

Belgium

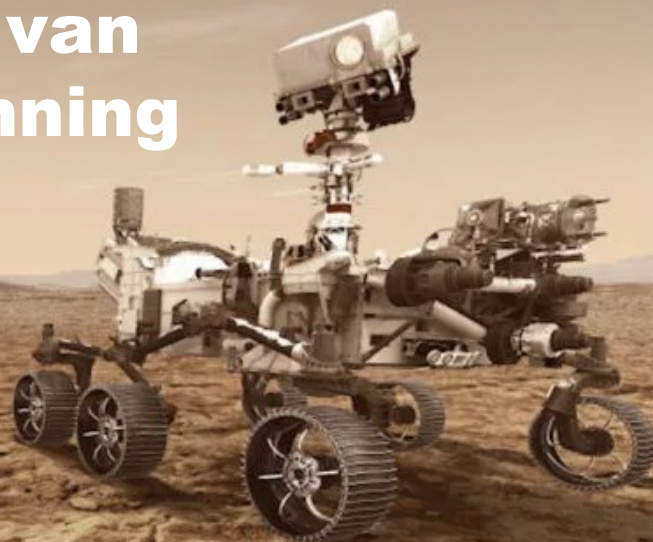
esero

La Scientothèque
L'ÉGALITÉ DES CHANCES PAR LES SCIENCES

De slimme Fetch-rover op Mars

Workshop rover programmeren met beeldherkenning
Lerarengids

Haal Mars-samples
van Perseverance
op met behulp van
AI beeldherkenning



- Train een eenvoudig beeldherkenningsmodel met 'Teachable Machine' ●●●●
- Programmeer de camera van de rover om Mars samples te detecteren ●●●●
- Programmeer de rover met Scratch voor leerlingen tot 14 jaar ●●●●●●
- Of programmeer de rover met Python voor oudere leerlingen ●●●●●●

OVER ESERO BELGIUM

ESERO is een scholenprogramma van de Europese Ruimtevaartorganisatie ESA. Het doel van dit programma is leraren van basisonderwijs en middelbaar onderwijs helpen om het populaire thema ruimtevaart in de klas te brengen, binnen hun lesopdracht. Dit doen we op drie manieren: **lesmateriaal** (online), **lerarenvormingen**, en **STEM projecten voor scholen**. Het aanbod is volledig gratis voor leraren in beroep en leraren in opleiding, en is afgestemd op de eindtermen in het onderwijs. Hedendaagse en inspirerende ruimtevaartmissies vormen de context diverse schoolvakken.

WWW.ESERO.BE

Nationale coördinator




Vlaamse coördinator




Frans- en Duitstalige coördinator




ESA Education beheert en coördineert alle ESERO's in Europa. Elke ESERO bestaat dankzij een cofinanciering van ESA en nationale partners. Het federaal wetenschapsbeleid (BELSPO) is de cofinancierende partner voor ESERO Belgium.



De slimme fetch-rover op Mars

Haal Mars-samples van Perseverance op met de fetch rover met behulp van AI (beeldherkenning)

Lerarenbundel

Cursus kenmerken

Doelgroep Leerlingen van 10-18 jaar.
Programmeren met Scratch: 10-14 jaar.
Programmeren met Python: 14-18 jaar.

Type Workshop, hands-on, in groepjes van twee

Lestijden Fase 1 (beeldherkenning): 2 uren
Fase 2a (programmeren met Scratch): 4 uren

Benodigheden

- Computer of tablet (1 toestel per 2 leerlingen) met internettoegang

Onderwerpen

- Artificiële intelligentie
- Geautomatiseerd leren
- Ordenen van afbeeldingen
- Voorspellende modellen
- Programmeren (Scratch, Python)

Samenvatting Perseverance is een NASA Marsrover die sinds 2021 bodemstaaltje neemt in de Jezero krater. De staaltjes ('samples') worden in gesloten buisjes op de grond gelegd, en later opgehaald door een 'fetch rover' om tenslotte terug te brengen naar de Aarde: de Mars Sample Return missie. De leerlingen gaan de camera van de Fetch rover zo programmeren dat hij automatisch bij elk camerabeeld kan zeggen of er een sample te zien is of niet. Zowel de oefening over het trainen de AI beeldherkenning als het programmeren is heel eenvoudig, en kan uitgevoerd worden door leerlingen (en leraren) zonder voorkennis.

Colofon

Eerste uitgave Mei 2023

Updates

Gebruiksvoorwaarden Dit lesmateriaal kan gratis gebruikt worden voor educatieve doeleinden. Wanneer je fragmenten eruit kopieert, dan vragen we om een correcte referentie op te geven naar het origineel. De meest recent versie kan je downloaden op www.esero.be > NL > Lesmateriaal.

AUTEURS

**ESERO Belgium / La
Scientothèque** Inhoud, concept (origineel “Reconnaissance d’images”).

**ESERO Belgium /
UGent
Volkssterrenwacht** Vertaling, aanpassing en layout van de Nederlandstalige versie

Je mening is belangrijk ESERO Belgium blijft steeds werken aan betere kwaliteit. We moedigen gebruikers van ons lesmateriaal aan om ons feedback te geven via ‘contact’ op www.esero.be. Wanneer je feedback een belangrijke bijdrage vormt aan de nieuwe versie, dan wordt je naam opgenomen in de auteurslijst (colofon). Op die manier kunnen jullie toekomstige gebruikers helpen om nog beter lesmateriaal te krijgen.

