Accuracy metrics for object detection, tracking and re-identification.

E.W.J. Bangma, B. van Beusekom, B.A.C. Dekker, R. Fidder,
M. van der Hart, T.A.W. van Kemenade, J.T.S. Kwa, M.D. van der Plaat,
G.J. van Schie, J. Teeuwissen, L. Xu

Software Project Computer Science,
Utrecht University
Department of Computer Science

Key words: computer vision; surveillance systems; object tracking accuracy; multi-object detection; multi-object tracking; object re-identification; real-time tracking; real-time object re-identification.

How to cite this article:
E.W.J. Bangma, B. van Beusekom, B.A.C. Dekker, R. Fidder, M. van der Hart,
T.A.W. van Kemenade, J.T.S. Kwa, M.D. van der Plaat, van Schie G.J., J. Teeuwissen, L. Xu
Accuracy metrics for object detection, tracking and re-identification..
Department of Computer Science, Utrecht University
https://github.com/UU-tracktech/tracktech

Abstract

This paper focuses on a brief overview of accuracy metrics for object detection, tracking and re-identification. The paper concerning TrackTech [1], an open-source repository aimed at creating a scalable system that can be connected to cameras and used for the real-time following of objects across multiple cameras. The metrics were split across detection, tracking, and re-identification. The team used the Mean Average Precision metric for detection. YOLOv5 [2] scored 0.509, and YOLOR [3] scored 0.622. These results are worse than the top-scoring algorithms for object detection [4], but this is due to their real-time constraint. The ClearMOT metric was used for tracking accuracy. SORT [5] scored 35.871 and SortOH [6] scored 12.174. These results are quite low, but this is mainly due to the bad performance of the detection stage. Finally, the team used the Mean Average Precision metrics and the Rank-1 metric for re-identification. Torchreid [7] scored 82.5% mAP and 94.4% Rank-1, while Fastreid [8] scored 89.96% mAP and 95.69% Rank-1. These results are above average when compared to other re-identification algorithms [9].

Contents

1. Introduction

The performance of the final product depends on the performance of its individual components. This holds for speed but accuracy as well. For this purpose, it would be nice to define clear measurements for algorithmic accuracy performance.

This document describes accuracy metrics that the team can use to measure and compare the accuracy of different algorithms of all three stages (detection, tracking and re-identification). The chapters for each stage briefly describe the accuracy metrics used in research for that stage and free-to-use data sets and benchmarks available for measuring accuracy.

For the object tracking system (note that object tracking is used as an umbrella term in this introduction, not related to the tracking stage in the product), the system should measure visual object tracking accuracy. Visual object tracking is relatively new in the field of computer vision. However, there have been tremendous improvements in visual object tracking over the last ten years. Still, there are a few theoretical limitations and growth pains concerning accuracy metrics for detection, tracking and re-identification.

The first issue that the team encountered was that the total system accuracy is largely dependent on the accuracy of the detection stage. The accuracy of the tracking stage can differ up to 25% depending on the detection algorithm used. Using a standard set of detections to measure the accuracy would introduce bias favouring tracking algorithms that can better handle the errors most commonly found in the standard set of detections.

The second issue that the team encountered was that almost all accuracy metrics regarding tracking are made for object tracking systems that don't use a re-identification stage. A few initiatives propose a standard measurement for tracking with re-identification or tracking with re-identification on multiple cameras. Still, these measurements are wildly divergent, and a scientific consensus has not yet been reached.

The team encountered the third issue with training data; accuracy cannot accurately represent the program's quality when the accuracy is calculated on the same data the algorithm is trained on.

2. Terminology

Accuracy uses a lot of jargon. However, this terminology is the same for all accuracy sub-parts (detection, tracking and re-id). Therefore this section will explain some terms to make the accuracy section easier to read.

Term	**-**	**Definition**
Box/Bounding box	-	A rectangle around an object in the image, which represents the position and size of said object.
Ground truth (gt)	-	A file that contains the correct boxes for a test set.
Prediction	-	The boxes that this project generates using different algorithms.
True positive (TP)	-	A box that has been predicted and present in the ground truth.[10]
False-positive (FP)	-	A box has been predicted and is not present in the ground truth.[10]
True negative (TN)	-	A box was not predicted and is not present in the ground truth.[10]
False-negative (FN)	-	A box that was not predicted but should have been.[10]
Classification	-	determines what type of instance an instance is.
Intersection over Union (IoU)	-	Describes the overlap between the predicted bounding box and the grounding truth bounding box. See figure 1 for a visual explanation of IOU.
IoU threshold	-	When the IoU is lower than the IoU threshold, the algorithm interprets the predicted box. as a False Positive. Otherwise, the algorithm interprets it as a true positive. See figure 2 for a visual example of the IoU threshold.

