EasyChair Preprint

№ 13111

Performance Comparison of YOLOv7 and YOLOv8 Using the YCB Datasets YCB-M and YCB-Video

Samuel Hafner, Markus Schneider and Benjamin Stähle

May 20, 2024

Performance comparison of YOLOv7 and YOLOv8 using YCB-M and YCB-Video datasets

Samuel Hafner, Markus Schneider, and Benjamin Stähle

RWU Hochschule Ravensburg-Weingarten University of Applied Sciences,
Doggenriedstraße 70, 88250 Weingarten, Germany

Abstract. In this paper, the two frameworks YOLOv7 and YOLOv8 are compared using two labeled YCB datasets YCB-M and YCB-Video. We provide an additional dataset, called Robot Domain Dataset (RDD), to evaluate the performance of the two YOLO frameworks on a new data domain, to simulate situations were it is not possible to retrain the models due to a lack of data or time. Furthermore, the impact of different amounts of training data on performance is observed. For comparability, the training and validation pipelines are provided. We were able to show that both frameworks perform very well on the datasets we retrained on. But on our new dataset YOLOv7 significantly outperforms YOLOv8 by 22% mean average precision. The division of the datasets, the code of the training and validation pipelines, the trained models and the dataset RDD can be found at https://github.com/iki-wgt/yolov7_yolov8_benchmark_on_ycb_dataset

Keywords: Object Detection · YOLO · Benchmark · YCB Dataset · Service Robotics · MS COCO · MAP

1 Introduction

Manipulation of objects is one of the most important and complex tasks in service robotics and represents the most substantial interaction a robot can have with its environment. Many developers are working to find good algorithms and solutions intended to facilitate the manipulation of objects [3,6]. To better benchmark these algorithms, the YCB Object and Model set was released in 2015, featuring a total of 77 household Items [4]. Before a robot can manipulate objects, it first needs to recognize them, and currently there are hardly any available models that can reliably recognize the objects of the YCB Object and Model set. Each year, newer and improved object detection frameworks are released, including the additions to the YOLO family in 2022 with YOLOv7 [16] and YOLOv8 [12]. The advantage of the YOLO frameworks is their real-time capability and the low computational power required during the detection process. In this paper, the two YOLO frameworks are compared in terms of their performance on the YCB dataset. Two already labeled datasets (YCB-Video [17] and YCB-M [10]) are utilized, covering 21 of the 77 objects. Appendix A show the distribution of the two datasets and the 21 objects who are covered of the two datasets. The

comparison is based on various points. First, the influence of different amounts of data on the two frameworks performance is examined by initially training and comparing the frameworks with the individual datasets (YCB-M and YCB-Video) and also with the combination of the two datasets, by putting the datasets together (See Appendix B). Secondly, the performance is evaluated using test data for the YCB-M and YCB-Video datasets and additionally a specially own labeled test dataset from the robotics context, called Robot Domain Dataset (RDD), which is completely independent of the two datasets. Since the two datasets can never cover all possible situations, the RDD is used to check how the two frameworks behave in a new environment. Each model uses the same training and test pipeline, and no hyperparameters of the respective frameworks are changed, except for the batch size and epochs, which are determined once for all models at the beginning. This examines how the two frameworks function "out of the box."

2 Related Work

The benchmarks for the two Frameworks are by default, based on the MS COCO dataset [12,16]. In August 2023, a comprehensive comparison of "YOLO-based object detection models" titled YOLOBench was published [13]. In this comparison, the developers evaluated various YOLO models (from YOLOv3 to YOLOv8) across four different datasets, on four different hardware platforms, and with different backbones. The four datasets are the VOC dataset [7], the SKU110k dataset [9], the MS COCO dataset [14], and the WIDER FACE dataset [13,18]. In June 2023, another comparison of the YOLO frameworks from YOLOv5 to YOLOv8 in an underwater environment was released [8]. Furthermore, there are additional comparisons between YOLOv7 and YOLOv8, such as helmet detection [2] and smoke and wildfire detection [5]. There are no benchmarks comparing the performance of the two models in a robotics context and there are no YOLOv7 or YOLOv8 models that recognize YCB objects or have been trained on the two datasets. The most recent model released on the YCB-Video dataset is a YOLOX model from the "BOP: Benchmark for 6D Object Pose Estimation" [11] Challenge [15]. Therefore, this paper releases the first YOLOv7 and YOLOv8 models for 21 of the 77 YCB objects. These models are available in our git repository.

