

BEELD- VERWERKING

met de Nvidia Jetson Nano (Deel 2)



Figuur 1: De Jetson
Nano Developer Kit.
(Bron: developer.nvidia.com)

Beeldherkenning met Edge Impulse

Mathias Claußen (Elektor Lab)

Als u op zoek bent naar een eenvoudige opzet om aan de slag te gaan in de wereld van neurale netwerken, kijk dan eens naar Edge Impulse. Een grote verscheidenheid aan ontwikkelborden en -apparaten kan worden gebruikt om bijvoorbeeld een dataset van afbeeldingen vast te leggen en op te bouwen, waarmee in de cloud neurale netwerken kunnen worden gemaakt. Deze kunt u dan toepassen in uw eigen applicaties. Wij hebben het eens geprobeerd met Edge Impulse, een Jetson Nano-ontwikkelbord en wat speelkaarten.

Na de introductie van de Jetson Nano (**figuur 1**) en SparkFun JetBot AI Kit (**figuur 2**) in het eerste deel van deze serie [1], is het nu tijd om onze eerste stappen te zetten met beeldherkenning. Om een afbeelding te kunnen herkennen met de Jetson Nano, zal er software en een neurale netwerk getraind moeten worden. Zoals reeds beschreven in het online artikel "Koffie zetten met de MAX78000 en wat AI" [2], is het trainen van het netwerk geen gemakkelijke onderneming. Voor de console-toepassing in de eerste fase wordt PyTorch of TensorFlow gebruikt, waarmee u zich vertrouwd zult moeten maken. Het maken van een neurale netwerk op uw eigen computer zal ook enige tijd in beslag nemen, afhankelijk van de grafische kaart en CPU van de machine.

De uitdagingen houden daar niet op: het neurale netwerk moet eerst worden getraind met behulp van een dataset. Er zijn ook extra toepassingen nodig om de dataset op te bouwen en klaar te maken voor training. Het kan dus enige tijd duren voordat we ons eerste neurale netwerk getraind krijgen.

Edge Impulse

Als u op zoek bent naar een vrij eenvoudige introductie in de wereld van neurale netwerken, moet u eens kijken naar de mogelijkheden van Edge Impulse [3]. Dit is een web-gebaseerd platform dat uw datasets kan gebruiken om neurale netwerken te bouwen en ze te porten naar een grote verscheidenheid van apparaten. Het bereik strekt zich uit van Arduino tot een x86/AMD64-gebaseerde laptop. In dit

artikel maken we een neurale netwerk in Edge Impulse en gebruiken we dit met de JetBot om een paar objecten te herkennen.

De gegevens kunnen worden verzameld met verschillende apparaten, zoals een PC met een webbrowser of een smartphone. De dataset wordt rechtstreeks naar de Edge Impulse servers gestuurd en daar opgeslagen. Als u al over een dataset beschikt, kunt u deze eenvoudig uploaden via de webinterface. Het sorteren en labelen van de dataset vindt vervolgens plaats via de webinterface. Ook de parameters voor het creëren van het neurale netwerk worden daar gedefinieerd en geconfigureerd.

Het maken van een neurale netwerk vraagt veel rekenkracht. De gegevens worden daarom geüpload naar de Edge Impulse server en daar omgezet in een neurale netwerk. De training vindt dus plaats in de cloud en vormt geen belasting voor uw eigen computer. Het neurale netwerk kan daarna worden omgezet in een programmabibliotheek, die u kunt downloaden en gebruiken in eigen applicaties.

Hier zullen we al deze stappen behandelen door een kleine toepassing voor beeldherkenning te bouwen. In ons zusterblad *Elektor Industry* hebben we een artikel gepubliceerd waarin het gebruik van Edge Impulse voor ruisdetectie is behandeld [4].

Deel de kaarten

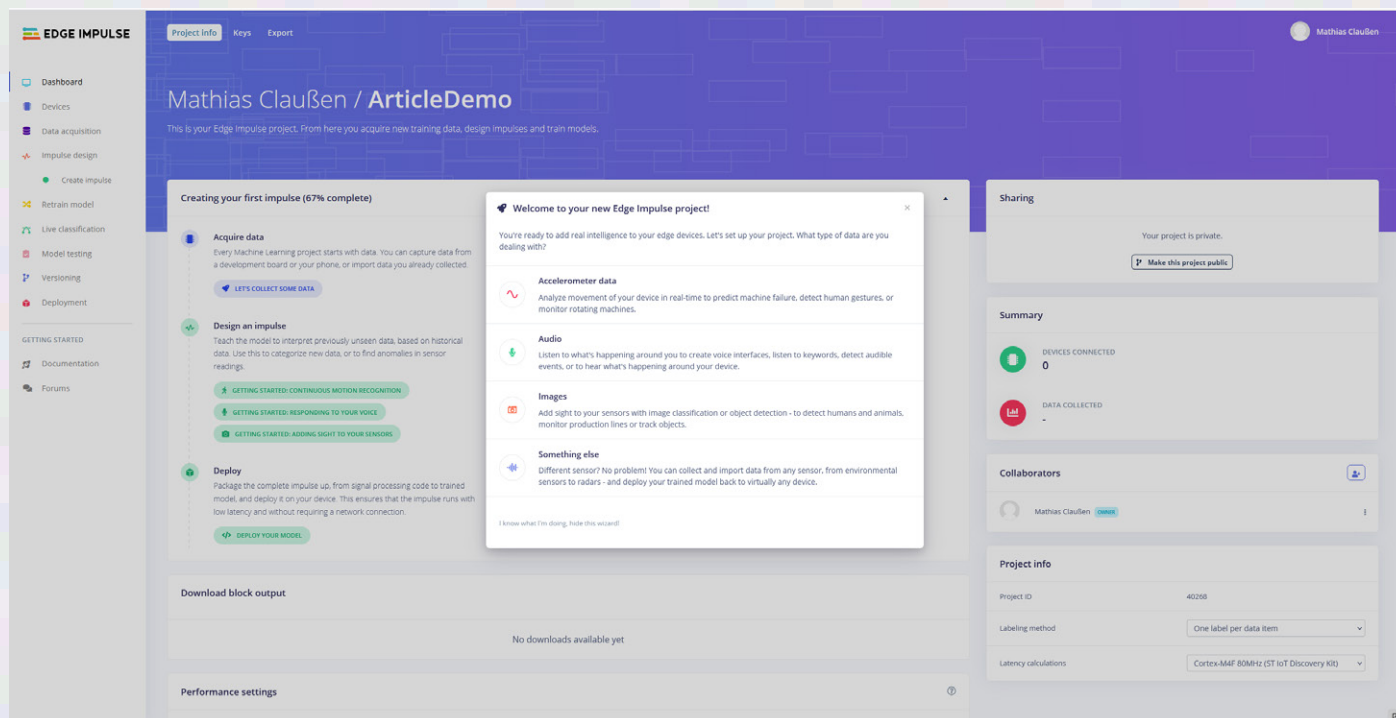
Om te beginnen is het een goed idee om als testobjecten voorwerpen te gebruiken