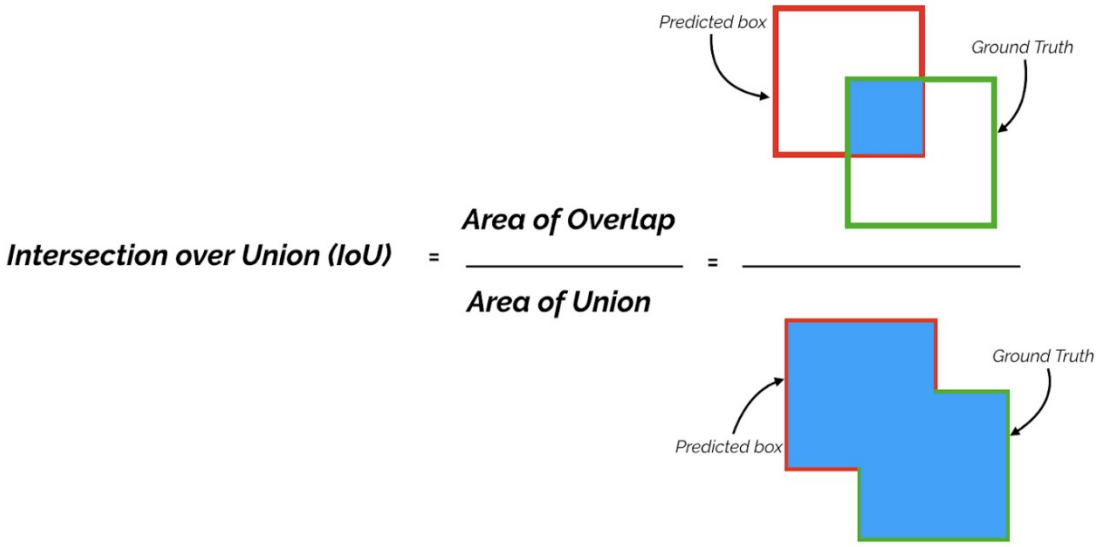

Fig. 1. Visual explanation of intersection over union. Picture taken from [11].

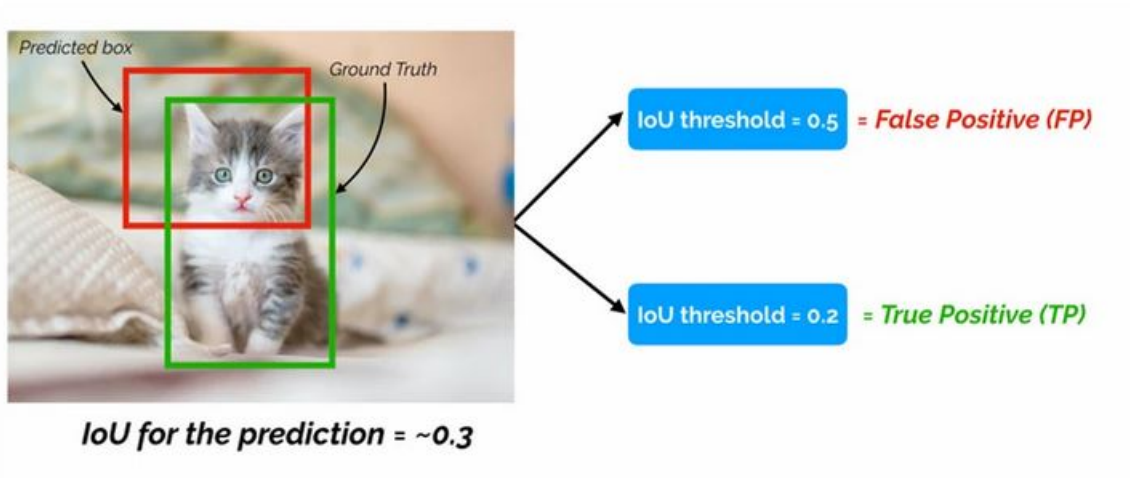

Fig. 2. This is a visual example of the IoU threshold. The IoU for this prediction is approximately 0.3. When the threshold is lower than 0.3, it will be marked as a true positive. Otherwise, it will be marked as a false positive. Picture taken from [11].