INHOUD

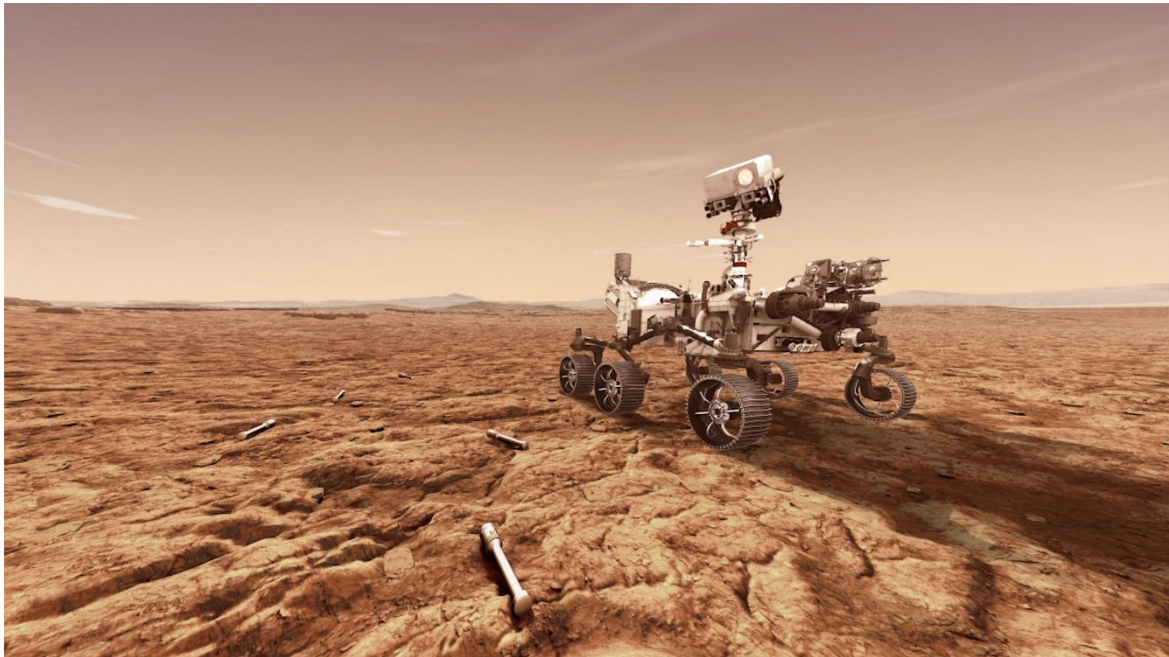
Cursus kenmerken	3
Colofon	4
FASE 1: Het model trainen	6
1 Inleiding	6
2 Teachable machine	7
3 Interface om het model te trainen	8
4 Klassen kiezen	9
5 Voorbeeld : een gezicht sorteerder.....	9
6 Modelleren en testen	10
7 Modellen exporteren.....	11
8 Herkenning van samples op Marsbodem.....	13
Fase 2a : Programmeren met Scratch	16
1 Overzicht	16
2 Adacraft.....	17
3 Een Adacraft project maken	17
4 Laat de ster spreken.....	18
5 Verander de ster in een Marsrover	19
6 Voorwaarden en lussen	21
7 Camera activering en bewegingsdetectie	24
8 Beeldherkenning met Teachable Machine-model	26
Fase 2b : Programmeren met Python	30
Bijkomende informatie	31

FASE 1: Het model trainen

Deze activiteit leert je werken met Google Teachable Machine. Het gebruikt kunstmatige intelligentie voor herkenning van afbeeldingen (Fase 1).

Daarna ga je de rover programmeren:

- in bloktaal (Block code) – Fase 2a
- of in tekstuele taal (Python) – Fase 2b



Perseverance nadert de bodemmonsters die moeten opgehaald worden (afbeelding: NASA/JPL).

1 Inleiding

Computerbeeld herkenning is een vorm van kunstmatige intelligentie op basis van machine learning: voorbeelden van afbeeldingen uit verschillende categorieën worden getoond aan de computer, en er wordt een leeralgoritme gebruikt om de computer nadien automatisch de verschillende categorieën te laten herkennen.

Enkele termen:

- De categorieën van afbeeldingen die herkend moeten worden (bijvoorbeeld gezichten, objecten, emoties, enz.) worden “**klassen**” genoemd.
- De set voorbeelden van de verschillende klassen die we gebruiken voor training wordt de **trainingsdataset** genoemd;
- **Machine learning** is de term die wordt gebruikt voor het feit dat we voorbeelden tonen van wat de computer moet leren herkennen;
- Het herkenningssysteem wordt ook het **voorspellingsmodel** genoemd.

Bij beeldclassificatie wordt een voorspellingsmodel getraind om verschillende klassen afbeeldingen uit een trainingsdataset te herkennen. De tool die je hier gaat gebruiken om het machine learning-model te trainen, is de Teachable Machine. Met behulp van foto's van een webcam kan je eenvoudig een beeldherkenningsmodel trainen. De afbeeldingen worden geassocieerd met klassen die het model zou moeten herkennen. De creatie van hun model (de training) zal in de cloud gebeuren door een dienst van Google.

We laten u twee voorbeelden zien: gezichtsherkenning en objectdetectie op Mars. Zodra de leerlingen begrijpen hoe het trainen van een model werkt, kunnen ze andere herkenningstoepassingen maken, bijvoorbeeld om objecten te herkennen, emoties op een gezicht, enz.

Deze tutorial laat je ook zien hoe je een model kunt testen op nieuwe afbeeldingen, om vervolgens dat model te exporteren om het te gebruiken in een Python of Scratch programmering.

2 Teachable machine

Vraag de leerlingen om de website van de teachable machine te bezoeken:

<https://teachablemachine.withgoogle.com>

Als je helemaal naar het onderste van de pagina scrolt dan kan je onderaan rechts de taal veranderen in Nederlands. Je kan de taal kiezen via het menu rechts onderaan de pagina.

Klik op "Aan de slag" (Get started):



Teachable Machine

Train een computer om je eigen afbeeldingen, geluiden en houdingen te herkennen.



Een snelle, makkelijke manier om machine learning-modellen te maken voor je sites, apps en andere platforms. Je hebt geen expertise of codes nodig.

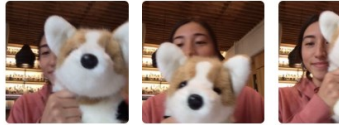
[Aan de slag](#)

Metal 95%
Not Metal 5%

Op de pagina « nieuw project » dat je dan te zien krijgt: klik op 'projectafbeelding':

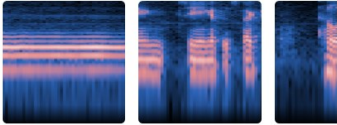
Nieuw project

 Een bestaand project openen vanuit Drive.  Een bestaand project openen vanuit een bestand.




Projectafbeelding

Train je model met afbeeldingen, bestanden of je webcam.



Audioproject

Leer op basis van geluiden van 1 seconde, uit bestanden of je microfoon.



Project met houdingen

Train je model met afbeeldingen, bestanden of je webcam.

En vervolgens klik je op 'Model voor standaard afbeelding':

Nieuw beeldproject

Model voor standaard afbeelding

Geschikt voor algemeen gebruik

Afbeeldingen in kleur, 224 x 224 pixels

Export naar TensorFlow, TFLite, and TF.js

Modelgrootte: ongeveer 5 MB

Model voor ingesloten afbeelding

Geschikt voor microcontrollers

Afbeeldingen in grijs tinten, 96 x 96 pixels

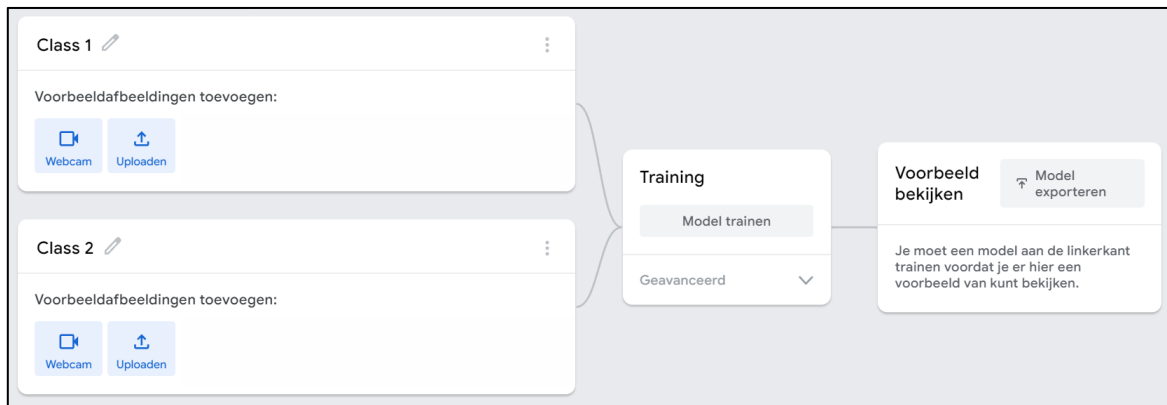
Export naar TFLite voor Microcontrollers, TFLite, and TF.js