Figuur 2: De SparkFun JetBot AI Kit. (Bron: SparkFun)



Figuur 3: UNO-kaarten.



Figuur 4: Edge Impulse wizard.

die visueel sterk van elkaar verschillen. Wij hebben gekozen voor verschillende speelkaarten uit het kaartspel UNO van Mattel (figuur 3). Deze lagen toevallig voor het grijpen, maar het kan ook met andere voorwerpen zoals verkeersborden. Een Jetson Nano plus een SparkFun JetBot AI Kit (figuur 2), die een 8-megapixel camera bevat, vormen het hardwareplatform.

Ons eerste doel is om in staat te zijn het beeld van UNO-kaart nummer één te onderscheiden van kaart nummer twee en van de achterzijde van een kaart. Om dit mogelijk te maken zullen we, zoals reeds gezegd, het neurale netwerk moeten trainen, gebruikmakend van beelden van de te herkennen objecten. Hiervoor kunnen we eerst kijken naar de data-acquisitie en de interface van Edge Impulse.

Een Edge Impulse Project

We moeten ons eerst registreren om Edge Impulse te kunnen gebruiken. Voor dit project gebruiken we de gratis versie van het platform, dat bepaalde beperkingen heeft in het gebruik (waarover later meer). Na registratie kan een nieuw project worden aangemaakt; we worden dan begroet met een wizard (figuur 4).

Op de vraag wat voor soort gegevens we willen verwerken, antwoorden we met *Images*, omdat afbeeldingen van de UNO-kaarten het enige zijn waarin we geïnteresseerd zijn voor deze test. In dit geval moeten de UNO-kaarten

worden geclassificeerd (*Classify a single object*), omdat we in de eerste stap slechts een enkele kaart willen verwerken en bepalen welke kaart het is, zoals figuur 5 laat zien. Het project is nu geconfigureerd met de herkenningparameters. Wat nu komt, is het bouwen van de dataset waarmee we kunnen beginnen met trainen.

Data Acquisitie

In dit project zullen we de data voor het trainen van het neurale netwerk zelf genereren. Voor beeldherkenning zijn datasets beschikbaar met duizenden afbeeldingen die al voorgesorteerd zijn in bepaalde groepen; afhankelijk van uw toepassing kan het zinvol zijn zo'n dataset te gebruiken. We konden geen dataset van UNO-kaarten vinden, dus zullen we er een moeten genereren. Dit zal een goede oefening zijn om de stappen te doorlopen en te demonstreren hoe u zelf een dataset van afbeeldingen kunt maken. We zullen laten zien hoe sommige valkuilen kunnen worden vermeden.

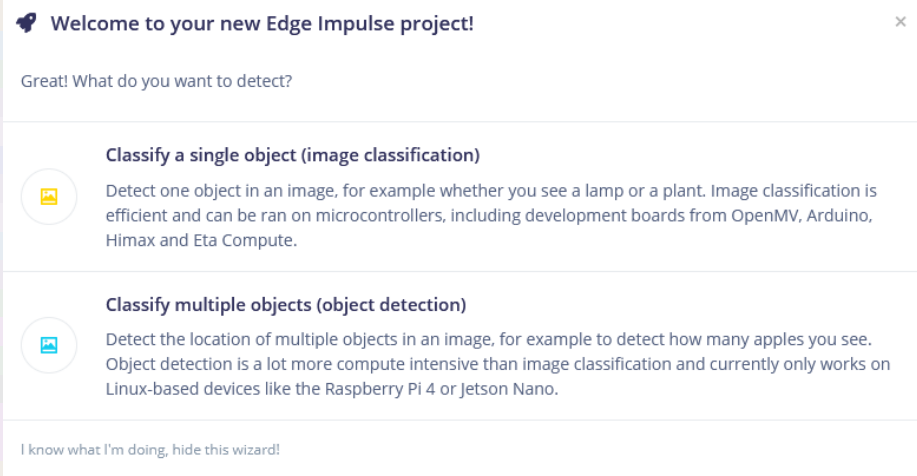
Om de beelden vast te leggen, is het ideaal als we dezelfde camera kunnen gebruiken die later voor de beeldherkenning zal worden gebruikt. Op die manier leert het neurale netwerk tegelijkertijd de eigenaardigheden van de camera kennen. Om de gegevens met de Nvidia Jetson te verzamelen, kunnen we een client voor Edge Impulse installeren. Deze kan dan de beelden (en eventueel audio) naar Edge Impulse sturen.

