Beschikbare hardware:

Nvidia Jetson Nano

Nvidia Jetson Xavier NX Development Kit

Raspberry Pi Camera

Coral Development Board & Accelerator

4k netwerkcamera

Verslag gebruik:

Coral bordje is underpowered, maar hebben we wel gebruikt om de dispenser te detecteren aan de hand van QR-codes, logos en tekstrecognition. Deze hebben we hiervoor uiteindelijk niet kunnen gebruiken, omdat we overgingen naar een statische detectiezone voor het detecteren van ontsmetting. Alle onderstaande pogingen tot computer vision zijn op de Jetson Xavier.

Jetpack is de SDK die draait op de Nvidia Jetsons. Omdat de Xavier redelijk nieuw is draaien er enkel versies 4.3, 4.4 en 4.5 op (allemaal gebaseerd op Ubuntu 18.04). Op deze versies draait Cuda versie 10.2. Cuda is vereist om computer vision performant te maken op de Jetsons. Tensorflow, Cuda en CuDNN (nog een deep learning library die vereist is op de Jetsons voor de meeste projecten) hebben elkaar niet graag. Ze zijn doorgaans incompatibel en zeer versie-afhankelijk. Er bestaan tabellen waar ze de compatibiliteit oplijsten. (<https://www.tensorflow.org/install/source#gpu>) Merk op dat in deze tabel beschreven staat dat Cuda 10.2 niet compatibel is met Tensorflow. Maar een installatie op de Nvidia forums lost dit (grotendeels) op. (zie volgende alinea)

Tensorflow wordt veel gebruikt in computer vision, het is een framework voor machine learning. Nvidia heeft hier ook zijn eigen sdk van die TensorRT noemt. Dit is veel meer niche en wordt vaak zelfs in combinatie met Tensorflow gebruikt. Sterker nog, meestal is het nodig om op de Jetsons gebruik van beide te maken om het performant genoeg te krijgen. Tensorflow is een programma dat niet standaard voor de ARM processor architectuur gemaakt is. Op forums van Nvidia bestaat er een guide over hoe men Tensorflow installeert voor een specifieke Jetson en Jetpack versie. (<https://forums.developer.nvidia.com/t/official-tensorflow-for-jetson-agx-xaviernx/141306> -> Let op, deze installatie duurt >2u)

Voor image processing wordt (vaak) OpenCV gebruikt dus besloten wij dit ook te doen. Bij het installeren moet er rekening mee gehouden worden dat OpenCV de dependency Numpy installeert. Numpy wordt ook gebruikt als dependency bij Tensorflow dus moet je zorgen dat je een OpenCV versie installeert dat gebruik maakt van een Numpy versie dat werkt met de geïnstalleerde Tensorflow versie. Voor ons was dat OpenCV versie 3.4.10.37, Numpy versie 1.16 en Tensorflow 1.15. (Tensorflow 2.x bestaat ook, maar wordt nog niet zo widespread gebruikt als versie 1.x)

Voor de Jetson Xavier hebben we volgende projecten opgezet:

<https://github.com/dusty-nv/jetson-inference> -> Deze werkte. Niet accuraat of performant genoeg. Weinig uitbreidingsmogelijkheden.

<https://github.com/pythonlessons/TensorFlow-2.x-YOLOv3> -> Deze hebben we eerst getest op een desktop. Performantie en nauwkeurigheid was veelbelovend. Toen we het eindelijk draaiende kregen op de Jetson bleek er te weinig geheugen om het out of the box te draaien. De modellen moesten omgezet worden naar TensorRT modellen waardoor er zeer veel nauwkeurigheid verloren ging en de performantie was ook niet naar behoren. (1-3 frames per seconde)

<https://github.com/ahmetozlu/tensorflow_object_counting_api> -> Werkte niet op de Jetsons.

Voor de rest nog veel projecten uitgeprobeerd maar de meesten zijn niet accuraat of performant genoeg of er zijn problemen met compatibiliteit.