Modelgrootte: ongeveer 500 KB

[Bekijk welke hardware deze modellen ondersteunt.](#)

3 Interface om het model te trainen

Je hebt nu een pagina waarop een machine training kan uitgevoerd worden:



Het bestaat uit drie delen:

- Aan de linkerkant kun je afbeeldingen toevoegen voor verschillende klassen. Standaard vraagt de interface om afbeeldingen toe te voegen voor twee verschillende klassen, genaamd 'Klasse 1' en 'Klasse 2'. Je kunt extra klassen toevoegen door onderaan op 'Klasse toevoegen' te klikken.
- In het midden kun je met de knop 'Training' het model trainen.
- Aan de rechterkant, in 'Voorbeeld bekijken', kun je het model testen en exporteren zodra je het hebt getraind.

4 Klassen kiezen

De keuze van klassen hangt af van wat je wilt dat de computer herkent. We laten je hier twee voorbeelden zien:

- een sorteerder die gezichten herkent,
- en een sorteerder die staaltjes op de Marsbodem kan vinden (deze sorteerder kan later worden gebruikt voor activiteiten met betrekking tot Mars en de constructie van een rover).

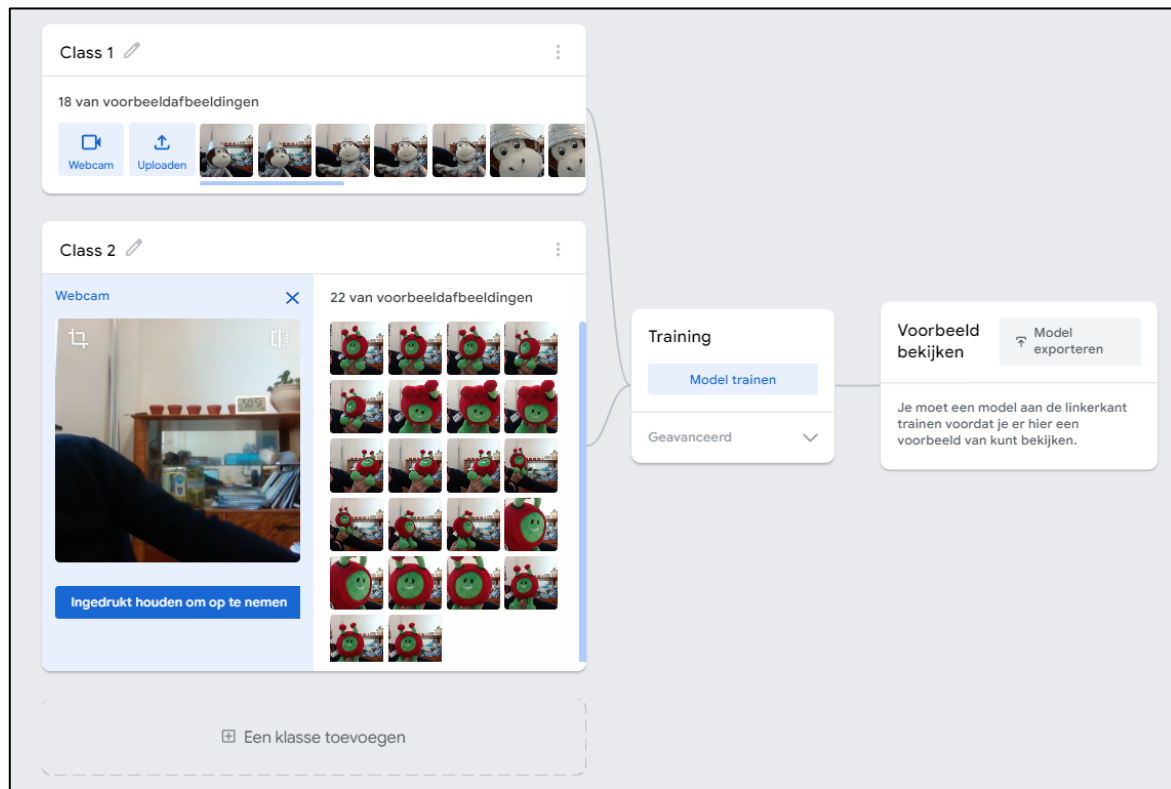
Studenten kunnen hun sorteerder natuurlijk trainen om andere dingen te doen, zoals herkennen van voorwerpen, fruit of gezichtsuitdrukkingen. Neem als oefening minstens enkele tientallen afbeeldingen. Hernoem de klassen naar wens door op het potloodpictogram te klikken dat bij elke klasse hoort en druk vervolgens op het camerapictogram om verschillende afbeeldingen van de overeenkomstige klasse te maken.

Wanneer je op de camera klikt, kan de webbrowser je om toestemming vragen om de camera te gebruiken. Dat moet je natuurlijk wel toestaan.

5 Voorbeeld : een gezicht sorteerder

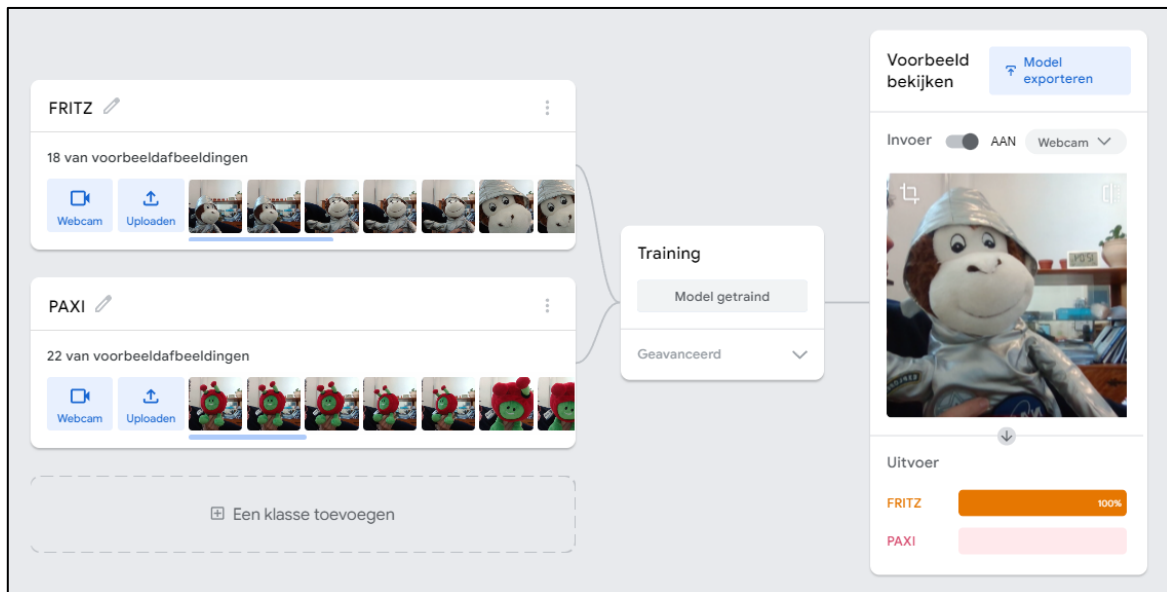
Neem in het geval van een gezicht sorteerder ongeveer twintig afbeeldingen voor elk van de te herkennen gezichten. Hier kunnen leerlingen de activiteit per twee doen, waarbij de twee klassen overeenkomen met hun twee gezichten. De namen van de twee klassen zijn de namen van de twee leerlingen.