3 Methods

In this chapter, a brief overview of the aspects of YOLOv7 and YOLOv8 relevant to this paper is provided. This is followed by an explanation of the two datasets and the Robot Domain Dataset.

3.1 Background

The development of the two YOLO versions (7 and 8) occurred in parallel, as they were created by different individuals. Both implementations were inspired

by YOLOv5 and were released around the same time. This means that a newer YOLO version does not necessarily indicate better performance. The developers of YOLOv7 aimed to increase detection accuracy without requiring additional computing power (inference costs), while reducing the model's parameters. They were able to reduce the parameters by 40% compared to other state-of-the-art models [16]. The developers of YOLOv8 did not publish a paper, their motivation, goals, and the theory behind their work are not clearly evident [12].

3.2 Model Types

Both models feature various model types, which differ in size and, consequently, in performance (the more parameters, the better the performance and the higher the detection time). For example, the YOLOv8n model has only 3.2 million parameters, while their largest model, YOLOv8x, has 68.2 million parameters, but the YOLOv8n model is faster in detection. Overall, YOLOv8 has five different model types which use as input a 640 pixel image size, which is changed by the framework itself in the preprocess. YOLOv7 originally had the same number, but in the meantime, they have narrowed it down to just the standard YOLOv7 and YOLOv7x as model types, which use also as input image size 640 pixel. But YOLOv7 has additional model types that run on an image size of 1280 pixels [16,12].

3.3 The YCB-Video and YCB-M Dataset

The **YCB-Video dataset**, with 92 videos and a total of 133,827 frames, is the largest available labeled dataset of the YCB objects. The dataset contains 21 of the 77 YCB objects. The creators of the YCB-Video dataset provide a 3D model for each object, as well as the 6D poses, 2D and 3D semantics, bounding box labels, and respective depth images per scene. The scenes were recorded with an RGB-D camera Asus Xtion Pro Live and have a resolution of 640 times 480. Each scene features between 3 and 5 objects [17]. The **YCB-M dataset**, in contrast, consists of 32 labeled scenes with a total of approximately 47 thousand frames and 20 instead of 21 YCB objects. The objects are the same as those in the YCB-Video dataset, except for the "Master Chef Can", which was not available at the time of recording. Like the YCB-Video dataset, this dataset also includes 3D models, 6D poses, 2D and 3D semantics, bounding box labels, and depth images for each scene. A unique feature of this dataset is that the scenes were recorded with 6 different cameras simultaneously, providing camera diversity. Each scene features between 3 and 8 objects [10].

3.4 Robot Domain Dataset

As mentioned in Chapter 1, in addition to the test datasets of the YCB-M and YCB-Video data, a new test dataset, called Robot Domain Dataset is created and labeled. The test dataset comprises a total of 3 scenes (couch table, table, and

shelf) with 4 recordings per scene, and in each recording, there are 5 objects, thus covering all 20 objects. The selection of which objects appear in each recording is randomized. All scenes are without the Sugar Box, as it was not available at the time of the recordings. In total, the test dataset contains 259 labeled images and is recorded with the RGB-D camera Asus Xtion Pro Live, which is often used in the robotics field. In Appendix C, few example frames from the test dataset can be seen. The purpose of this dataset is to test the performance of specific frameworks (here YOLOv7 and YOLOv8) in new environments, with different backgrounds, different lighting conditions, etc. Often, it is not feasible to retrain the frameworks on new environments due to a lack of data or insufficient time. For example, when a service robot is in a different household than the training environment, or an autonomous car is in a different city. Therefore, it is interesting to know how the frameworks behave outside of the training environment, and this is what the Robot Domain Dataset aims to achieve.