Om de client te installeren kunt u de instructies [5] volgen die door Edge Impulse zijn gegeven. Voor de Nvidia Jetson is het nodig om verbinding te maken met het bord door gebruik te maken van het SSH-protocol en commando's in te voeren via een console. Als alternatief kunt u ook een muis, toetsenbord en monitor aansluiten en de commandoregels direct invoeren in Linux, dat op de Jetson Nano draait.

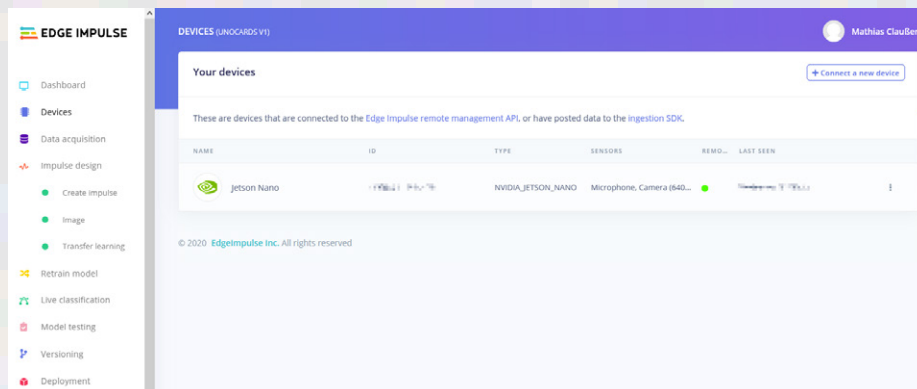
Wanneer een SSH-verbinding tot stand is gebracht of een console is geopend, hoeven de commando's uit de instructies alleen maar te worden ingevoerd. Een Internetverbinding is vereist voor de installatie. Als de Jetson Nano nog niet met het Internet verbonden is, moet dit eerst gedaan worden.

Dan typen we `wget -q -O - https://cdn.edgeimpulse.com/build-system/jetson.sh | bash` op de console om de installatie van *edge-impulse-linux te starten*. De installatie zal enige tijd duren. Nadat het voltooid is, geeft u `edge-impulse-linux -clean` in op de console. Het programma heeft onze toegangsinformatie nodig en het audio- en video-apparaat dat we zullen gebruiken voor data-acquisitie. Zodra alles is ingesteld, verschijnt de nieuw ingestelde Jetson Nano op de Edge Impulse pagina onder *Devices* (figuur 6). Hiermee kunnen we nu foto's maken met de camera.

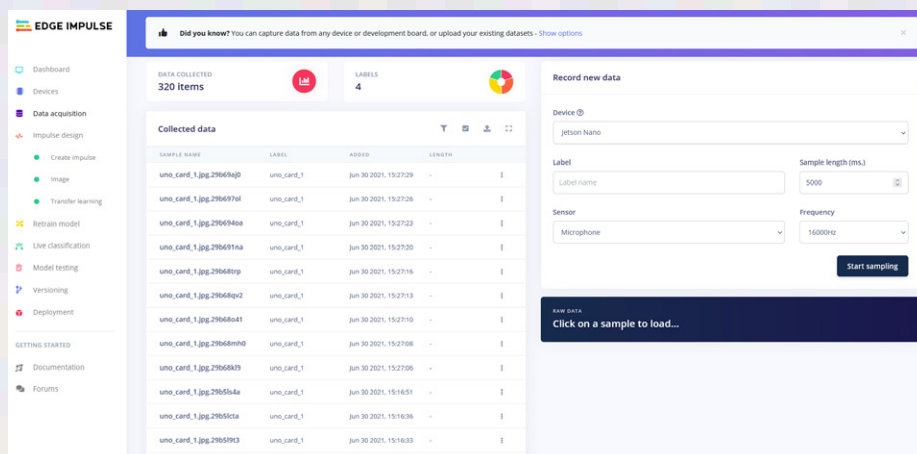
Op de Edge Impulse webpagina kunnen we nu beelden vastleggen via de camera met de menuoptie *Data acquisition* (figuur 7).



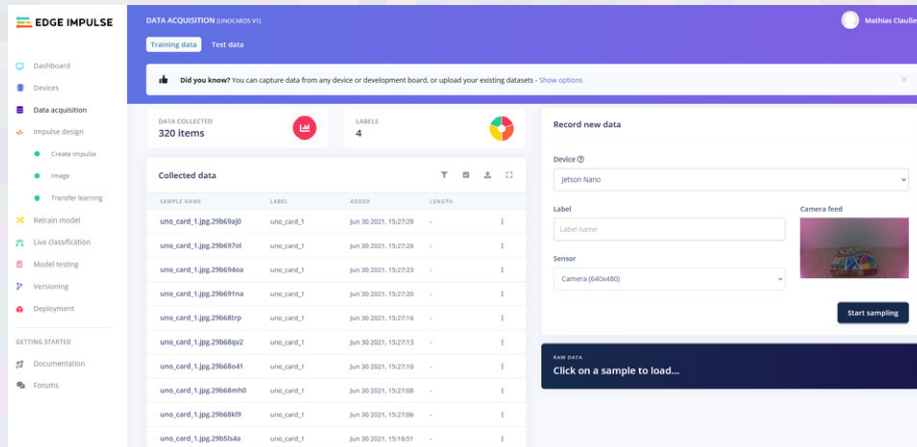
Figuur 5: Selecteer de object-classificatie.



Figuur 6: Stel JetBot in als het apparaat dat gegevens moet verzamelen.



