On the scalability of CNNs for apple detection using RGBD data

Koen Molenaar - Bachelor Software Engineering

Supervisors: Maya Aghaei and Willem Dijkstra

Winter 2022

Introduction

- Picking apples can be automated using AI
- We can use CNNs for localizing these objects in an image
- The importance of portable devices becomes present in these tasks
- These devices have insufficiently been researched on the usability for these tasks

Abstract

Picking apples is a highly repetitive task that can be automated using AI. In these tasks, the importance of portable devices become present. An example is the Jetson Nano from Nvidia which is a portable device specifically designed to run neural networks. When testing models trained on this dataset, the inference time of the best model was 159ms whereas on VM-CVDS It was 7ms. Due to this large difference, the usability of the Jetson Nano relies on the maximum inference time allowed to execute the task effectively.

Materials and Methods

Models: EfficientDet [1] and YOLOv5 [2]
Metrics: precision, recall, F1-score, mean average precision and inference time
Dataset:

- KFuji RGB-DS database [3]
  - 967 images, each 373x548 pixels
  - Includes RGB, Depth and IR intensity data
  - Data collected using Microsoft Kinect V2
  - All 12.839 have been manually annotated

Computer Name	GPU	GPU RAM	RAM	CPU
VM-CVDS	GeForce RTX 2070	8 GB	16 GB	Intel i9-9760X
Jetson Nano	128-core Maxwell	-	4 GB	Quad Core A57

Table 1. Hardware specifications

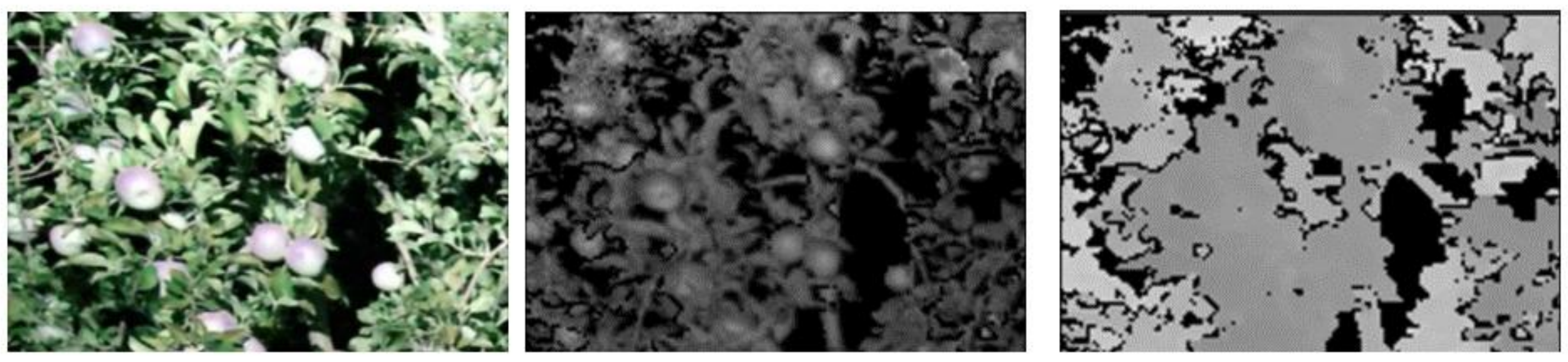

Figure 1. RGB, depth, and IR intensity visualization respectively

Experiments and Results

In figure 2, the predictions of both EfficientDet-D0 and YOLOv5 models are shown of the same image.