3. Detection

The accuracy of the detection algorithm might be the most important out of the three parts. Tracking and Re-id only work with bounding boxes that the detection algorithm has created. Therefore, a bad detection algorithm automatically results in low accuracy for re-id and tracking.

Object detection in computer vision is, compared to tracking and re-id, a relatively well-researched subject. More information is to be found, but with that amount of research also come more different ways to look at the problem.

There are many different accuracy metrics and datasets for detection, but there is no one true standard. The paper will cover some of these metrics and datasets in this section and discuss the choices made in these regards.

3.1 Dataset

The team used COCO for its abundance of pictures and a wide range of used metrics compared to the other datasets. Furthermore, COCO has a built-in data downloader. This downloader allows for the creation of a use-case-specific dataset with relative ease. For example, The team can retrieve a dataset with only pictures of people. Other datasets and their properties are listed in table 1.

Dataset	No. of objects	No. of Images	No. of types	Evaluation method	Paper usage
COCO [12]	1,500k/250k persons	328k/200k persons	80	BoxAP/AP50/more [4]	3806
PASCAL VOC [13]	27,450	11,530	20	BoxAP/mAP [14]	158
Visual Genome [15]	3,800,000	108,077	2,800,000	mAP [16]	471

Table 1. Detection datasets

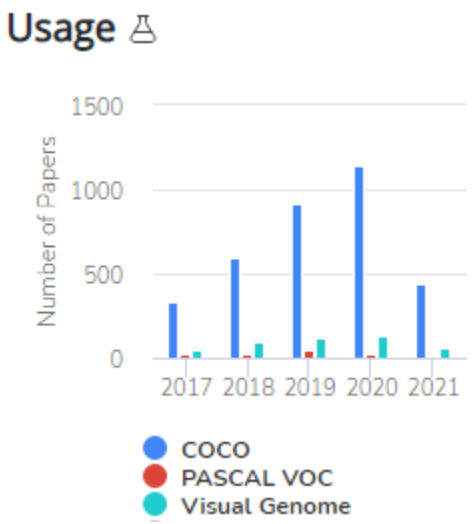

Fig. 3. Usage of the COCO, PASCAL VOC and Visual Genome datasets in research papers.

3.2 Usage Permission

According to the Terms of use on the COCO website [17], The end-user has full responsibility for the usage. All pictures are taken from the image site Flickr [18].

3.3 Metrics

Multiple metrics have been defined for multi-object detection. These metrics can be calculated for an algorithm to compare its accuracy to that of other detection algorithms. This section lists some common metrics for detection algorithms and then discusses the metrics selected for this project (and why).

3.3.1 mAP Metric

Mean average precision (mAP) is the average precision (AP) calculated over all classes and/or overall IoU thresholds, [11]. For this project, mAP is the average of the AP over all classes [19]. When a dataset only contains one class, there is no difference between AP and mAP. Average precision is the area under the precision-recall curve. When precision and recall are plotted against one another, this is called a precision-recall curve. See figure 4 for example, a precision-recall curve.

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

Precision (also known as positive predictive value or PPV) is the percentage of all predicted bounding boxes as positive as true positives. See equation 1.

$$Recall = TPR = \frac{TP}{TP+FN} \tag{2}$$

Recall (also known as sensitivity) is the percentage of all true positives that the algorithm predicted as a positive. See equation 2. [20]

