

```
> $ git clone https://github.com/google-coral/tflite.git
```

En ga dan naar de juiste map en installeer de afhankelijkheden:

```
> $ cd tflite/python/examples/classification
> $ ./install_requirements.sh
```

Er missen nog enkele afhankelijkheden in het installatiescript:

```
> $ sudo apt install libopenjp2-7-dev libtiff5
```

Het project in deze map herkent vogelsoorten in foto's. Het installatiescript downloadt een model dat op vogels getraind is, samen met een bestand dat de bijbehorende namen van de herkende vogels bevat. Het script downloadt ook een voorbeeldfoto van een geelvleugelara, zodat je het onmiddellijk kunt uitproberen.

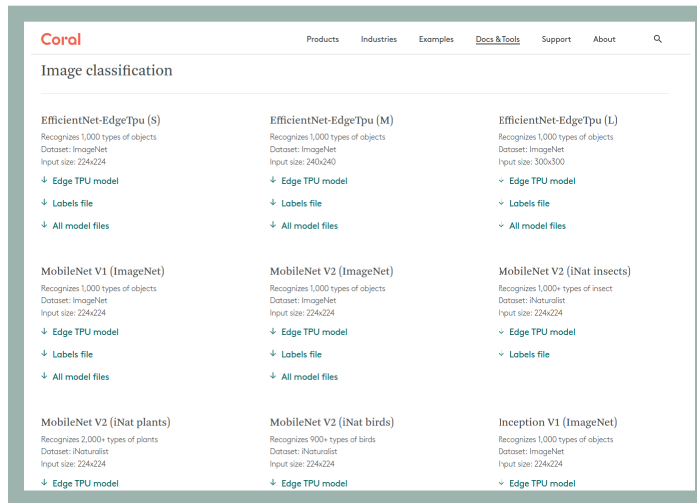
Je voert het classificatieprogramma als volgt uit:

```
> $ python3 classify_image.py
--model models/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite --labels models/inat_bird_labels.txt --input images/parrot.jpg
----INFERENCE TIME----
Note: The first inference on Edge TPU is slow because it includes loading the model into Edge TPU memory.
17.5ms
4.4ms
4.4ms
4.4ms
4.4ms
-----RESULTS-----
Ara macao (Scarlet Macaw):0.77734
```

Je ziet dat we hier het model, de labels en de invoer doorgeven aan het Python-programma. De uitvoer toont dat het inderdaad om een geelvleugelara gaat, of toch met een kleine 78% waarschijnlijkheid.

Probeer dit nu eens uit met enkele andere foto's die je van internet haalt. De resultaten zijn meestal heel goed.

> Zodra het script draait, luistert het continu naar de audiostroom van je microfoon <



Op de website van Google Coral vind je allerlei modellen voor de Edge TPU.

SNELHEIDS-VERBETERING

Maar welke snelheidsverbetering haal je nu door gebruik te maken van de Coral USB Accelerator? Dat kun je eenvoudig uittesten. Het model dat je hiervoor gebruikt hebt (`mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite`), is gecompileerd voor de Edge TPU. Geef je in de plaats het model `mobilenet_v2_1.0_224_inat_bird_quant.tflite` door, dat gecompileerd is voor de CPU, dan voert TensorFlow de bewerkingen puur op de processor van de Raspberry Pi uit. Laten we dat eens bekijken:

```
> $ python3 classify_image.py
--model models/mobilenet_v2_1.0_224_inat_bird_quant.tflite
--labels models/inat_bird_labels.txt --input images/parrot.jpg
----INFERENCE TIME----
Note: The first inference on Edge TPU is slow because it includes loading the model into Edge TPU memory.
117.3ms
114.4ms
114.7ms
113.9ms
113.8ms
-----RESULTS-----
Ara macao (Scarlet Macaw):
0.77734
```

Op de Raspberry Pi 4 waarop we deze code uitprobeerden, voert de coral USB Accelerator het model dus 27 keer sneller uit. Naar de eerste tijdsduur moet je niet kijken, want zoals de opmerking in de uitvoer al zegt is die trager omdat de Edge TPU dan het model nog in zijn geheugen moet inladen. Daarom dat het

het herkende woord in het geel. Er worden zelfs een top drie van herkende woorden getoond als je iets zegt, elk met hun waarschijnlijkheid.

