Using Simulated Data for Deep-Learning Based Real-World Apple Detection

Winter 2023

Dylan Hasperhoven

Supervisors: Maya Aghaei Gavari, Klaas Dijkstra

Introduction

- Deep-Learning based object detectors require a sufficient amount of data.
- Creating a traditional dataset requires manual, time-consuming labour.
- Can simulated data be used as a substitute for real data?
- How does combining real images and simulated images affect the performance of detectors in real scenarios?

Abstract

Creating datasets requires lots of manual and time-consuming labour. Can simulating data be a solution? In this research we introduce a tool that can generate (simulated) datasets containing bounding-box annotated images of apples in orchards. Our experiments conclude that, models trained on simulated data can achieve very similar results as models trained on real-world data on images of real apples.

Materials and Methods

- Apple Orchard simulator made in Unity 3D.
- 3D models of apples are placed in front of a 2D background.

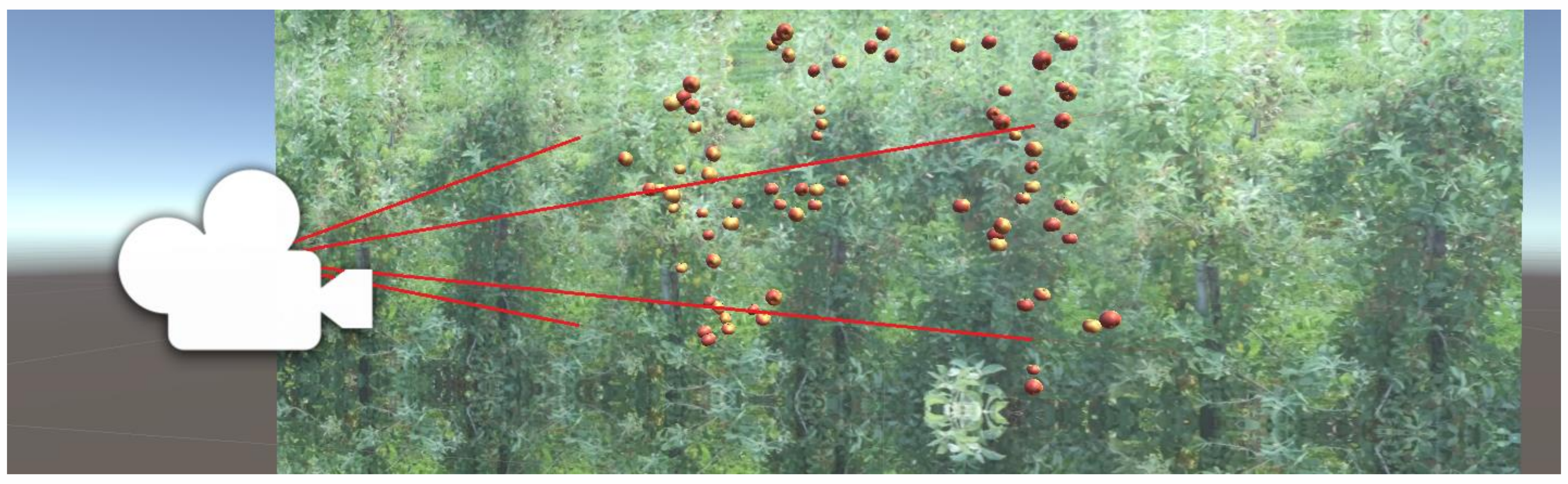

Figure 1. View of the apple orchard simulator. The white icon on the left denotes the virtual camera looking at the 3D apples in front of a 2D plane.

- Augmentations to the simulator introduce variety to the dataset, potentially making the simulation more representative of real data.