4 Experiments

4.1 Metric

In this paper, the MS COCO Metric [1] is used, as it is also utilized in the benchmarks of YOLOv7 and YOLOv8 [16,12]. The metric includes the AP@[.5:.05:.95], the PascalVOC metric mAP_{50}, and the strict metric mAP_{75} [7].

4.2 Training & Validation pipeline

To have a better comparability, the training of each model always follows the same procedure. For each model, the corresponding existing MS COCO model is used as a pretrained model. In example, when a YOLOv7x model is trained, the YOLOv7x MS COCO model is taken as the pretrained model. The performance improves by approximately 20% mAP when using an MS COCO model as a pre-trained model compared to not using any pre-trained model, as shown in Appendix D. Another advantage of using a pretrained model is the time savings in training, as we only need to retrain an already finished model (even if it is from a different domain) and thus already achieve good performance in just a few epochs. For the comparison, the respective X model type of YOLOv7 and YOLOv8 are used, as they demonstrably deliver better performance. Similar to the pretrained models, the difference in performance with various YOLOv7 model types was initially tested. Between the standard YOLOv7 model and the YOLOv7x model, is a difference of 4% mAP. This comparison can be seen in Appendix E. The training is conducted over 100 epochs, as this is standard for both frameworks, with a batch size of 40 (maximum gpu usage on our training server). Subsequently, predictions are made on the respective test data for which the model was trained and on the Robot Domain Dataset (refer to Section 3.4). Finally, based on the predictions and the ground truth (GT) labels, the mean average precision is calculated.

5 Results

As already highlighted in Section 4.1, the models are evaluated based on performance. These are presented in detail in this chapter.

5.1 Model Performance

Table 1 displays the results of the model performance on the corresponding test dataset. Table 2 shows the results on our own created test dataset (see Section 3.4). In both tables, the results of the models that performed better on the respective dataset are highlighted in bold. As shown in Table 1, YOLOv8

Table 1. Benchmark of YOLOv7 and YOLOv8 models (100 epochs, batchsize 40) on their corresponding Test Dataset

Train/Test Dataset	Model	mAP	mAP_{50}	mAP_{75}
YCB-M	YOLOv7x	90.58%	**98.43%**	96.72%
YCB-M	YOLOv8x	**91.60%**	98.42%	**96.74%**
YCB-V	YOLOv7x	97.45%	99.25%	99.15%
YCB-V	YOLOv8x	**97.77%**	99.25%	99.15%
Combination	YOLOv7x	96.63%	99.01%	98.90%
Combination	YOLOv8x	**97.00%**	**99.06%**	**98.91%**

performs most of the time better than YOLOv7 on all three test datasets, in terms of mAP, mAP_{50}, and mAP_{75}. Nevertheless the performance on all test datasets by all models is very good (at least 90.58% mAP and up to 97.77% mAP) and on each dataset YOLOv7 and YOLOv8 thus perform most likely the same. The biggest difference between YOLOv7 and YOLOv8 is on the combined dataset with 0.37% mAP. Interestingly, YOLOv7 and YOLOv8 deliver better results when trained and tested only on the YCB-Video dataset than on the combined dataset, although the latter contains more and varied data. This is probably due to the dataset difference between the YCB-M dataset and the YCB-Video dataset. Since the YCB-Video dataset has significantly more data, the respective combination models are better trained on the YCB-Video dataset and thus perform worse on the combination test dataset, as it also contains YCB-M test data. However, the difference in YOLOv8 performance from the YCB-Video to the combination dataset is 0.77% mAP, which could also just be noise. Table 2, on the other hand, indicates that YOLOv7 performs significantly better on the own created dataset (RDD) with all three models. For the combination model, with a total of 66.19% mAP, the model is about 22% mAP better than the YOLOv8 model, and for the mAP_{50}, with a total of 84% mAP, the difference is about 27%. This suggests that YOLOv7 generalizes better and performs very well on completely new and different data respectively different environments,

while YOLOv8 generalizes worse and performs significantly worse across all three models. The best for YOLOv8 is 44.10% mAP on the combination model, and the worst is 13.05% mAP on the YCB-M model. Table 2 also clearly shows the impact of the amount of data on the performance of the models. For YOLOv7, there is a performance difference of 19.41% mAP between the YCB-M dataset with 47 thousand images and the YCB-V dataset with about 133 thousand images, and a performance difference of 23.5% mAP between the YCB-Video and the combination dataset. A similar pattern is observed with YOLOv8.