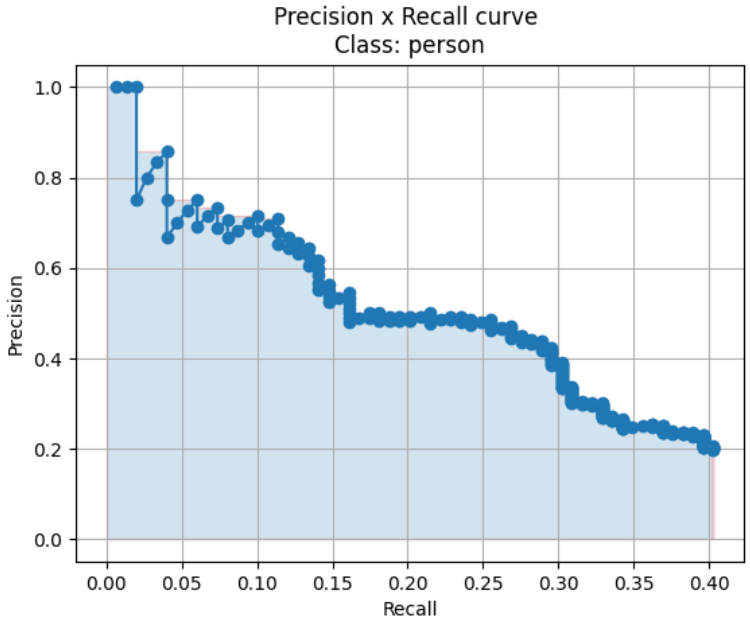

Fig. 4. The precision recall curve for yolov5 for the class person over 50 pictures of the COCO dataset, generated using the podm metric library. [19]

3.3.2 Accuracy metric

The accuracy metric has a confusing name since all the metrics discussed are metrics used for measuring accuracy. This is just another metric for measuring accuracy. The accuracy metric is the percentage of all predictions which was correct. See equation3. This metric makes use of TN's. TN's are hard to evaluate in computer vision. There are endless options for boxes of different sizes and shapes that the program should not detect. Therefore, the amount of TN's is also endless. This creates a distorted evaluation of the algorithms. [21]

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

3.3.3 AUC metric

The Area Under The Curve metric (AUC) is defined as the area under the Receiver Operating (ROC) curve. The ROC curve is a graph that plots the true positive rate (TPR) against the false positive rate (FPR). TPR is the same as recall. See equation 2.

$$FPR = \frac{FP}{TN+FP} \tag{4}$$

FPR is similar to TPR, but FPR takes the percentage of all negatives in the ground truth that have been perceived as a false positive. See equation 4. [22]

FPR makes use of the amount of TN's. However, the amount of TN's is countless because there are countless different locations and sizes of boxes that the algorithm should not detect. Therefore using ROC may cause a distorted vision of how well the algorithm is performing.

3.3.4 Selected metrics

The team has chosen to evaluate the detection algorithm using the mAP metrics. It is a metric that examines both recall and precision. This is important because precision and recall are often "at war" with one another. Decreasing the precision increases recall and vice versa. "To fully evaluate the effectiveness of a model, you must examine both precision and recall. [20]."

Furthermore, mAP is a metric that does not use the amount of TN's. As mentioned above, the number of TN's is countless; therefore, using this in a metric would provide a distorted view of the accuracy.

Lastly, mAP is a metric used as a standard detection metric in many papers that also use COCO as their dataset. Therefore, using mAP makes it easier to compare this project's algorithms to those discussed in those papers. [23]

3.4 Benchmarks

The benchmarks that are selected are for object detection are the Coco [23] and MOT benchmarks [24]. The MOT benchmarks only detect people, so the average precision (AP) is equivalent to the teams preferred metric, mean average precision (mAP). This metric is used due to it being the standard metric for person detections. The coco benchmark is the teams preferred benchmark for multiple object detection accuracy. It is tested as the largest and most used dataset for multi-object detection using the mAP metric.

4. Tracking

As mentioned before, visual object tracking is relatively new in the field of computer vision. There have been tremendous improvements in tracking over the last ten years, but there's still much room for improvement. One area where improvements are needed is the consistency in the accuracy metrics and datasets for multi-object tracking. There have been several initiatives to standardise accuracy metrics, most notable VOT and MOT. Both VOT and MOT will be briefly discussed, and the team will discuss the decisions in using these datasets in the paper.

This section first gives a brief overview of the commonly used datasets, then briefly discusses the most commonly used metrics, and after that, there is a brief discussion about the benchmarks.

4.1 Dataset

Datasets that are useful for the current use case are listed in table 2. MOT20's [25] challenge is based on pedestrian tracking. Because of this, it is solely focused on tracking people. This makes it well-suited for the defined use-case, and therefore the team chose this dataset as its preferred dataset. Its low usages are mostly due to how the recent MOT team issued the challenge.

Compared to the other datasets, MOT20 has some advantages. All other listed data sets(OTB [26], VOT2018 [27] and TrackingNet [28]) are, unfortunately, single-object tracking datasets. However, the defined use case requires multi-object tracking.