MODELLEN

Google heeft op één webpagina (<https://coral.ai/models/>) al zijn modellen voor de Edge TPU geplaatst, waar je ze eenvoudig kunt downloaden. Vooral voor beeldclassificatie zijn er veel modellen beschikbaar. Zo zijn er drie versies van EfficientNet, waarmee je 1000 types objecten kunt herkennen in kleine, gemiddelde of grotere afbeeldingen. Ook van MobileNet zijn er diverse varianten: om 1000 objecten te herkennen, 1000 insecten, 2000 planten en 900 soorten vogels. Ook van Inception, dat 1000 types objecten herkent, zijn er vier versies beschikbaar. Wel model voor jouw toepassing geschikt is, hangt af van de beelden die je wilt verwerken. We raden aan om verschillende modellen te testen.

Ook voor objectherkenning zijn er meerdere modellen. Objectherkenning doet meer dan beeldclassificatie: bij die laatste krijg je gewoon een label "vogel" als er een vogel in het beeld staat. Een model voor objectherkenning vertelt je niet alleen dat er een vogel te zien is, maar toont je ook de locatie. Dat kun je dan visualiseren met een kader rond het object. Er zijn twee versies van MobileNet SSD die de locatie van 90 types objecten kunnen herkennen, en één versie die de locatie van een menselijk gezicht detecteert.

Als je wat verder zoekt, vind je ook elders nog wel andere modellen. Maar je moet er altijd zeker van zijn dat ze voor de Edge TPU zijn gecompileerd.

HERTRAIN EEN MODEL

Google heeft nog zo enkele voorbeeldprogramma's op hun website staan, maar allemaal maken ze gebruik van een bestaand model. Wat als je andere klassen van objecten wilt herkennen in afbeeldingen? Gelukkig hoeft je dan niet onmiddellijk zelf een volledig nieuw model te trainen. Er bestaat een techniek transfer learning die een bestaand model hertraint om nieuwe klassen te herkennen. En het resulterende model kun je gewoon op de Edge TPU draaien.

Eén manier is om de laatste laag van het neurale netwerk te hertrainen met backpropagation. Dat

programma standaard het algoritme vijf keer uitvoert.

Op een Raspberry Pi 3B voert dit script overigens zijn werk uit in 11,8 ms omdat de Coral USB Accelerator dan beperkt wordt door de lagere snelheid van USB 2.0. Zonder het versnellerbordje heeft de processor 269 ms nodig.

WOORDEN HERKENNEN

Een ander voorbeeldproject van Google laat je toe om enkele sleutelwoorden te herkennen. Dat vereist dat je een werkende microfoon op je Raspberry Pi aangesloten hebt. Download de code en installeer de afhankelijkheden:

```
> $ git clone https://github.com/google-coral/project-keyword-spotter.git
> $ cd project-keyword-spotter
> $ sh install_requirements.sh
```

Voor daarna het herkenningsscript uit:

```
> $ python3 run_model.py
```

Als het afbreekt met een foutmelding over een timeout, dan heeft het script waarschijnlijk de verkeerde standaardmicrofoon gekozen. Voer het dan nog eens uit met de optie `-mic ID`, waarbij je ID vervangt door het ID dat het script bij de bijbehorende geluidskaart toont.

Zodra het script draait, luistert het continu naar de audiostroom van je microfoon. In het bestand `config/labels_gc2.raw.txt` staat de lijst van sleutelwoorden die het model herkent, zoals `yes`, `no`, `mute`, `volume up`, `volume down`, `next song`, `start video`, `channel eleven` enzovoort.

Spreek maar eens een van deze woorden uit, dan zie je in de uitvoer