Object Detection Models
For Detecting Apples in Orchards

Dylan Hiemstra - Bachelor Software Engineering
Wouter Welling & Dennis de Graaf - Bachelor Electrical Engineering
Supervisors: Willem Dijkstra and Jaap van de Loosdrecht

Winter 2021

Introduction

- Apples in orchards can be hard to detect due to random variables such as obstruction by leaves.
- Training, validation and testing is done using a dataset for apple detection.

Materials and Methods

Models: Faster R-CNN [1], EfficientDet [2] with tiling, YOLOv5 [3]
Metrics: precision, recall, F1-score, mean average precision and inference time
Datasets:

- MinneApple[4]
  - 1000 images, each 720x1280 pixels
  - Collected using video footage from a Samsung Galaxy S4 cell phone
  - Every 5th frame of the video sequence was taken and human-annotated
- RIWO dataset
  - 32 images, different resolutions like 720x1280, 720x960 and 720x720
  - Provided by our client
  - Relatively small dataset. Only used for testing.

Abstract

To build a system that helps reduce the working hours of human apple pickers, an object detection model for detecting apples is essential. In this research, we create a benchmark of three different object detection models. The best model we found is YOLO v5 with a F1 score of 71%.

Figure 1. An example image from the RIWO dataset.

Experiments and Results

In Figure 2, three cutouts of an image from the MinneApple dataset is shown. These cutouts show the detections (white boxes) of each network. The green boxes are the ground truth. In this case, YOLOv5 L performs better han EfficientDet D0 and Faster R-CNN.

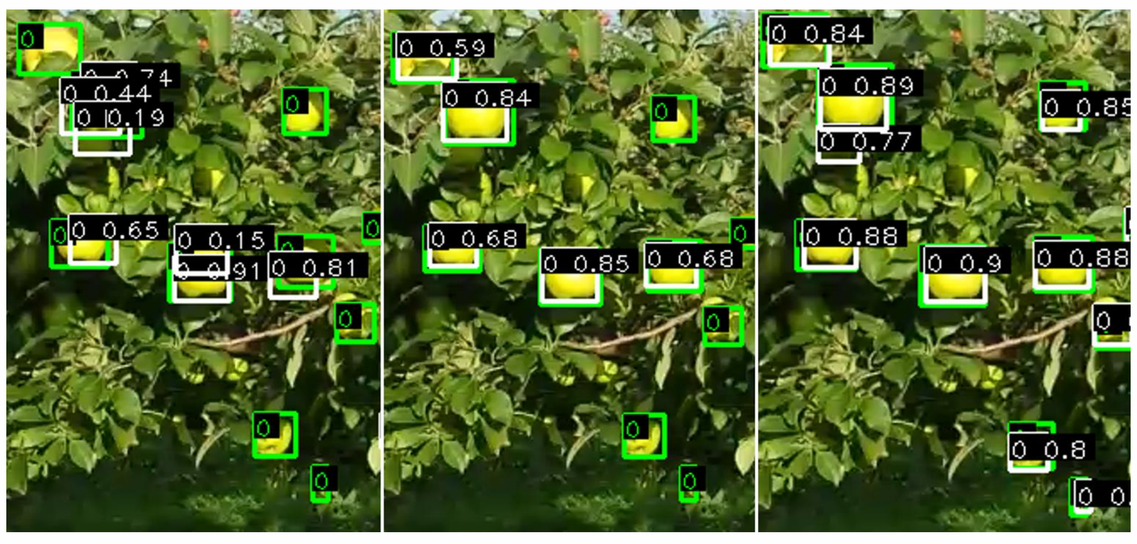

Figure 2. From left to right: Faster R-CNN, EfficientDet D0 and YOLO v5 L

Experiments and Results

The following experiments have been conducted:
A) Baseline experiment, no augmentations are applied and vanilla models are used.
B) Augmentations experiment, realistic augmentations are applied. See Figure 3.
B2) Built-in YOLOv5 augmentations from YOLOv5 implementations of Ultralytics[3].
C) To determine how robust a model is, select best performing models on MinneApple dataset then apply it on the RIWO dataset.

Original image Hue change Horizontal flip

Figure 3. A couple examples of our chosen augmentations we used for experiment B.

Experiments and Results

To summarize the results, we chose to only pick the results of the best model for each experiment in Table 1.

Experiments	Best model	F1-score	mAP	Inference
Experiment A	EfficientDet D3	0,7	0,65	194ms
Experiment B	YOLOv5 X	0,71	0,6	39ms
Experiment B2	YOLOv5 L (built-in augmentations)	0,8	0,71	21ms
Experiment C	YOLOv5 L (built-in augmentations)	0,45	0,44	25ms

Table 1. Results of the best performing model for each experiment. Note that YOLOv5 L with built-in augmentations is not part of our benchmark because those models have a unfair advantage of different augmentations.

One of the phenomenons, shown in Table 1, is that the inference time of EfficientDet D0 is much higher than to other inference times. This is happening because we are using tiling for EfficientDet and stitching back the tiles to one image is also taken into account when calculating the inference time. We use tiling for EfficientDet because EfficientDet has fixed input sizes. If we would feed an entire image into it, the apples would not be recognizable. Although we did not check that claim due to time constraints.

Conclusions

- Tiling images adds overhead. It increases the inference time.
- Our chosen augmentations did not affect the performance that much. The reason behind this is because the dataset already includes those variations.
- The built-in augmentations from the YOLOv5 implementation from Ultralytics[3] improves the performance. The average F1-score for YOLOv5 L with our chosen augmentation is 0.66, while the F1-score of YOLO v5 L with the built-in augmenations is 0.8.
- The potential best-suited models are YOLO v5 X(F1-score of 0.71) and EfficientDet D3(F1-score of 0.7). .

Acknowledgements

- This project is financially supported by SIA RAAK-mkb under the Focus op Vision project.
- The work is supported by NHL-Stenden Computer Vision Data Science and Hogeschool Saxion

NHL STENDEN computer vision & data science
SAXION HOGESCHOOL
SIA Regieorgaan

References

- **[1]** Y. Chen. A simple and fast implementation of faster r-cnn. [Online]. Available: https://github.com/chenyuntc/simple-faster-rcnn-pytorch
- **[2]** rwightman. Efficientdet (pytorch). [Online]. Available: https://github.com/rwightman/efficientdet-pytorch
- **[3]** G. Jocher. Ultralytics (yolov5). [Online]. Available: https://github.com/ultralytics/yolov5
- **[4]** P. R. N. H¨ani and V. Isler, “Minneapple: A benchmark dataset for apple detection and segmentation.” University of Minnesota, 2020.