Dataset	Number of videos	Avg. number frames	Evaluation method	Paper usage
VOT2018 [27]	?	?	Expected average overlap [29]	89
TrackingNet [28]	30,643	470,9	Accuracy/Precision [30]	158
OTB-2015 [26]	100	?	AUC/Precision [31]	538
MOT20 [25]	8	1676,25	MOTA [32]	3

Table 2. Tracking datasets

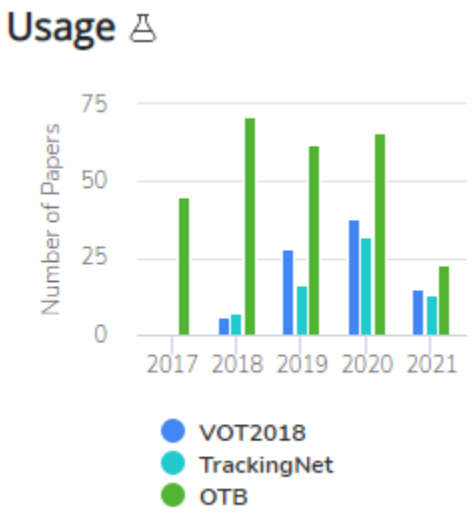

Fig. 5. Usage of the VOT2018, TrackingNet and OTB datasets in research papers.

4.2 Usage Permission

According to MOT20's website, it is for non-commercial use only [33].

4.3 Metrics

This section describes the different metrics used for object tracking.

4.3.1 VOT Metric

The VOT metrics used for the VOT challenge have found increasing popularity over the years because they are easy to interpret, the quality of the metrics and not too expensive to compute. The VOT metrics are designed for trackers that meet the following requirements:

- Single Object
- Does not make use of any past/future frames before initialization

The main idea behind VOT is that most metrics can be divided into two different categories, namely accuracy and robustness. Accuracy focuses mainly on how accurately the tracking algorithm follows an object during successful tracking periods by calculating the average area of overlap [34]. Robustness focuses on how long the tracker can successfully track the object. It is important to note that the VOT guidelines reinitialize the tracker 5 frames after losing the object to better measure the robustness and compensate for lucky and unlucky start sequences.

4.3.2 MOT Metrics

The metrics MOTA and MOTP are commonly used metrics in multiple object tracking papers. They were originally designed in a paper attempting to standardize metrics for multiple object tracking. These metrics over two advantages over other multiple object tracking metrics, namely that they are commonly used and easy to interpret.

4.3.2.1 MOTA

The MOTA metric reflects the ability of the tracker to track an object consistently over a long period of time. The MOTA score ranges from 0 to 100 per cent and is based on the number of misses, mismatches and false positives. Even though this score represents the ability of the tracker to track objects, it is still dependent on the object detection algorithm used. For example, some trackers can better deal with imperfect bounding boxes, while others can better deal with missing detections.

4.3.2.2 MOTP

The MOTP metric determines how well the tracker can estimate exact object positions. This is done by calculating the average error of the position of the bounding box on the correctly tracked frames. This is calculated by using the average area of overlap, which is heavily dependent on the accuracy of the object detection stage.

4.3.3 Selected metrics

The previous section briefly explained what the most promising metrics are. The team have selected them based on both popularity and interpretability. They both try to measure the same thing, namely how well the tracker can track objects and how accurate the location of the tracked objects are.

The main difference between the metrics is in the implementation; whilst VOT tries to measure the accuracy by tracking one person, Clear MOT does this by tracking multiple people simultaneously. The team decided to go with Clear MOT because Clear MOT better fits to determine the accuracy of a multiple object tracking algorithm. Multiple objects tracking better reflects how the tracking algorithm is used.

4.4 Benchmarks

The benchmark that is selected is for tracking is the MOT Challenge benchmark [35]. The MOT challenge focuses on multi-object tracking and ranks the trackers by the preferred metric, MOTA. The team decided to go with this benchmark because it is a commonly used benchmark that uses the teams preferred metric.

5. Re-ID

Re-identification is a relatively new research area. Accuracy measures are therefore also relatively new. However, some tracking measures also include re-identification of object. Therefore there have been earlier use cases for re-identification accuracy measures.

5.1 Dataset

Different re-identification datasets are listed in table 3. The team chose to use the Market1501 [9] dataset. Initially, we chose the Duke [36] dataset, but we chose not to use it due to legal constraints, as described in the section 5.2.1. The other sets contain significantly fewer images and are therefore not used.