Neem hier zeker zo gevarieerd mogelijke voorbeelden (positie van het gezicht ten opzichte van de camera, lichtomstandigheden, beeldachtergrond) om het herkenningssysteem zo betrouwbaar mogelijk te maken. Zodra er ongeveer twintig foto's zijn gemaakt voor elk van de twee gezichten, zal de interface er zo uitzien:



6 Modelleren en testen

Klik vervolgens op "Training". Het duurt meestal minder dan een minuut voordat de training is voltooid. Als het klaar is, zal het rechterdeel 'Voorbeeld bekijken' je het beeld laten zien dat van de webcam komt, en kun je je model testen.



- Wat als er geen gezicht aanwezig is, of het gezicht van een andere leerling?
- Wat gebeurt er als beide gezichten op de camera staan?

We kunnen zien dat het systeem, nadat het slechts twee klassen heeft geleerd die overeenkomen met twee gezichten, altijd zal proberen te associëren wat het waarneemt met één van de twee gezichten, zelfs als deze niet aanwezig zijn (of beide aanwezig zijn).

Om het systeem te verbeteren, kunnen andere klassen worden toegevoegd (bijvoorbeeld de klasse 'andere', met voorbeelden zonder gezicht, of beide gezichten tegelijkertijd).

Experimenteer met het systeem om betere resultaten te krijgen.

7 Modellen exporteren

Exporteer je model om het later te kunnen gebruiken met Scratch (Adacraft) of Python. Klik hiervoor op 'Exporteren'. Het volgende venster verschijnt:

Je model exporteren om in projecten te gebruiken. ✕

Tensorflow.js ⓘ Tensorflow ⓘ Tensorflow Lite ⓘ

Je model exporteren:

Uploaden (deelbare link)
 Downloaden
Upload mijn model

Jouw deelbare link:

https://teachablemachine.withgoogle.com/models/[...]

When you upload your model, Teachable Machine hosts it at this link. (FAQ: [Who can use my model?](#))

Codefragmenten om in je model te gebruiken:

Javascript
p5.js
Bijdragen op Github

Learn more about how to use the code snippet on [github](#).

```

<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@latest/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@latest/dist/teachablemachine-image.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

  // the link to your model provided by Teachable Machine export panel
  const URL = "/my_model/";

```

Kopiëren

Het model exporteren voor gebruik met Scratch (Adacraft)

Op het tabblad 'Tensorflow.js' :

Klik op de knop 'Upload mijn model'.

Het duurt ongeveer twee minuten. Er verschijnt een link naar de sjabloon zoals hieronder:

Jouw deelbare link:

https://teachablemachine.withgoogle.com/models/vQ9QJ6skp/ Kopiëren

When you upload your model, Teachable Machine hosts it at this link. (FAQ: [Who can use my model?](#))

✓ Je cloudmodel is up-to-date.

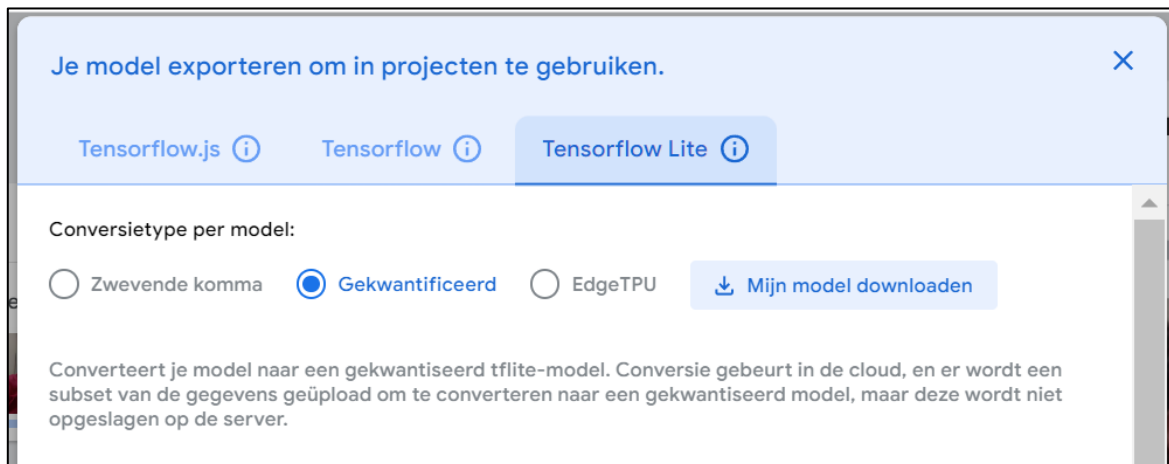
<https://teachablemachine.withgoogle.com/models/GkRrSvtDO/>

Kopieer het zodat je het later opnieuw kunt gebruiken in het modelblok van Adacraft (je kunt ook rechtsonder in het venster op 'Kopiëren' klikken om de link te kopiëren, die je vervolgens in Adacraft kunt plakken).

Het model exporteren voor gebruik met Python

Voor het gebruik van het model met Python is het noodzakelijk om de Tensorflow Lite-versie van het model te gebruiken.

Ga hiervoor naar het tabblad 'Tensorflow Lite' en selecteer 'Gekwantificeerd':



Klik dan op 'Mijn model downloaden'.

Je moet ongeveer 30 seconden wachten voordat het model is geconverteerd. Er verschijnt dan een venster waarin u een bestand met de naam 'converted_tflite.zip' kunt downloaden, dat ongeveer 2 megabyte groot is. Download het bestand door het 'zip'-archief te openen. Het archief bevat twee bestanden:

- Een tekstbestand 'labels.txt'
- Een 'model.tflite'-bestand dat door Python kan worden geopend om het model te gebruiken.

Opmerking: als je een Coral hebt, selecteer je op het tabblad 'Tensorflow Lite' 'EdgeTPU' en vervolgens 'Mijn model downloaden'.