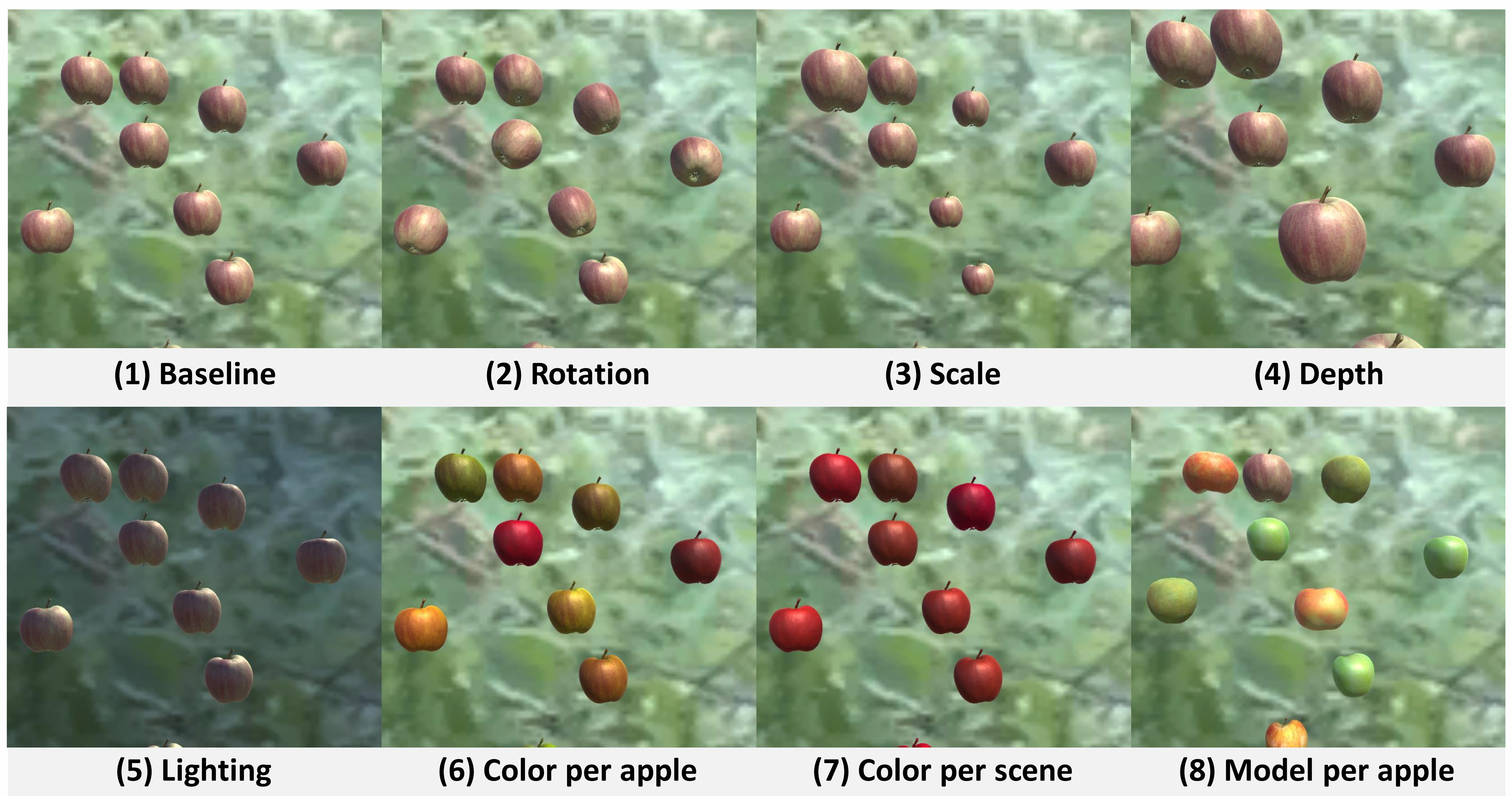

Figure 2. Augmentations to the simulator introduce variety to the dataset, potentially making the simulation more representative of the real data.

- RGB images are generated by randomizing the positions of the camera and apples.
- Bounding-box annotations are generated without human input, by applying inverse camera projection and checking for intersections with apples, for every pixel in the image.
- YOLOv5 [2] object detectors are trained on simulated, real, and hybrid data.
- The Mini-Orchards real-world dataset is used to test our models.