Dataset	No. of ID's	Number of Images	No. of cameras	Evaluation method	Paper usage
CUHK03 labeled [37]	1,467	14,097	6	CMC/mAP [38]	20
Market-1501 [39]	1,501	32,668	6	CMC/mAP [40]	87
Duke MTMC [36]	2,700	2,700,000	8	CMC/mAP	70
ViPeR [41]	632	1264	2	-[42]	62

Table 3. Re-identification datasets

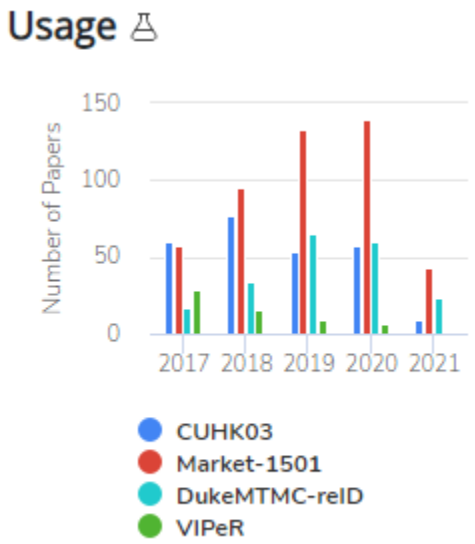

Fig. 6. Usage of the CUHK03, Market-1501, DukeMTMC-reID and VIPeR datasets in research papers.

5.2 Usage Permission

The Market dataset allows for non-commercial use only and requires the original paper to be cited [43].

5.2.1 Notes

The Duke MTMC [36] is by far the largest dataset. Unfortunately, the team should not use it due to IRB violations. You can find more information be found here [44]. . The CUHK03 dataset wasn't chosen due to lower usage in papers when compared to Market-1501 and Duke MTMC. The ViPeR dataset contains pairs of persons under different camera angles. The set is used increasingly less over the past few years. The dataset is rather small.

5.3 Metrics

Two metrics are used in the datasets: CMC and mAP. Both metrics measure the accuracy of the program. The mAP is used more frequently compared to CMC. CMC has three different ways of ranking.

5.3.1 CMC

CMC [45] stands for the cumulative matching curve. It uses Euclidean Distance to determine accuracy. The CMC uses different ways of ranking: top-1 ranking, top-5 ranking and top-10 ranking. top-1 is used most often, top-10 least often. These ranking are defined as follows:

$$Acc_k = \begin{cases} 1 & if\ top-k\ ranked\ gallery\ samples\ contain\ the\ query\ identity \\ 0 & otherwise \end{cases}$$

In other words: if the top k images predicted by the algorithm to contain the query actually contain the subject, CMC is equal to 1. The datasets use varying methods to calculate the CMC. This makes it hard to get a consistent metric across papers.

5.3.2 mAP

mAP stands for mean average precision. The mean Average Precision or mAP score is calculated by taking the mean AP over all classes and/or IoU thresholds, depending on different detection challenges. When applied to the formulated use case, namely the detection, tracking and re-identification of people, the mean average precision becomes a consistent metric across a large range of papers.

5.3.3 Selected Metrics

The team used mAP and the Rank-1 metric since they are the most commonly used metrics for re-identification algorithms.

5.4 Benchmarks

The benchmark that is selected for re-identification is the mAP metric on the Market-1501 dataset. The reader can find an extensive overview of the results achieved by other papers at the papers with code website [40].

6. Results

This section describes the accuracy results of this project. It discusses the metrics of other algorithms and compares these values to the values found for this project. These discussions take place for each part of the re-identification logic: detection, tracking, and re-identification.

6.1 Detection

At the start of the project, the team decided that they should keep everything generic. This would create a more extendable project. The team has also done this for the accuracy of the detection algorithm. The accuracy is extensible because it is possible to validate it on any dataset using special dataloaders. However, it turned out that many datasets have three types of subsets. There is a training set (for training), a validation set (for checking if the training went well) and an actual test set. The leader-boards in the world of detection are measured using the test sets. These test sets turn out to have hidden annotations. The datasets have their own projects to validate these sets, to combat cheating. Therefore, the data discussed in this section are tested on the validation set of COCO instead of the test set. The validation set is much smaller, and therefore the values found are less representative of the actual accuracy. It would be useful to research the accuracy of the actual test set in the future.