8 Herkenning van samples op Marsbodem

Het tweede voorbeeld heeft tot doel een sorteerder te maken die kan bepalen of een samplebuisje aanwezig is op Marsbodem. Er zijn dus twee klassen:

- ofwel bevat het door de camera waargenomen beeld een sample (een buisje met Marsbodem),
- ofwel is er geen sample buisje op te zien.

We zullen de eerste klasse "**buisje**" noemen en de tweede klasse "**andere**".

Om de training te doen print je eerst onderstaande afbeelding. We zien een buis op Marsbodem geplaatst (de buis zit ongeveer in het midden).



Opmerking: in plaats van de afbeelding van Marsbodem met de buis, kan je een buisvormig voorwerp gebruiken, zoals een potlood, een doos of iets anders.

De stappen zijn dan hetzelfde als voorheen: het model trainen, testen en exporteren. Voor deze training moeten de leerlingen twee klassen definiëren: 'Buisje' en 'andere'. Voor de klasse 'Andere': voeg foto's toe door aan de webcam alleen Marsbodem te tonen (van bovenstaande foto die je geprint hebt), zonder dat het sample buisje in beeld komt. En voor dezelfde klasse ('andere') kan je ook foto's van leerlingen toevoegen. Voor de klasse 'Buisje' ga je natuurlijk de foto aan de webcam tonen waarop het sample buisje zichtbaar is.

Nadat de foto's voor beide klassen zijn gemaakt, zou je een dataset moeten hebben zoals hieronder weergegeven.

Begin dan met trainen door op de knop 'Trainen' te klikken. Als het klaar is (ongeveer een minuut), kan je uw model testen door het blad voor de webcam te bewegen. Controleer of het model de aanwezigheid van de buis correct herkent.

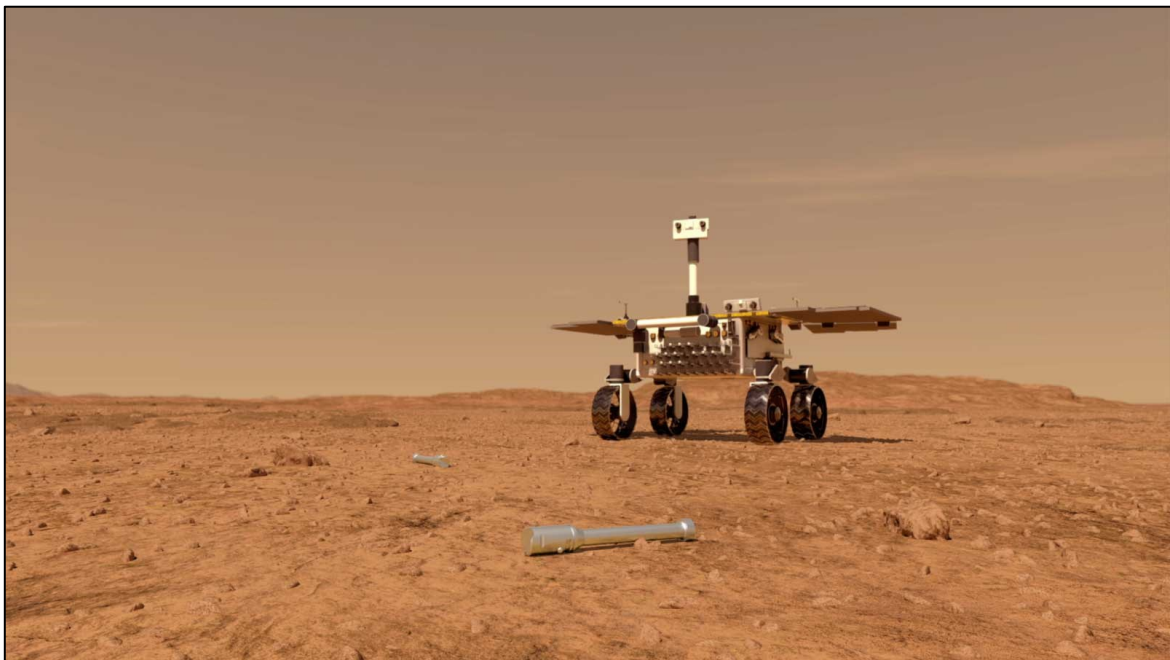
The screenshot displays the esero machine learning interface. On the left, two classes are defined: 'buisje' (19 example images) and 'andere' (23 example images). A central 'Training' panel indicates the model is trained and advanced. On the right, a 'Voorbeeld bekijken' (View Example) panel shows an input image of a pipe on a sandy surface. The output shows the 'buisje' class with a 100% confidence score, while the 'andere' class is not detected.

Je kan het model exporteren voor gebruik in Adacraft (Scratch) of Python op dezelfde manier als hierboven beschreven voor gezichtsherkenning.

<https://teachablemachine.withgoogle.com/models/lfHRfGGd/>

Fase 2a : Programmeren met Scratch

Eens de leerlingen een model getraind hebben voor beeldherkenning, kunnen we dit model gaan gebruiken om een Marsrover te programmeren, zodat deze de samples zou kunnen terugvinden op de Marsbodem. De rover die dit doet wordt de Fetch rover genoemd (Fetch = iets gaan halen).

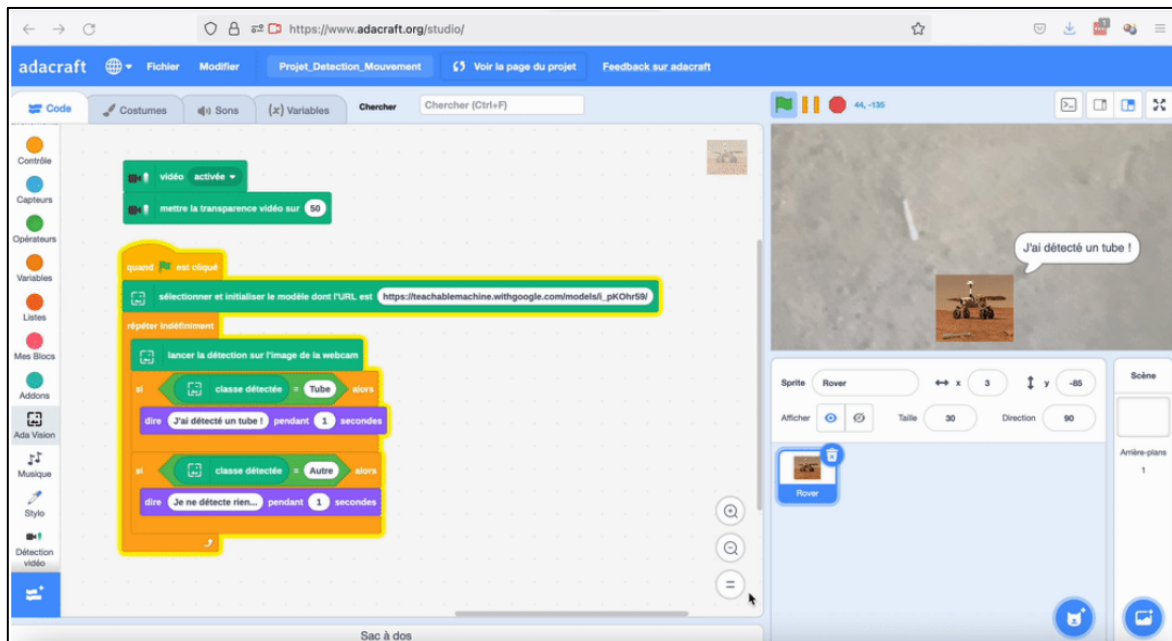


De Fetch rover

1 Overzicht

Het definitieve Scratch-programma wordt weergegeven in de onderstaande afbeelding. De afbeeldingen die op het scratch programmascherm worden weergegeven, worden 'sprites' genoemd.