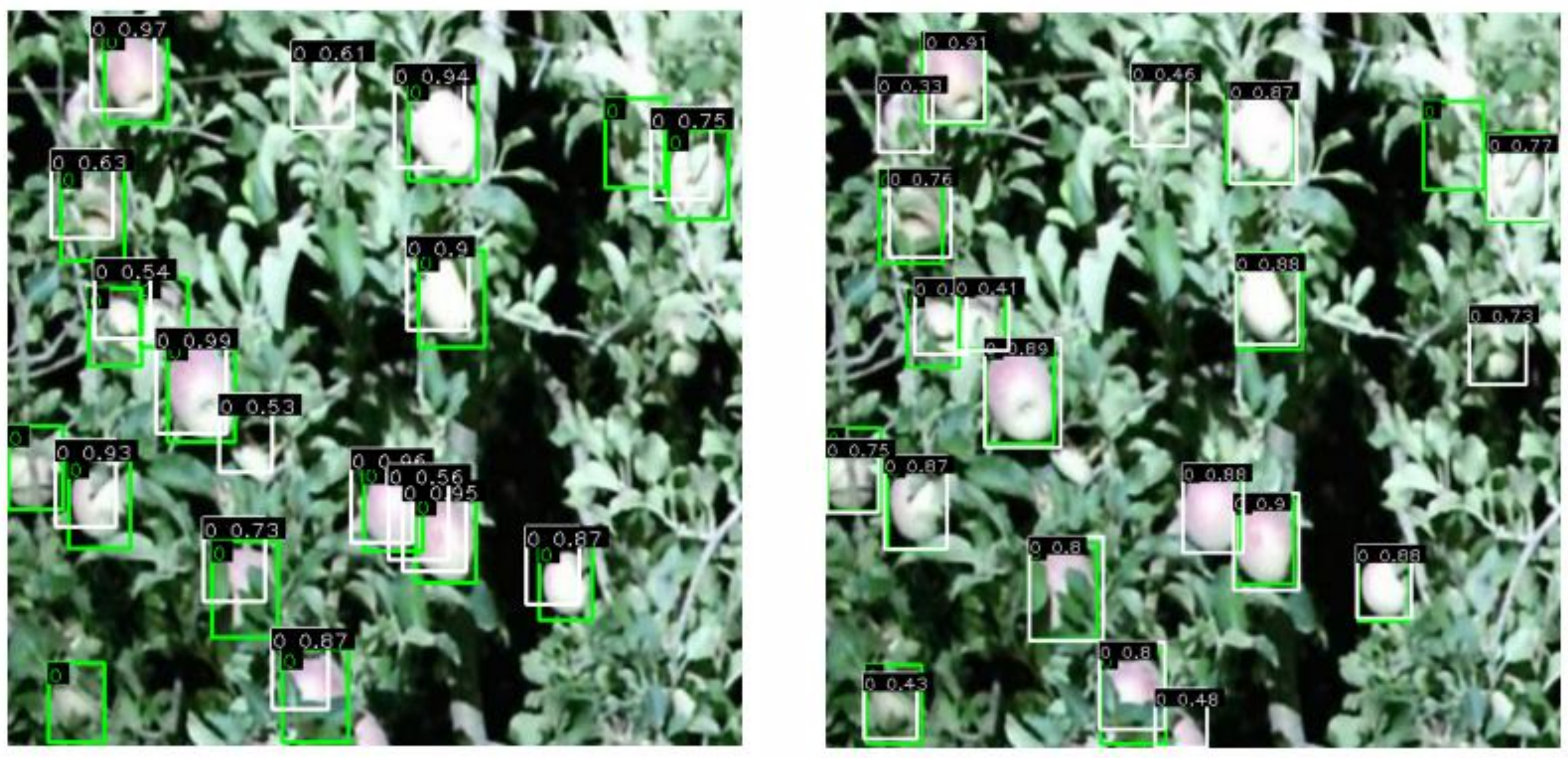

Figure 2. From left to right: EfficientDet-D0 and YOLOv5s

Acknowledgements

- This project is financially supported by Regieorgaan SIA (part of NWO) under the RAAK PRO projects Mars4Earth and Focus op Vision

Experiments and Results

The following experiments have been conducted:
A) Baseline experiment, models trained only on RGB data
B) Augmentations experiment, realistic augmentations are applied. See Figure 3.
C) RGBD experiment, models trained only RGB and depth data
D) Performance testing on the Jetson Nano and VM-CVDS

Original image　　Brightness 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 2.

Experiments	Best model	F1-score	mAP	Inf. VM-CVDS	Inf. JN (Exp. D)
Experiment A	YOLOv5s	0.853	0.796	7ms	159ms
	EfficientDet-D0	0.836	0.713	24ms	298ms
Experiment B	YOLOv5s	0,846	0,795	8ms	159ms
	EfficientDet-D2	0.836	0.695	60ms	603ms
Experiment C	YOLOv5s	0,867	0,801	8ms	163ms
	EfficientDet-D1	0.856	0.8	27ms	317ms

Table 2. Results of the best performing model for each experiment. Note that the differences in F1-scores are very small

In Table 1 we can see that the YOLOv5 models take considerably less time than the EfficientDet models perform inference. We also see that augmentations (Experiment B) practically do not change the F1-scores in a significant way. Using the additional depth data however causes an increase in the F1-scores consistently. When testing the performance on VM-CVDS and Jetson Nano, YOLOv5 takes half the time of the best EfficientDet models while giving better predictions. The difference between inference times on the Jetson Nano is often 15x larger than on the VM-CVDS.

Conclusions

- YOLOv5 models gives better predictions than EfficientDet consistently
- Our chosen augmentations did not affect the performance much. The reason is that these variations were already apparent in the training data.
- Using depth data on top of RGB data consistently gives better predictions.
- The inference time on the Jetson Nano is often 15x larger than on VM-CVDS a high-end, non-portable machine.

References

- **[1]** rwightman. Efficientdet (pytorch). [Online]. Available: https://github.com/rwightman/efficientdet-pytorch
- **[2]** G. Jocher. Ultralytics (yolov5). [Online]. Available: https://github.com/ultralytics/yolov5
- **[3]** J. Gené-Mola and V. Vilaplana – Kfuji RGB-DS Database: Fuji apple multi-modal images for fruit detection with color, depth, and range-corrected IR data.

NHL STENDEN computer vision & data science

SIA Regieorgaan