Experiments and Results

- **Experiment 1:** YOLOv5 models are trained on datasets with different augmentations and tested on the Mini-Orchards real-world dataset.

	YOLOv5n		YOLOv5s		YOLOv5m	
Trainset	AP	F1@0.1	AP	F1@0.1	AP	F1@0.1
baseline	0.397 ± 0.315	0.507 ± 0.354	0.927 ± 0.007	0.962 ± 0.004	0.937 ± 0.019	0.950 ± 0.027
color_per_scene	$\mathbf{0.948 \pm 0.004}$	$\mathbf{0.969 \pm 0.009}$	$\mathbf{0.962 \pm 0.004}$	$\mathbf{0.975 \pm 0.009}$	$\mathbf{0.964 \pm 0.006}$	$\mathbf{0.978 \pm 0.005}$
model_per_scene	0.949 ± 0.009	0.965 ± 0.007	0.951 ± 0.011	0.972 ± 0.007	0.952 ± 0.016	0.972 ± 0.007
model_per_apple	0.490 ± 0.355	0.589 ± 0.371	0.942 ± 0.004	0.967 ± 0.006	0.942 ± 0.018	0.967 ± 0.010
color_per_apple	0.447 ± 0.342	0.550 ± 0.349	0.913 ± 0.035	0.956 ± 0.019	0.953 ± 0.010	0.974 ± 0.004
lighting	0.075 ± 0.099	0.133 ± 0.158	0.449 ± 0.335	0.543 ± 0.380	0.928 ± 0.009	0.940 ± 0.048
rotation	0.000 ± 0.000	0.000 ± 0.000	0.326 ± 0.401	0.364 ± 0.406	0.934 ± 0.012	0.965 ± 0.008
scale	0.094 ± 0.197	0.150 ± 0.268	0.316 ± 0.304	0.432 ± 0.327	0.938 ± 0.012	0.968 ± 0.007
depth	0.000 ± 0.000	0.000 ± 0.000	0.328 ± 0.349	0.441 ± 0.350	0.928 ± 0.056	0.950 ± 0.062

Table 1. Comparison YOLOv5 models of different sizes, trained on simulated data with different augmentations.

- 'color_per_scene' models have the best performance.
- Smaller networks need good augmentations to get acceptable performance.

- **Experiment 2:** YOLOv5 models are trained on simulated and real data, both having the same number of images. Trained models are tested o Mini-Orchards.
- Models trained on real data consistently perform better.
- Performance of models trained on simulated data is very similar for the bigger networks.
- Performance of smaller networks is less.

	AP	F1@0.1
YOLOv5m		
Mini-Orchards	0.982	0.983
Simulated-Orchards	0.955	0.967
YOLOv5s		
Mini-Orchards	0.976	0.977
Simulated-Orchards	0.845	0.834
YOLOv5n		
Mini-Orchards	0.972	0.987
Simulated-Orchards	0.789	0.862

Table 2. Comparison YOLOv5 models trained on real and simulated data.

- **Experiment 3:** YOLOv5 models are trained on a combination of real and hybrid data.
- Trained models are tested on Mini-Orchards.
- Performance improves when more real images are added.
- Smaller networks perform almost as good as larger networks when adding real images.

	AP	F1@0.1
YOLOv5m		
Simulated-Orchards	0.955	0.967
Simulated-Orchards + 100 real	0.966	0.983
Simulated-Orchards + 200 real	0.970	0.984
Simulated-Orchards + 300 real	0.970	0.986
YOLOv5s		
Simulated-Orchards	0.845	0.834
Simulated-Orchards + 100 real	0.940	0.960
Simulated-Orchards + 200 real	0.970	0.988
Simulated-Orchards + 300 real	0.970	0.987
YOLOv5n		
Simulated-Orchards	0.789	0.862
Simulated-Orchards + 100 real	0.950	0.943
Simulated-Orchards + 200 real	0.954	0.961
Simulated-Orchards + 300 real	0.964	0.980

Table 3. Comparison YOLOv5 models trained on hybrid data.