De sprite (tekening) van de Fetch rover zegt of hij een buisje (of niets) detecteert op het beeld dat aan de camera wordt gepresenteerd.



2 Adacraft

Deze activiteit maakt gebruik van **Adacraft**, een aangepaste versie van **Scratch** inclusief extensies voor kunstmatige intelligentie. In deze activiteit gebruiken we de extensies

- Videodetectie'
- 'AdaVision'.

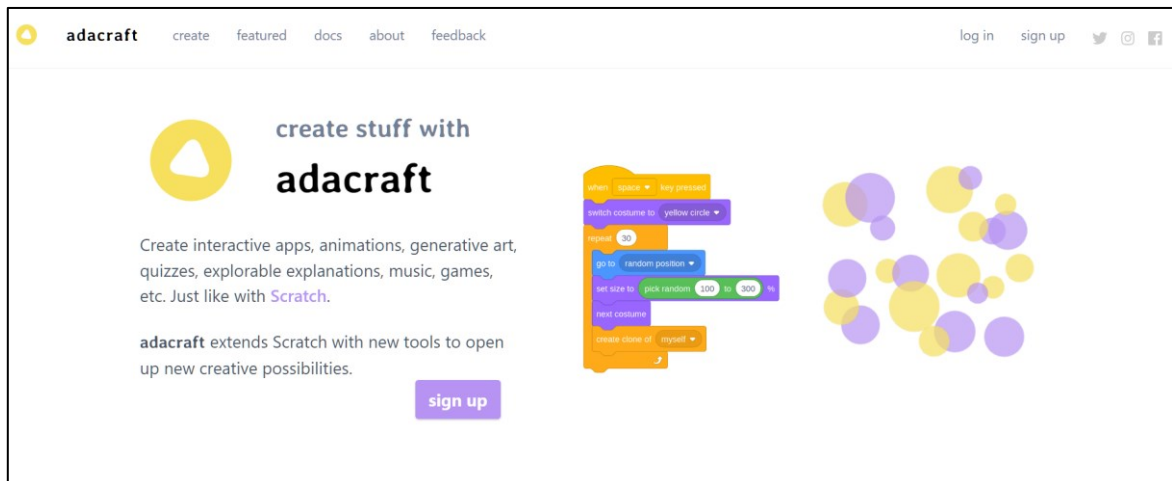
Met de extensie '**Videodetectie**' laat toe dat je beelden van de camera weergeeft en beweging detecteert.

Met de extensie '**AdaVision**' kan je een herkenningsmodel importeren die gemaakt is met Google Teachable Machine, en acties uitvoeren op basis van de afbeelding die door de camera wordt herkend.

3 Een Adacraft project maken

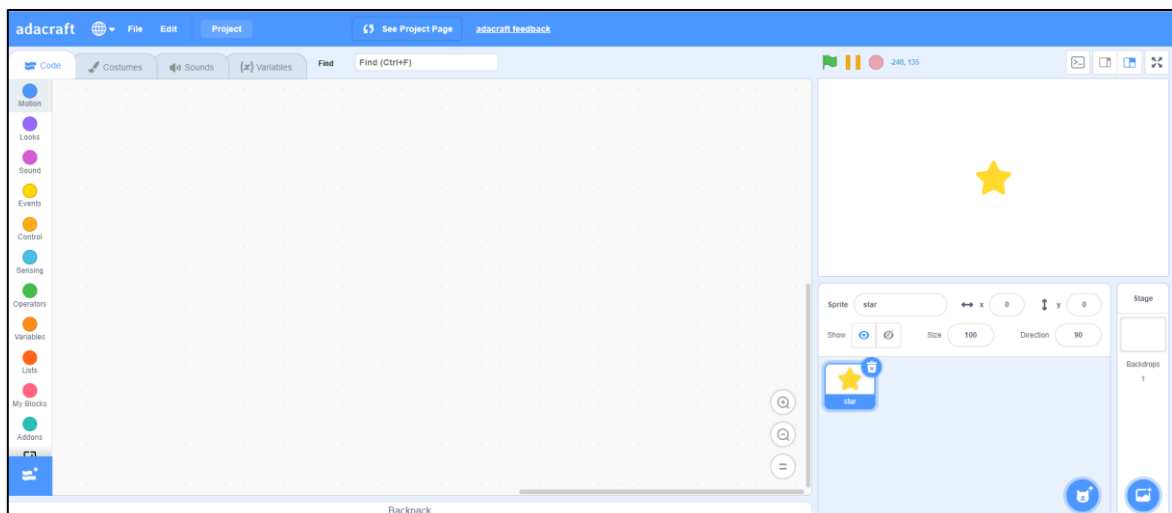
Ga naar <https://www.adacraft.org>

Klik daar op "**create**" bovenaan links.



Als ze niet zijn geregistreerd, verschijnt er een bericht dat hun werk niet kan worden opgeslagen. Het is daarom interessanter om u op de site te registreren door op 'Aanmelden' te klikken om de programma's op te slaan. Wil je alleen de interface testen, klik dan op 'Start met creëren zonder online opslaan'. In elk geval kunnen ze hun werk opslaan door het naar hun computer te downloaden.

Zodra het bericht over registratie en online opslaan weg geklikt is, komen ze terecht op de studio-interface waarin ze hun programma's kunnen maken.



