Detection Methods	Metrics					Platform			Implementation	Reference
	F1 score	mIoU	Accuracy	AP/mAP	PRC	File [MB]	Inference [s]	Characteristics		
Mask R-CNN + FPN (ResNet101)	X	0.575	X	0.739 (IoU>0.5), 0.706 (IoU>0.7)	√	256	368.6	X	nearthlab/image-segmentation	1
Mask R-CNN + FPN (ResNet50)	X	0.595	X	0.759 (IoU>0.5), 0.724 (IoU>0.7)	√	179.2	327.1	"Mobile implementation"	nearthlab/image-segmentation	1
Mask R-CNN + FPN (MobileNet)	X	0.567	X	0.693 (IoU>0.5), 0.591 (IoU>0.7)	√	95.6	211.2	"Mobile implementation"	nearthlab/image-segmentation	1
YOLOv4 (ResNet, InceptionV3, DenseNet)	0.8	X	0.9	0,759 (IoU=0.5), 0.270 (IoU=0.75)	√	X	X	X	github.com/pythonlessons/TensorFlow-	3
YOLOv3	0.579	X	X	0.394 (IoU=0.5), 0.125 (IoU=0.7)	X	X	X	X	github.com/pythonlessons/TensorFlow-	4
YOLOv2	0.652	X	X	0.478 (IoU=0.5), 0.139 (IoU=0.7)	X	X	X	X	github.com/experiencor/	4
YOLOv3-SPP	0.64	X	X	0.730 (IoU=0.5)	X	X	>= 33.30 ms	4 different HW	github.com/AlbertoSabater/Keras-YOLO-	6
YOLOv3-Tiny (Darknet-50)	X	X	0.96	X	X	72.5	120 ms	Nvidia Jetson nano	github.com/pythonlessons/TensorFlow-	7
SSD (MobileNet v2)	X	X	0.82	X	X	54	180 ms	Nvidia Jetson nano	github.com/ManishSoni1908/Mobilenet-	7
Faster RCNN (ResNet-50)	X	X	0.94	X	X	76	230 ms	Nvidia Jetson nano	CNN.ipynb,https://	7

	TX2	Nano	RP 4	RP 3B+	RP 3A+	A. Uno	A. Mega
Ease of Use	Disk, New	Power, New	Installation	Installation	Installation	Intuitive	Intuitive
Functionality	Ubuntu	Ubuntu	Raspian	Raspian	Raspian	.ino	.ino
Cost	$400	$99	$35-55	$35	$25	$18	$14
RAM	8GB	4 GB	1-4 GB	1 GB	.5 GB	32 kB	256 kB
CPU	2GHz	1.43GHz	1.5GHz	1.4 GHz 4-core	1.4 GHz 4-core	16 MHz	16MHz
GPU	1.3GHz	921MHz	None	None	None	None	None
Wi-Fi	Yes	Yes	Yes	Yes	Yes	With shield	With shield
Disk Space	32GB	microSD	microSD	microSD	microSD	None	None

From https://www.linkedin.com/pulse/jetson-vs-raspberry-pi-arduino-robotics-nathan-george/ (2019)

References Legend

1 Blueberries quantification through instance segmentation, deepblueberry dataset. Data Augmentation. FPN for small objects.

2 Similar to previous reference. 2020. Mask R-CNN + FPN (ResNet101).

3 Segmentation of grape bunches. All training and validation of the object detection and classification have been performed on a Tensorbook laptop with Intel Core i7-11800H, NVIDIA® RTX™ 3080 Max-Q GPU, and 64 GB of RAM.

4 Mask R-CNN (He et al., 2017) a convolutional framework for instance segmentation vs YOLO (Redmon et al., 2016), a single-stage network that can detect objects without a previous region-proposal Stage. 2020. Training was performed on a computer containing a single NVIDIA TITAN Xp GPU (12 GB memory), an Intel i7-x990 CPU and 48 GB RAM, and running Ubuntu 18.04 LTS. For Mask R-CNN, training was performed in approximately 10 h (100 epochs). YOLO training (v2 and v3) spent four days.

5 Mask R-CNN + FPN. Mask R-CNN over bounding box detection models, such as Faster R-CNN, and RetinaNet. RetinaNet is a single-stage detector applied on densely sampling candidate objects. Also CenterMask (VoVNet-**).

6 Debris detection system for blueberries, YOLOv3 vs YOLOv3-SPP. Data Augmentation. Hardware tests.

7 Pitaya Harvesting. SLAM. Data Augmentation. NN comparison, YOLO looks better then Mask-RCNN for embeddability. Not bad also SSD and Faster RCNN. Jetson emulator available (https://pypi.org/project/jetson-emulator/). Will it be good for small objects like berries?

8 Object detection methods, a comparison. RCNN, Fast RCNN and YOLO (2020)