Model	mAP
YOLOR-D6	55.4
YOLOv4-CSP-P7	55.4
EfficientDet-D7x	55.1
YOLOR-E6	54.8
YOLOv4-CSP-P7	54.3

Table 4. The top five real-time object detection models in terms of mAP tested on the COCO dataset. [46]

Average Precision	Yolov5	Yolor
Mean Average Precision	0.509	0.622
Person	0.734	0.760
Car	0.622	0.678
Horse	0.691	0.808
Dog	0.638	0.822
Bicycle	0.503	0.605
Backpack	0.225	0.346

Table 5. Six relevant Average Precision (AP) values for yolov5 and yolor and the mean Average Precision (mAP) of yolov5 and yolor. This mAP is based on a total of 80 classes, not just these 6.

Table 4 contains the top 5 models for real-time object detection models. These models all have an mAP of around 0.55. This project's mAP's are test on another part of the COCO set, and therefore it is not completely fair to compare them. Nevertheless, YOLOR has an mAP of around 0.62, and YOLOv5's mAP is around 0.51. These values are close to the top 5.

When we compare YOLOv5 and YOLOR to one another, YOLOR is clearly more accurate. YOLOR's mAP is higher, and every AP of the 6 object classes that the team has deemed as most important.

6.2 Tracking

The tracking accuracy is measured on different YOLO weights with a confidence threshold of 0.1 on the MOT20 dataset. This gave the following results:

Weights	Sort	SortOH
small	29.314	8.958
medium	33.828	14.335
large	34.571	9.8414
extreme	35.871	12.174

Table 6. The results from Sort and SortOH on the MOT20 dataset. [32]

The results show that the sort algorithm outperforms the SortOH algorithm for every weight size on the MOT20 dataset. This is mainly due to many false negatives detrimental to the SortOH accuracy because SortOH requires a subject to be seen in three consecutive frames to start tracking.

An important side note is that the tracking accuracy does not use algorithms such as Torchreid and Fastreid. This is a deliberate choice because there is no garbage collection for feature maps. Implementing this functionality to improve accuracy would result in the accuracy not being representative of the performance for the intended use case of the program, namely tracking on or a few suspects during long periods of time over multiple cameras.

6.3 Re-identification

The re-identification accuracy is measured on the osnet_x1_0 model for Torchreid and the sbs_R101-ibn model for Fastreid.

Metrics	Torchreid	Fastreid
mAP	82.5%	89.96%
Rank-1	94.4%	95.69%
Rank-5	97.8%	98.66%
Rank-10	98.6%	99.23%

Table 7. The results from Torchreid and Fastreid on the Market1501 dataset. [40]

The results show that Fastreid performs slightly better compared to Torchreid across all metrics. Both algorithms are fairly close in performance; however, different factors should be considered, such as frames processed per second.

7. Conclusion

This document compared the accuracy of the different algorithms used in the TrackTech project. It contains an overview of the different accuracy metrics used for object detection, tracking and re-identification and compares them for the current use case. Different datasets for the problems are also compared and chosen for the current use case.

In the end, for the detection stage, YOLOR was the most accurate algorithm. Further research on detection accuracy using the COCO test dataset is advised, however. In the tracking category, SORT was the most accurate algorithm. However, it should be noted that the accuracy of the tracking stage largely depends on the accuracy of the detection stage, as there are no benchmarks that separate these stages. Fastreid was the most accurate re-identification algorithm, though both re-identification algorithms performed very well.

8. References

[1] Tracktech. https://github.com/UU-tracktech/tracktech, 2021. Online; accessed 9-June-2021.

[2] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0, April 2021.

[3] YOLOR implementation - GitHub. https://github.com/WongKinYiu/yolor, 2021. Online; accessed 10-June-2021.

[4] Object Detection on COCO minival. https://paperswithcode.com/sota/object-detection-on-coco-minival, 2021. Online; accessed 26-May-2021.

[5] SORT: A simple online and realtime tracking algorithm for 2D multiple object tracking in video sequences - GitHub. https://github.com/abewley/sort, 2020. Online; accessed 10-June-2021.

[6] SortOH - GitHub. https://github.com/mhnasseri/sort_oh, 2021. Online; accessed 9-June-2021.

[7] Torchreid: Deep learning person re-identification in PyTorch - GitHub. https://github.com/KaiyangZhou/deep-person-reid, 2021. Online; accessed 10-June-2021.

[8] fast-reid: SOTA Re-identification Methods and Toolbox - GitHub. https://github.com/JDAI-CV/fast-reid, 2021. Online; accessed 10-June-2021.

