neural_Networks/
3. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks arXiv-2015, arXiv:1506.01497.
4. Zhaowei Cai, Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. arXiv-2017, arXiv:1712.00726.
5. Ross Girshick. Fast R-CNN arXiv-2015, arXiv:1504.08083.
6. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. arXiv-2016, arXiv:1506.02640.
7. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector. arXiv:1512.02325.
8. Carranza-García, Manuel and Torres-Mateo, Jesús and Lara-Benítez, Pedro and García-Gutiérrez, Jorge. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. https://www.mdpi.com/2072-4292/13/1/89 - 2021
9. P. Bergmann, T. Meinhardt, and L. Leal-Taixe. Tracking without bells and whistles. In ICCV, pages 941–951, 2019.
10. JonathonLuiten (accessed on 28.02.2022) https://github.com/JonathonLuiten/TrackEval/blob/master/docs/MOTChallenge-Official/Readme.md.
11. Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, Ping Luo. arXiv 2021, arXiv:2011.12450.
12. MOT Challenge Datasets (accessed on 28.02.2022) https://motchallenge.net/

13. Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun. YOLOX: Exceeding YOLO Series in 2021. arXiv:2107.08430, 2021.
14. DeepSORT — Deep Learning applied to Object Tracking by Ritesh Kanjee (accessed on 01.03.2022) https://medium.com/augmented-startups/deepsort-deep-learning-applied-to-object-tracking-924f59f99104.
15. Z. Lu, V. Rathod, R. Votel, and J. Huang. Retinatrack: Online single stage joint detection and tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14668–14678, 2020.
16. P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo. Transtrack: Multiple-object tracking with transformer. arXiv preprint arXiv:2012.15460, 2020.
17. Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang. Towards real-time multi-object tracking. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, pages 107–122. Springer, 2020.
18. Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, Xinggang Wang. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. arXiv:2110.06864, 2021.
19. C. Liang, Z. Zhang, Y. Lu, X. Zhou, B. Li, X. Ye, and J. Zou. Rethinking the competition between detection and reid in multi-object tracking. arXiv preprint arXiv:2010.12138, 2020.
20. Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. arXiv preprint arXiv:2004.01888, 2020.
21. A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In ICIP, pages 3464–3468 IEEE, 2016.
22. R. E. Kalman. A new approach to linear filtering and prediction problems. J. Fluids Eng., 82(1):35–45, 1960.
23. N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP), pages 3645–3649. IEEE, 2017.
24. X. Zhou, V. Koltun, and P. Krahenb ¨ uhl. Tracking objects as ¨ points. In European Conference on Computer Vision, pages 474–490. Springer, 2020.
25. Blaschko, M.B.; Lampert, C.H. Learning toocalize objects with structured output regression. In Proceedings of the European Conference Computer Vision, Marseille, France, 12–18 October 2008; pp. 2–15.
26. Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature Pyramid Networks for Object Detection. arXiv:1612.03144, 2016.
27. Jonathan Hui. (accessed on 28.02.2022) https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c
28. Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement. arXiv:1804.02767, 2018.
29. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934, 2020.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
31. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.
32. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.Deep Residual Learning for Image Recognition. arXiv 2015, arXiv:1512.03385.
33. Olga Russakovsky and Jia Deng and Hao Su and Jonathan Krause and Sanjeev Satheesh and Sean Ma and Zhiheng Huang and Andrej Karpathy and Aditya Khosla and Michael Bernstein and Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), volume 115, number 3, pages 211-252.
34. Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition.arXiv-2014, arXiv:1409.1556.
35. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. arXiv-2014, arXiv:1409.4842.
36. Anton Milan, Laura Leal-Taixe, Ian D. Reid, Stefan Roth, ´ and Konrad Schindler. Mot16: A benchmark for multi-object tracking. arXiv:1603.00831, 2016.
37. endorfer, P. and Rezatofighi, H. and Milan, A. and Shi, J. and Cremers, D. and Reid, I. and Roth, S. and Schindler, K. and Leal-Taixe L. MOT20: A benchmark for multi object tracking in crowded scenes. arXiv: 2003.09003, 2020.
38. MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. (accessed on 02.03.2022) https://github.com/open-mmlab/mmtracking
39. Shao, Shuai and Zhao, Zijian and Li, Boxun and Xiao, Tete and Yu, Gang and Zhang, Xiangyu and Sun, Jian. CrowdHuman: A Benchmark for Detecting Human in a Crowd. arXiv preprint arXiv:1805.00123, 2018.
40. J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu.Quasi-dense similarity learning for multiple object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 164–173, year 2021.
41. H.W. Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, year 1955.
42. NVIDIA RTX 6000 GPU. (accessed on 01.03.2022) https://www.nvidia.com/en-us/design-visualization/quadro/rtx-6000/
43. Blaschko, M.B.; Lampert, C.H. Learning toocalize objects with structured output regression. In Proceedings of the European Conference Computer Vision, Marseille, France, 12–18 pp. 2–15; October 2008.
44. Rangachar Kasturi, Dmitry B. Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John S. Garofolo, Rachel Bowers, Matthew Boonstra, Valentina N. Korzhova, and Jing Zhang. Framework for performance evaluation for face, text and vehicle detection

and tracking in video: data, metrics, and protocol. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 31(2):319–336, year 2009.
45. E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multicamera tracking. In ECCV, pages 17–35. Springer, year 2016.
46. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. arXiv:1706.02677, arXiv year 2018.
47. Chen, K.;Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng,W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. arXiv:1906.07155, arXiv year 2019.