Table 2. Benchmark of YOLOv7 and YOLOv8 models (100 epochs, batchsize 40) on the Robot Domain Dataset

Train Dataset	Model	mAP	mAP_{50}	mAP_{75}
YCB-M	YOLOv7x	**23.55%**	**33.37%**	**29.08%**
YCB-M	YOLOv8x	13.05%	18.76%	15.00%
YCB-V	YOLOv7x	**42.69%**	**60.17%**	**52.30%**
YCB-V	YOLOv8x	35.37%	48.77%	43.18%
Combination	YOLOv7x	**66.19%**	**84.39%**	**75.29%**
Combination	YOLOv8x	44.10%	57.14%	50.97%

6 Conclusion and Future Work

In this paper, the performance of YOLOv7 and YOLOv8 on the YCB Object and Model set was compared. The comparison was made with the already labeled datasets YCB-M and YCB-Video. Various aspects were addressed, such as the mAP and the influence of different amounts of data. Additionally, a unique test dataset in the robotic context called Robot Domain Dataset, was created to observe how the two frameworks behave in a new environment (other backgrounds, other camera, other lighting conditions, etc.), compared to the two training datasets. This aspect is particularly important for household robots, as they constantly find themselves in new environments with each new household. Models with the MS COCO model as a pretrained model perform about 20% better, and for both frameworks the Model Type X performs the best of all. See Appendix D and E. We was able to demonstrate that both YOLOv7 and YOLOv8 perform very well on the test data of the two datasets YCB-M and YCB-Video, with YOLOv8 most of the time performs better or equal to YOLOv7, as shown in Table 1. We also showed that YOLOv7 performs significantly better on the Robot Domain Dataset, with a peak of about 23.5% mAP performance difference, which speaks to YOLOv7's better generalization and that YOLOv7 performs better in new environments. See Table 2 for this.

7 Appendix

7.1 A

Fig. 1 show the distribution of the two datasets YCB-M and YCB-Video, over the 21 objects.

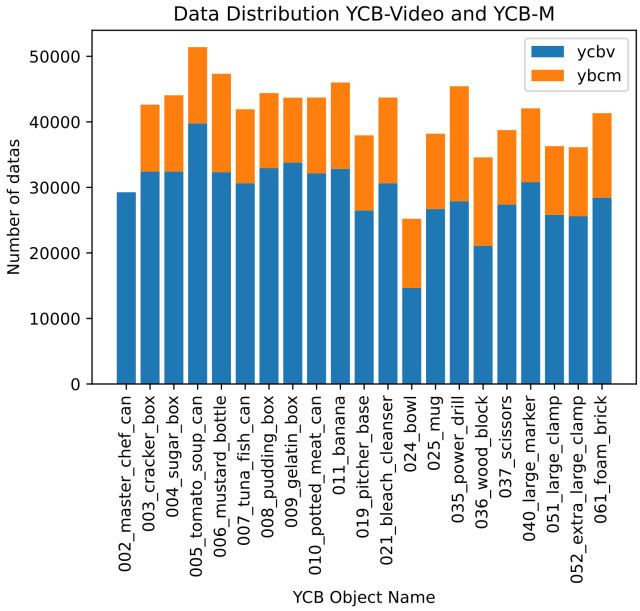

Fig. 1. Data Distribution of YCB-Video and YCB-M. The X-axis indicates the name of each object, the Y-axis shows the frequency of each object in the respective dataset.

7.2 B

Table 3 shows the data distribution of the two datasets, YCB-M and YCB-Video, across training, validation, and test data.