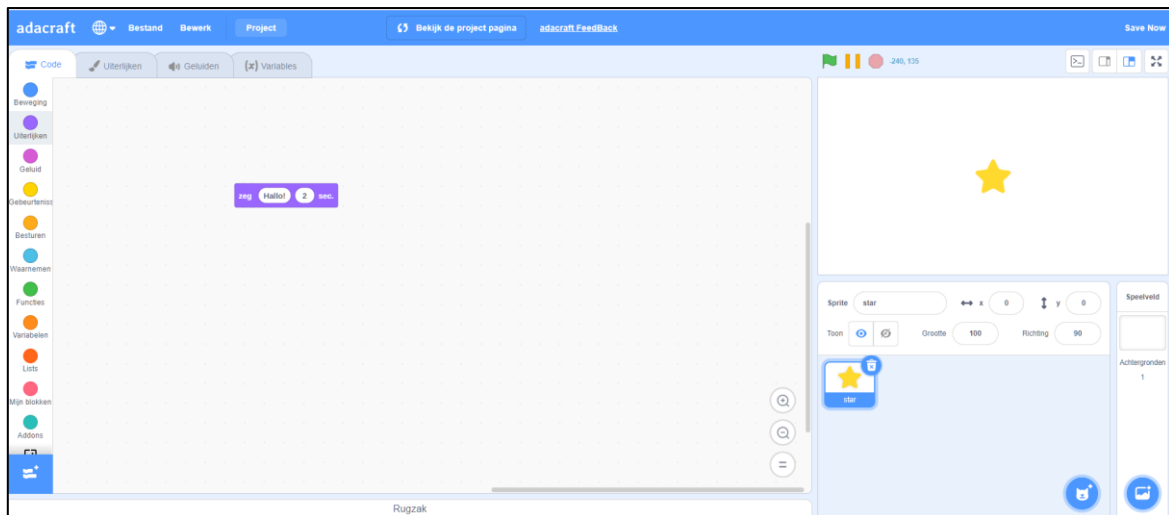
Je kan de taal van het programma kiezen (Nederlands is er bij) door het icoon van wereldbol te klikken bovenaan links.

4 Laat de ster spreken

De sprite die in het werkgebied wordt weergegeven, is standaard die van de ster. Om vertrouwd te raken met de interface gaan we even de ster laten spreken. We gebruiken hiervoor een codeblokje die er al in zit.

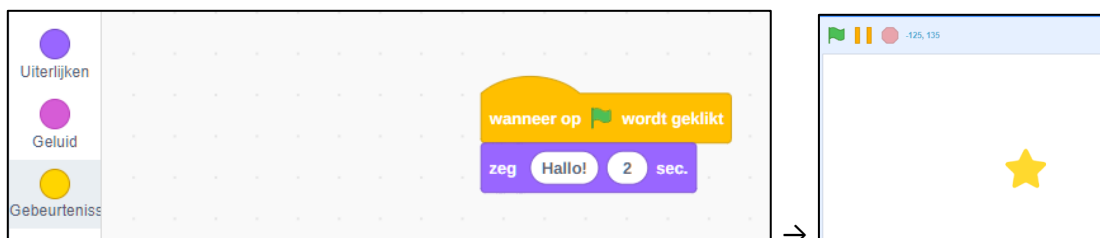
De kleurcodes links bevatten meerder codeblokjes.

Klik op de categorie 'uiterlijken'. Sleep het blokje "Zeg hallo" naar rechts in het programmavenster. Als je nu op dat blokje klikt, dan krijgt de ster een tekstballonnetje.



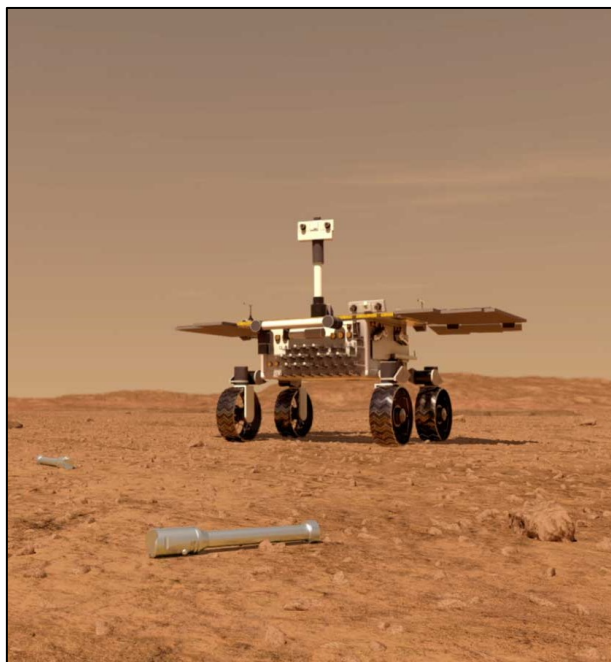
Ga nu naar de categorie "gebeurtenissen" in het instructiegebied links, en sleep het blokje "Wanneer op 'vlag' wordt geklikt" juist boven het reeds aanwezige paarse blokje van "Zeg Hallo".

Vanaf nu kan je het programma laten lopen door de groen vlag te klikken boven de sprite, en stoppen door de rode zeshoek te klikken.



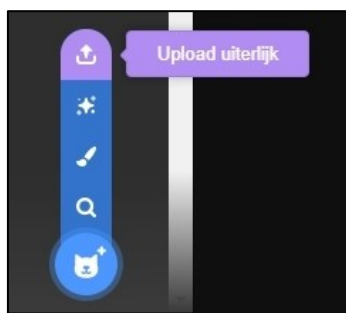
5 Verander de ster in een Marsrover

Je gaat nu de ster vervangen door een Marsrover zoals hieronder:



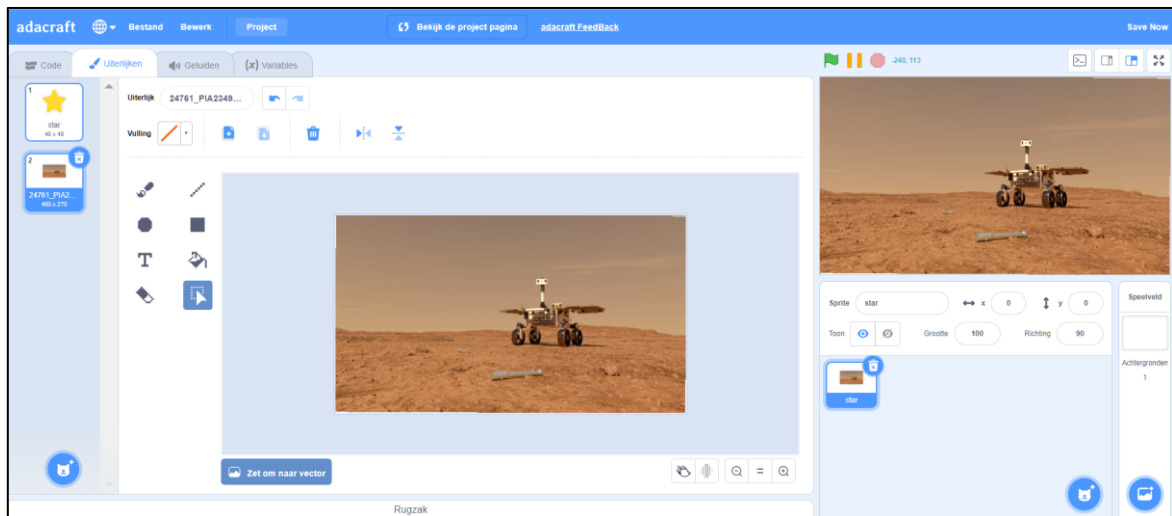
Deze afbeelding van NASA kan je downloaden op www.esero.be (de pagina van dit lesmateriaal).