Figuur 7: Data-acquisitie met Edge Impulse.



Figuur 8: Camera-opstelling.

Omdat hier ook een audiobron is geconfigureerd, moet eerst worden overgeschakeld naar *Camera (640x480)*. De *Camera feed* verschijnt nu met live beelden van de camera (**figuur 8**). Nu kunnen we met *Label* invoeren tot welk type beeld ("class") het behoort (geen kaart, kaart nummer één, kaart nummer twee of de achterzijde van de kaart).

Nu komt het saaie gedeelte - data acquisitie van de individuele kaarten in verschillende oriëntaties. In het begin moeten ongeveer 80 tot 100 beelden worden genomen voor elke kaart die we later willen herkennen. Het label dat bij de opgenomen beelden hoort, moet ook meteen worden toegekend.

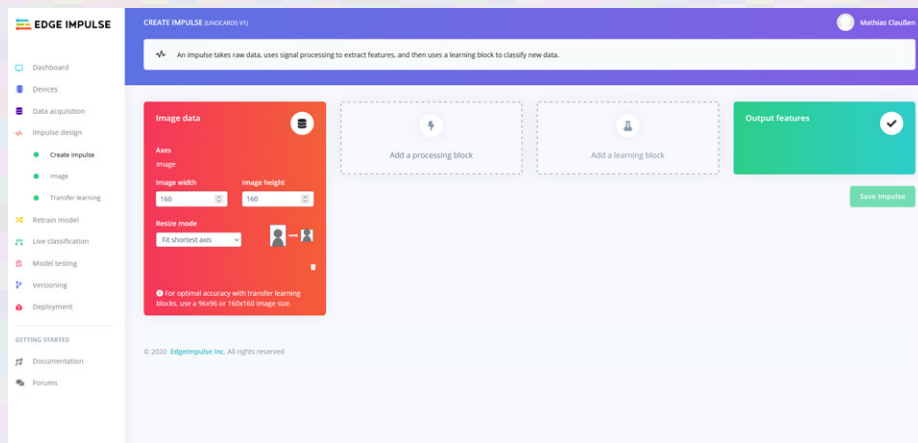
Initiële Training

Als alle beelden zijn genomen, is het tijd voor de eerste trainingssessie. Onder *Impuls design* (**figuur 9**) kunnen we nu de impuls definiëren (d.w.z. de operationele modus van het neurale netwerk). Voor de beeldgrootte gebruiken we 160 x 160 pixel en *Fit shortest axis* uit het *Resize mode* drop-down veld. Aangezien er geen kleuren zullen worden geëvalueerd, kunnen we *Grayscale* selecteren (**figuur 10**). Na het klikken op *Save parameters* kunnen de herkenningsklassen worden aangemaakt op de volgende pagina.

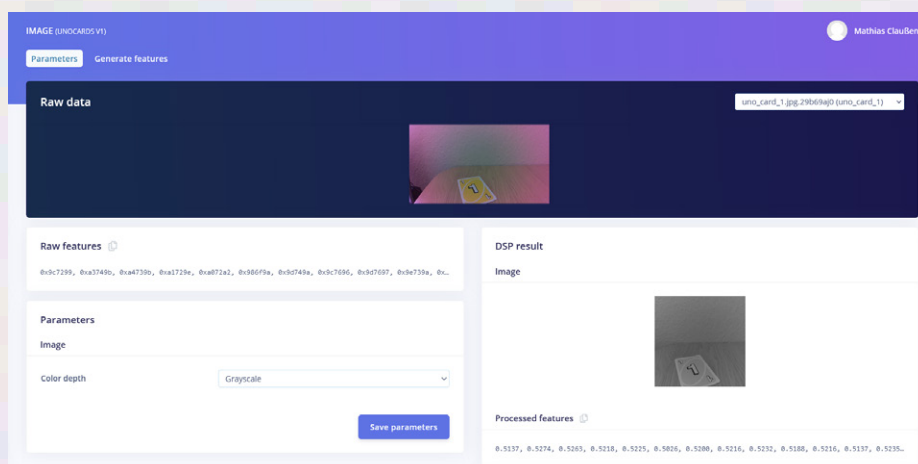
De puntenwolk in Feature explorer toont de verdeling van onze beelden op basis van een eerste ruwe training. Dit identificeert eventuele onjuist gelabelde gegevens. De onderlinge afstanden tussen de afzonderlijke punten (cluster) geeft een indicatie van de moeite die het neurale netwerk heeft om de kaarten van elkaar te onderscheiden (**figuur 11**).

De parameters voor de training kunnen nu worden ingesteld in *Transfer learning* (**figuur 12**). De beperkingen van de gratis versie werden in het begin al genoemd, we zullen ze nu bekijken. Er wordt door de Edge Impulse servers 20 minuten rekentijd uitgetrokken voor het trainingsproces. Voor onze vier klassen is dit voldoende om bruikbare resultaten te bereiken. Als het trainingsproces meer dan 20 minuten nodig heeft, wordt het gewoon afgebroken.