[9] Papers with Code SOTA Market-1501. https://paperswithcode.com/sota/person-re-identification-on-market-1501, 2021. Online; accessed 21-March-2021.

[10] Google LLC. Classification: True vs. False and Positive vs. Negative. https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative. Online; accessed 9-June-2021.

[11] Shivy Yohanandan and Sabina Pokhrel. mAP (mean Average Precision) might confuse you! https://www.xailient.com/post/map-mean-average-precision-might-confuse-you. Online; accessed 10-June-2021.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2015.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):"303–338", jun 2010.

[14] Object Detection on PASCAL VOC 2007. https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007, 2021. Online; accessed 26-May-2021.

[15] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations, 2016.

[16] Object Detection on Visual Genome. https://paperswithcode.com/sota/object-detection-on-visual-genome, 2017. Online; accessed 26-May-2021.

[17] COCO dataset: Terms of Use. https://cocodataset.org/#termsofuse, 2015. Online; accessed 26-May-2021.

[18] Flickr Terms & Conditions of Use. https://www.flickr.com/help/terms, 2020. Online; accessed 26-May-2021.

[19] https://github.com/rafaelpadilla/Object-Detection-Metrics#precision-x-recall-curve, 2021. Online; accessed 16-June-2021.

[20] Google LLC. Classification: Precision and Recall. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. Online; accessed 9-June-2021.

[21] Google LLC. Classification: Accuracy. https://developers.google.com/machine-learning/crash-course/classification/accuracy. Online; accessed 9-June-2021.

[22] Google LLC. Classification: ROC Curve and AUC. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc. Online; accessed 9-June-2021.

[23] Real-Time Object Detection on COCO. https://paperswithcode.com/sota/real-time-object-detection-on-coco. Online; accessed 10-June-2021.
[24] MOT Challenge Det. https://motchallenge.net/results/MOT20Det/. Online; accessed 11-June-2021.
[25] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes, 2020.
[26] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object Tracking Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.
[27] Alan Lukežič, Luka Čehovin Zajc, Tomáš Vojíř, Jiří Matas, and Matej Kristan. Now you see me: evaluating performance in long-term visual tracking, 2018.
[28] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
[29] Visual Object Tracking on VOT2018. https://paperswithcode.com/sota/visual-object-tracking-on-vot2018, 2020. Online; accessed 26-May-2021.
[30] Visual Object Tracking on TrackingNet. https://paperswithcode.com/sota/visual-object-tracking-on-trackingnet, 2021. Online; accessed 26-May-2021.
[31] OTB. https://paperswithcode.com/dataset/otb, 2020. Online; accessed 26-May-2021.
[32] MOT20. https://paperswithcode.com/dataset/mot20, 2020. Online; accessed 26-May-2021.
[33] MOT: license. https://motchallenge.net/#License, 2021. Online; accessed 26-May-2021.
[34] Aleš Leonardis Luka Čehovin and Matej Kristan. Visual object tracking performance measures revisited, 2016.
[35] MOT Challenge. https://motchallenge.net/results/MOT20/. Online; accessed 11-June-2021.
[36] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking, 2016.
[37] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. In *CVPR*, 2014.
[38] CUHK03 (Chinese University of Hong Kong Re-identification). https://paperswithcode.com/dataset/cuhk03, 2021. Online; accessed 27-May-2021.
[39] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable Person Re-identification: A Benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, 2015.
[40] Market-1501. https://paperswithcode.com/dataset/market-1501, 2021. Online; accessed 27-May-2021.
[41] Doug Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *In IEEE International Workshop on Performance Evaluation for Tracking and Surveillance, Rio de Janeiro*, 2007.
[42] VIPeR (Viewpoint Invariant Pedestrian Recognition). https://paperswithcode.com/dataset/viper, 2021. Online; accessed 27-May-2021.
[43] Market-1501: License (readme.txt). https://www.kaggle.com/pengcw1/market-1501/data, 2018. Online; accessed 27-May-2021.
[44] DUKE MTMC DATASET. https://exposing.ai/duke_mtmc/. Online, accessed 27-may-2021.
[45] Reza Ghiass. How is CMC produced (recognition rate vs Rank) for unknown faces?, 02 2015.
[46] Real-Time Object Detection on COCO. https://paperswithcode.com/sota/real-time-object-detection-on-coco. Online; accessed 25-June-2021.