Table 3. Distribution into Training, Validation and Test dataset per dataset

Dataset	**Train**	**Val**	**Test**
YCB-M	38287	4259	4732
YCB-Video	74967	18788	40181
Combination	113254	23047	44913

7.3 C

Fig. 2 shows some example frames of the RDD dataset. There are overall 3 scenes and for each scene are two images shown with different objects.

Fig. 2. Some frames of the test dataset(left: couch table, center: shelf, right: table)

7.4 D

Table 4 shows a comparison of performance between two YOLOv7 models: one trained without a pretrained model and the other with the MS COCO pretrained model. Both models were trained for 10 epochs with a batch size of 40.

Table 4. Benchmark between pretrained and no pretrained YOLOv7 model (10 epochs, batchsize 40)

mAP	mAP_{50}	mAP_{75}	Dataset	Pretrained Model
0.478	0.681	0.544	Combination	None
0.671	**0.877**	**0.778**	Combination	MS COCO

7.5 E

Table 5 shows a comparison of performance between two YOLOv7 models types. One is the YOLOv7 and the other the YOLOv7x model type. Both models were trained for 10 epochs with a batch size of 40.

Table 5. Benchmark between different YOLOv7 model types (10 epochs, batchsize 40)

mAP	mAP_{50}	mAP_{75}	Dataset	Model Type
0.671	0.877	0.778	Combination	yolov7
0.711	**0.909**	**0.809**	Combination	yolov7x

References

1. COCO Detection Evaluation. https://cocodataset.org/#detection-eval, accessed: 2023-09-17
2. Aboah, A., Wang, B., Bagci, U., Adu-Gyamfi, Y.: Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5349–5357 (2023)
3. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M.G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.W.E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-2: Vision-language-action models transfer web knowledge to robotic control (2023)
4. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The ycb object and model set: Towards common benchmarks for manipulation research. In: 2015 International Conference on Advanced Robotics (ICAR). pp. 510–517 (2015). https://doi.org/10.1109/ICAR.2015.7251504
5. Casas, E., Ramos, L., Bendek, E., Rivas-Echeverría, F.: Assessing the effectiveness of yolo architectures for smoke and wildfire detection. IEEE Access **11**, 96554–96583 (2023). https://doi.org/10.1109/ACCESS.2023.3312217
6. Coleman, D., Șucan, I.A., Chitta, S., Correll, N.: Reducing the barrier to entry of complex robotic software: a moveit! case study. Journal of Software Engineering for Robotics **5**(1), 3–16 (may 2014)
7. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge 2012 (voc2012) results (2012)
8. Gašparović, B., Mauša, G., Rukavina, J., Lerga, J.: Evaluating yolov5, yolov6, yolov7, and yolov8 in underwater environment: Is there real improvement? In: 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech). pp. 1–4 (2023). https://doi.org/10.23919/SpliTech58164.2023.10193505
9. Goldman, E., Herzig, R., Eisenschtat, A., Ratzon, O., Levi, I., Goldberger, J., Hassner, T.: Precise detection in densely packed scenes (2019)
10. Grenzdorffer, T., Gunther, M., Hertzberg, J.: YCB-m: A multi-camera RGB-d dataset for object recognition and 6dof pose estimation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE (may 2020). https://doi.org/10.1109/icra40945.2020.9197426, https://doi.org/10.1109%2Ficra40945.2020.9197426
11. Hodan, T., Michel, F., Brachmann, E., Kehl, W., Buch, A.G., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., Sahin, C., Manhardt, F., Tombari, F., Kim, T.K., Matas, J., Rother, C.: Bop: Benchmark for 6d object pose estimation (2018)
12. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics. https://github.com/ultralytics/ultralytics, accessed: 2023-09-17
13. Lazarevich, I., Grimaldi, M., Kumar, R., Mitra, S., Khan, S., Sah, S.: Yolobench: Benchmarking efficient object detectors on embedded systems (2023)
14. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (2015)

15. Liu, X.: gdrnppbop2022. https://github.com/shanice-l/gdrnpp-bop2022, accessed: 2023-09-17
16. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (2022)
17. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes (2018)
18. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)