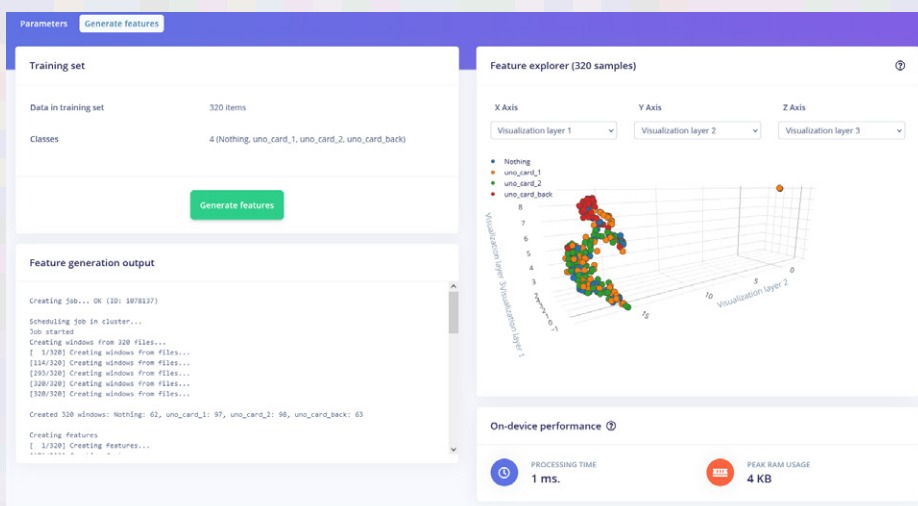
Het aantal trainingscycli beïnvloedt de kwaliteit van de training, evenals de tijd. In dit geval is de waarde ingesteld op 20 voor gegevenstraining; verhoging van het aantal cycli van 20 tot 50 verbetert het resultaat. Aangezien de hier gebruikte dataset zeer



Figuur 9: Maak een nieuwe Impulse.



Figuur 10: Werken met grijstinten.



Figuur 11: Neuraal Netwerk, kenmerken genereren.

hanteerbaar is, kan *data augmentation* worden gebruikt om de herkenning te verbeteren. *Data augmentation* vervormt, rekt en roteert beelden om de dataset kunstmatig uit te breiden. Dit kan het betrouwbaarheid van de beeldherkenning verbeteren.

Nadat de training is voltooid, worden statistieken berekend (figuur 13) die laten zien hoe goed ons neurale netwerk heeft gewerkt. Met het netwerk dat nu is gemaakt, kunnen we een test uitvoeren op de Jetson.

Eerste test met de Jetson

Nadat het neurale netwerk getraind is, zullen we opnieuw een console moeten openen op de Jetson Nano. De gemakkelijkste manier is om in te loggen met de Nano via SSH. Met `edge-impulse-linux-runner --clean` start een wizard en vraagt naar onze toegangsgegevens. Als er meer dan één project is aangemaakt, moet het huidige project worden gekozen..

Zodra alles is ingesteld, geeft de console een classificatielijst weer terwijl de camera draait (figuur 14). De numerieke waarde achter elke individuele klasse geeft het betrouwbaarheidsniveau aan waarmee de klasse in het zojuist opgenomen beeld voorkomt, waarbij 1,0 het maximum is (d.w.z. 100% zeker) en 0,00 het minimum.

Als we nu kaart nummer één met de beeldzijde naar boven voor de camera leggen, zien we dat het betrouwbaarheidsniveau verschuift en dat een één wordt herkend met een betrouwbaarheid van 0,9420 (d.w.z. een zeer hoge mate van zekerheid).

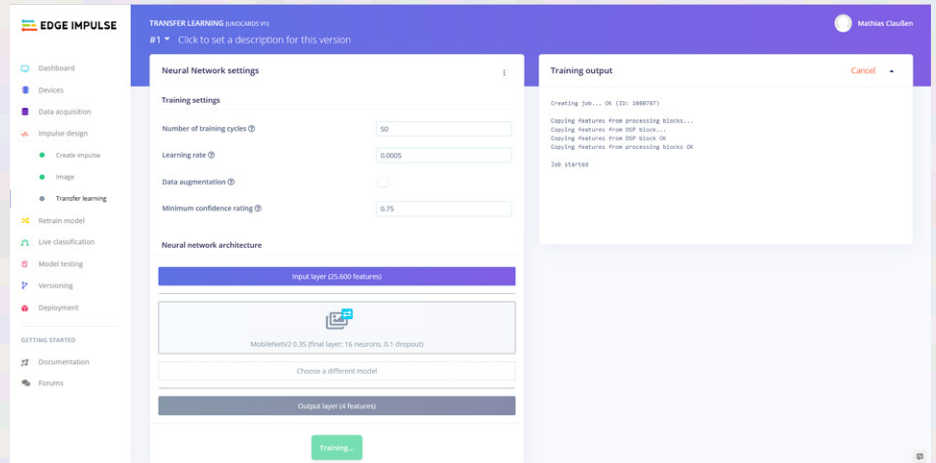
Volledig vertrouwen op de console-uitvoer is nu niet zo handig. Het kan zijn dat een kaart niet correct wordt herkend en dat je wilt achterhalen wat de Jetbot eigenlijk ziet. Hiervoor draait de `edge-impulse-linux-runner` tool niet alleen het neurale netwerk, het levert ook een webserver die bereikbaar is via poort 4912. Op de webpagina kan live worden gevolgd welke kaart op dat moment herkend wordt of, zoals figuur 15 laat zien, of de kaart fout herkend wordt. Ook is er de mogelijkheid om nieuwe beelden te genereren, waarmee het neurale netwerk kan worden hertraind. Om dit te doen, gebruik CTRL + C om `impulse-linux-runner` te stoppen. Met `edge-impulse-linux` verschijnt Jetbot nu weer als gegevensbron in het Edge Impulse-webportal en kunnen nieuwe beelden worden opgenomen.