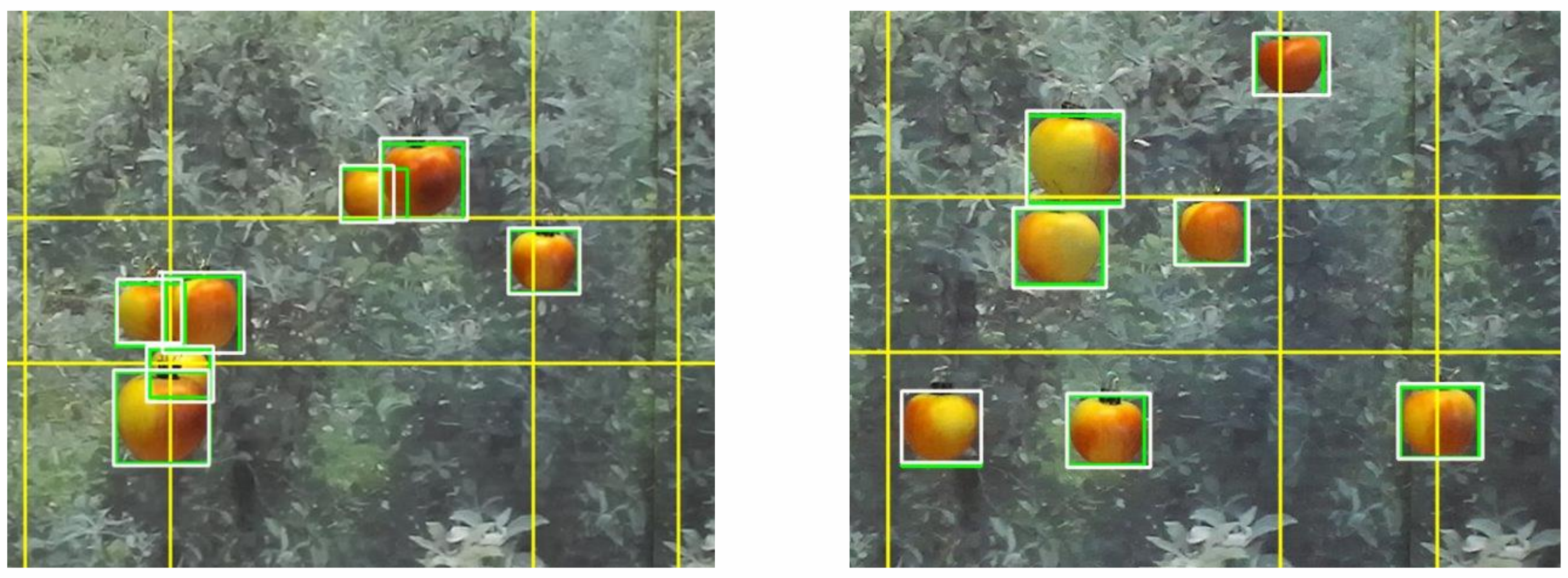

Figure 3. Detection results of a model trained solely on simulated data, on images from the Mini-Orchards dataset. White boxes denote predictions, green boxes denote the manually annotated ground truth.

Conclusions

- Our simulated dataset generator can annotate 18 apples per second and generate over 2000 annotated RGB images per hour.
- To achieve good performance, the simulated dataset needs to be representative of the real data.
- On Mini-Orchards, YOLOv5m models achieve almost the same results when trained on simulated data, as when trained on real data.
- Results of smaller YOLOv5 networks can be improved by adding real data to the training set.
- Simulated data could be very useful for a wide range of small data deep-learning projects.

Acknowledgements

- SIA - This project is financially supported by Regieorgaan SIA (part of NWO) and performed within the RAAK PRO project Mars4Earth.
- We would like to thank our collaborators at Saxion University of Applied Sciences for insightful discussions.

NHL STENDEN computer vision & data science

Regieorgaan SIA

SAXION University of Applied Sciences

References

[1] Unity Technologies. Unity real-time development platform — 3d, 2d vr & ar engine, 2022.
[2] G. Jocher. Ultralytics (yolov5). [Online]. Available: https://github.com/ultralytics/yolov5