Training Ronde Twee

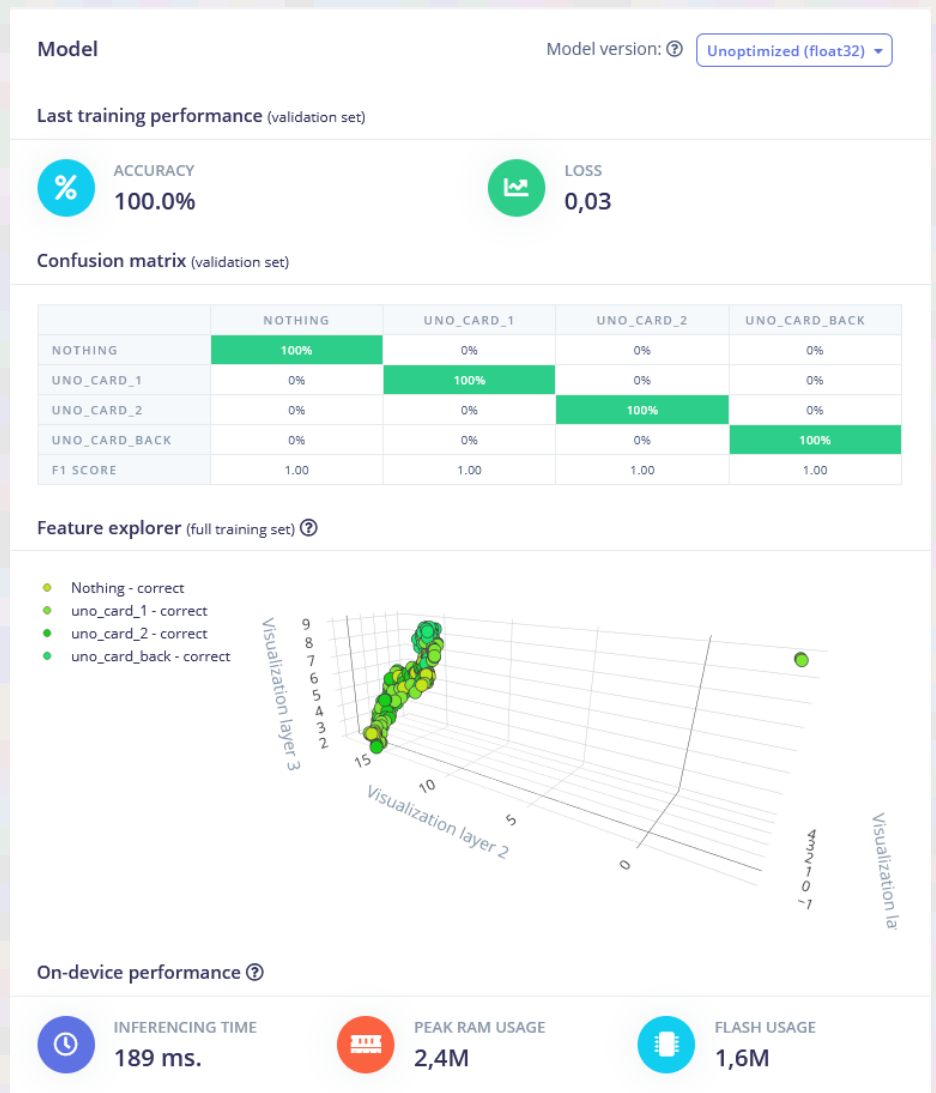
Met de extra, nieuwe gegevens kunt u een nieuwe trainingsprocedure starten. Hoe meer gegevens worden toegevoegd aan de categorie waar het huidige neurale netwerk moeite mee heeft, hoe beter het resultaat. Met `edge-impulse-linux` wordt de JetBot weer de dataverzamelaar; we kunnen opnieuw afbeeldingen van de speelkaarten maken en ze vervolgens van het juiste label voorzien. Wanneer het verzamelen van deze gegevens is voltooid, kan het trainingsproces opnieuw worden gestart met *Retrain Model* om de vorige parameters te hergebruiken, samen met de nieuwe gegevens (figuur 16). Nadat dat is gedaan, kunnen we de nieuwe versie van het netwerk testen op de Jetson. Met `edge-impulse-linux-runner` wordt het nieuwe neurale netwerk gedownload en uitgevoerd op de JetBot. We kunnen nu met een webbrowser in real time zien hoe de kaartbeelden met verbeterde betrouwbaarheid worden geclassificeerd (figuur 17).

Van Training naar Toepassing

Nadat de gegevens zijn opgenomen en het neurale netwerk is getraind, rijst de vraag hoe het kan worden gebruikt in een eigen applicatie. Edge Impulse biedt SDK's om verschillende programmeertalen te ondersteunen. Als C / C++ moet worden gebruikt, heeft de JetBot momenteel een probleem met de camerabesturing en de OpenMV bibliotheek die wordt gebruikt. Dit betekent dat er geen beelden van de camera kunnen worden vastgelegd met behulp van deze bibliotheek. (De hierboven beschreven beeldopname via Edge Impulse wordt niet beïnvloed; deze lijkt geen gebruik te maken van OpenMV). De GPU van de Jetson Nano wordt ook nog niet gebruikt voor beeldverwerking, dus alle berekeningen worden uitgevoerd door de CPU. Dit is misschien geen groot verlies voor kleine borden zoals de Raspberry Pi, maar bij de Jetson Nano betekent het dat een groot deel van zijn AI-rekenvermogen verloren gaat. Al het trainingswerk dat we voor het netwerk hebben gedaan, is niet voor niets geweest. Het getrainde neurale netwerk kan als platform-onafhankelijke bibliotheek worden geëxporteerd en dus ook op andere systemen draaien (zie download [6]). Het hoeft geen Raspberry Pi te zijn; een ESP32 met een camera is net zo geschikt om het netwerk



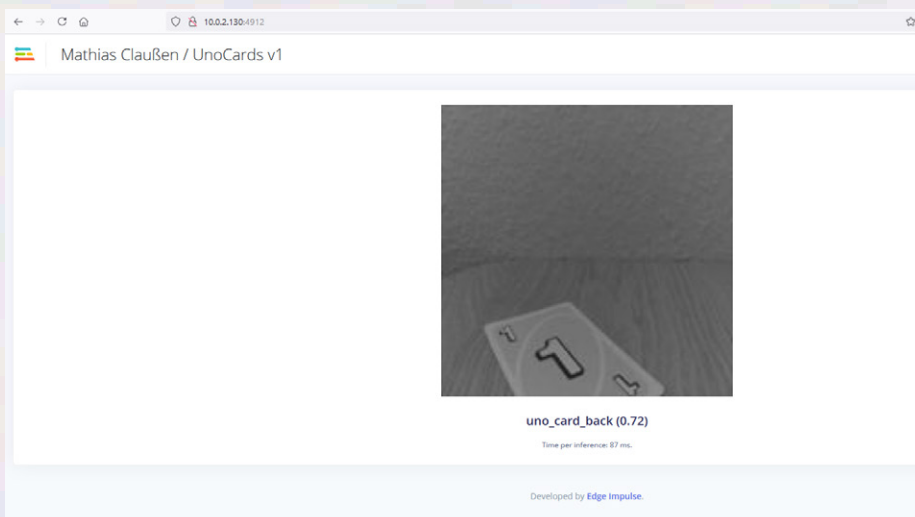
Figuur 12: Parameters instellen voor Training.



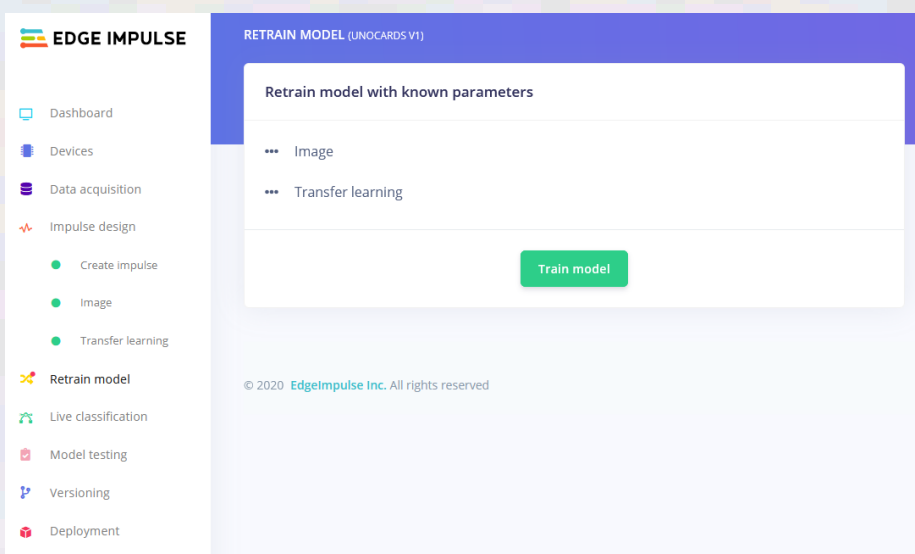
Figuur 13: Trainingsresultaten.

```
jetbot@jetson: ~  
uno_card_1: '0.0175',  
uno_card_2: '0.0173',  
uno_card_back: '0.0513'  
}  
classifyRes 66ms. {  
  Nothing: '0.9402',  
  uno_card_1: '0.0153',  
  uno_card_2: '0.0133',  
  uno_card_back: '0.0312'  
}  
classifyRes 56ms. {  
  Nothing: '0.9235',  
  uno_card_1: '0.0178',  
  uno_card_2: '0.0166',  
  uno_card_back: '0.0420'  
}  
}
```

Figuur 14: Beeldherkenningsdisplay op de console.



Figuur 15: Realtime weergave van beeldherkenning via een browser.



Figuur 16: Opnieuw trainen met meer gegevens.

dat we hier getraind hebben te gebruiken. Helaas zijn de ruwe gegevens die met Edge Impulse zijn opgenomen alleen daar beschikbaar. De beelden die we hebben gelabeld, kunnen niet worden geëxporteerd naar andere platforms of naar een lokaal systeem voor verwerking. Dit is een compromis dat we zullen moeten aanvaarden wanneer we het Edge Impulse-platform gebruiken.

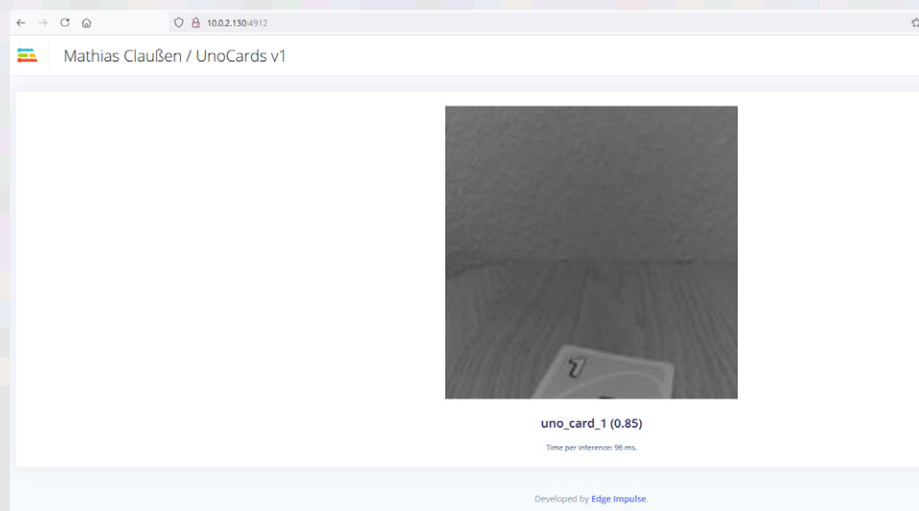
Watermerk? Ergo, het is een paard.

De evolutie heeft ons uitgerust met een verfijnd vermogen om patronen te herkennen. Wij leren bijvoorbeeld hoe een paard eruit ziet. Als wij een aantal afbeeldingen van paarden te zien krijgen, leren wij dat als het voorwerp vier benen heeft, als het hoeven heeft, als het een bepaalde grootte heeft, en als het bepaalde verhoudingen heeft, het waarschijnlijk een paard is. In de loop van ons leven wordt onze eigen definitie van een paard steeds verfijnder (een ezel is bijvoorbeeld geen paard, ook al hebben ze veel eigenschappen gemeen).

Bij klassieke beeldverwerking wordt een algoritme geleerd naar welke kenmerken in beelden moet worden gezocht om paarden te herkennen (bv. hoeven, ogen, manen, kleur, verhoudingen). Dit heeft echter zijn beperkingen wanneer het aankomt op het genereren van beelden van paarden. Dit is waar de neurale netwerken in het spel komen. Je geeft ze gewoon een enorm aantal beelden (enkele duizenden) en laat de manier waarop een paard wordt geïdentificeerd over aan de algoritmen die het neurale netwerk vormen. Deze aanpak klinkt plausibel. Wat kan er fout gaan?

Zoals Stuart Cording vertelde in het webinar "Get to know the NVIDIA Jetson Nano Developer Kit" [7], uploadde hij, terwijl hij aan een AI-project werkte, een reeks afbeeldingen die hij via een zoekmachine had gevonden, om te gebruiken als dataset om een paard op een foto te identificeren. Helaas bevatten de afbeeldingen een watermerk. Dit betekende dat het neurale netwerk het beeld als een paard identificeerde als het een watermerk in het verwerkte beeld detecteerde.

Iets dergelijks gebeurde toen we voor het eerst probeerden een UNO kaart dataset samen te stellen. Een deel van een hand was opgenomen in sommige van de beelden;



Figuur 17: Livebeeld met verbeterde betrouwbaarheid.

daarom werd de hand een van de meest bepalende kenmerken bij het onderscheiden van kaartnummers één en twee. We moesten de oorspronkelijke dataset weggooien en een geheel nieuwe set maken, waarbij we er zorg voor droegen dat irrelevante informatie werd weggesneden. Het is belangrijk voorzichtig te zijn bij het opnemen van de beelden van de dataset; een watermerk of een ander irrelevant kenmerk dat per ongeluk in de opnamen is opgenomen, zal het herkenningproces vertekenen.

Samenvatting en vooruitzichten

Het is een vrij eenvoudige procedure om ons eigen neurale netwerk te trainen met behulp van Edge Impulse en de Jetson Nano. Integratie van dit netwerk in een praktische toepassing zou dan de volgende stap zijn. De Jetson Nano is niet het best ondersteunde hardware platform voor deze toepassing; sommige van zijn resources blijven ongebruikt en er zijn wat softwareproblemen waardoor de camera niet direct werkt met de Edge Impulse SDK in sommige toepassingen.

Als u op zoek bent naar een eenvoudige kennismaking met AI en beeldherkenning, moet u Edge Impulse toch eens proberen. Het hoeft niet op een Jetson Nano te draaien. Een Raspberry Pi 4 zou het net zo goed doen. So, what's next? De Jetson Nano heeft genoeg kracht aan boord om al het werk te doen dat Edge Impulse hier heeft uitgevoerd. Dat zou dan de volgende stap moeten zijn: een neurale

netwerk maken dat in staat is UNO kaarten te herkennen en dat onafhankelijk op de Jetson Nano draait. ◀

210318-B-03



GERELATEERDE PRODUCTEN

- **NVIDIA Jetson Nano Developer Kit (SKU 19001)**
www.elektor.nl/19001
- **SparkFun JetBot AI Kit v2.1 (without Jetson Kit) (SKU 19685)**
www.elektor.nl/19685
- **ESP32-Cam-CH340 Development Board (SKU 19333)**
www.elektor.nl/19333

Vragen of opmerkingen?

Heeft u technische vragen of opmerkingen over dit artikel? E-mail de auteur op mathias.claussen@elektor.com of neem contact op met Elektor via editor@elektor.com.

Bijdragen van:

Ontwikkeling en tekst: **Mathias Claußen**
Redacteur: **Jens Nickel**
Vertaling: **Jelle Aarnoudse**
Lay-out: **Harmen Heida**

WEBLINKS

- [1] M. Claussen, "Image Processing with the Nvidia Jetson Nano (Part 1)," Elektormagazine.com, 5/18/2021.: <https://bit.ly/3yvX3fw>
- [2] M. Claussen, "Making Coffee with the MAX78000 and Some AI," Elektormagazine.com, 3/30/2021.: <https://bit.ly/3rW2Bh9>
- [3] Edge Impulse: <http://www.edgeimpulse.com>
- [4] A. Garrapa, "MCU Machine Learning With Edge Impulse," Elektor Industry, 2/2021.: <https://www.elektormagazine.com/magazine/elektor-238/59631>
- [5] Edge Impulse - Nvidia Jetson Nano : <https://docs.edgeimpulse.com/docs/nvidia-jetson-nano>
- [6] Download the trained network: <http://www.elektormagazine.nl/210318-B-01>
- [7] Elektor Webinar, "Meet the Engineers (Part 2): Get to Know the NVIDIA Jetson Nano Developer Kit," 5/6/2021.: <https://www.elektormagazine.nl/news/ontmoet-de-ontwikkelaars-deel-2-maak-kennis-met-de-nvidia-jetson-nano-developer-kit>