Om de ster te vervangen door deze afbeelding, ga je naar het tabblad 'Uiterlijken' en plaats je de muis op het kattenpictogram linksonder ('Kies een uiterlijk'). Klik daar het bovenste: "upload uiterlijk".



Selecteer nu het bestand van de afbeelding van de fetch-rover in de map waarin je het hebt gedownload. Ga dan terug naar het tabblad 'Code'.

Het beeld is te groot en neemt meer dan het venster in beslag. Je kan het beeld aanpassen (uitzoomen) door in het venster onderaan rechts de vergrootglasjes te gebruiken.



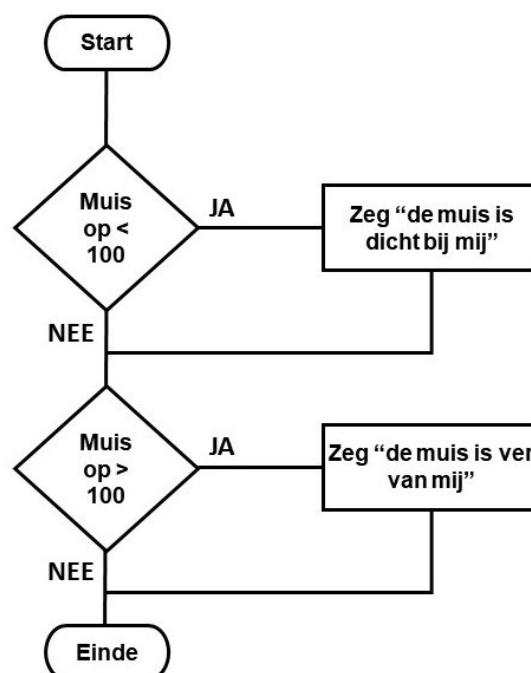
Als je nu opnieuw op de groen vlag klikt, dan zal de Fetch rover Hallo zeggen.

6 Voorwaarden en lussen

We gaan nu een extra functie programmeren: Laat de rover iets zeggen op basis van de muispositie:

- Als de muis dicht bij de sprite van de rover is, zegt de rover 'De muis is dicht bij mij!'
- Als de muis ver van de sprite van de rover verwijderd is, zegt de rover 'Muis is ver van mij verwijderd!'

Het stroomschema dat bij dit programma hoort, wordt hieronder weergegeven:



In woorden:

Bij het opstarten van het programma testen we eerst of de muis zich dicht bij de rover bevindt (een afstand minder dan 100 pixels).

Zo ja, dan toont het programma 'De muis is dicht bij mij!'

Dan testen we of de muis ver van de rover verwijderd is (een afstand groter dan 100 pixels).

Zo ja, dan toont het programma 'De muis is ver van mij verwijderd!'.

Om dit stroomschema in Scratch te programmeren, hebben we volgende codeblokkjes nodig:

Categorie in linkse instructiekolom Blokje

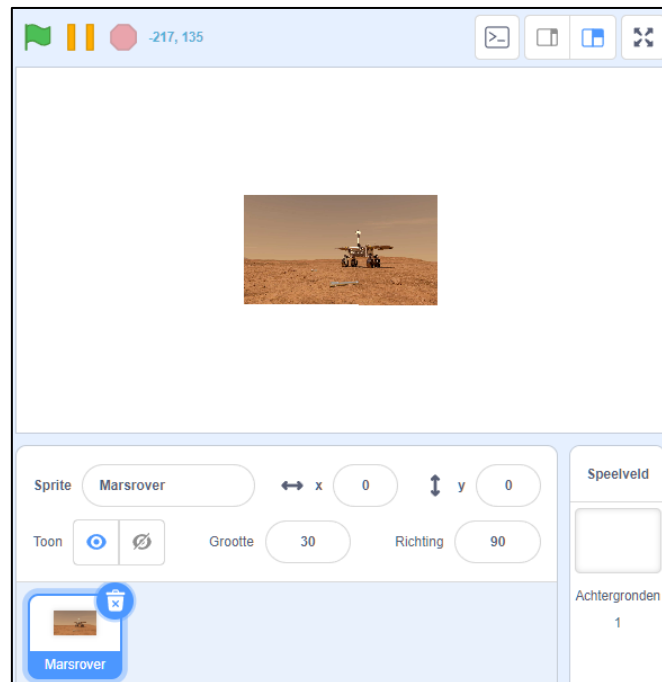
Besturen (oranje)	als ... dan
Functies (groen)	Vergelijking
Waarnemen (licht blauw)	Afstand tussen muis en sprite
Uiterlijken (paars)	Zeg "... " (zonder tijdsduur)

Begin eerst met het blokje "als...dan", en bouw zo het volgende programma op:



Je kan het programma nu al eens testen.

Zet eerst de 'grootte' van de sprite (de Marsrover) op 30 (in plaats van 100), zoals op het voorbeeld hieronder.

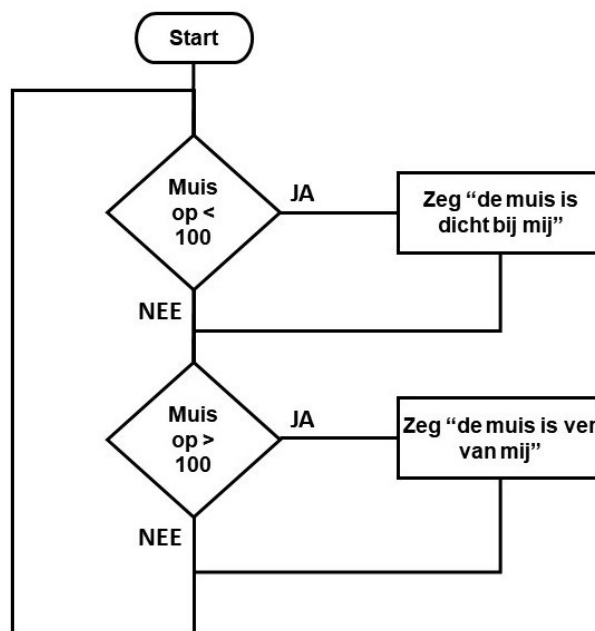


Klik op de groene vlag, en het programma begint te lopen.

Je zal zien dat de rover altijd zegt “de muis is ver van mij”.

Dat klopt, want het programma stopt altijd meteen na het klikken op de groene vlag. En dat is verder dan 100 pixels van de rover.

Daarom moeten we een lus toevoegen opdat het programma oneindig zou blijven doorlopen. Het nieuw stroomdiagram ziet er als volgt uit (er is geen einde meer):



Dit gaan we maken in Scratch:

Uit de categorie 'Besturen' (oranje): neem het blokje "Herhaal" (zonder einde, zonder cijfer), en sleep dat over de andere blokjes in het programmavenster:



Als je nu de groene vlag klikt, dan zal de rover wel de juiste boodschappen geven wanneer je de muis er naartoe beweegt.

7 Camera activering en bewegingsdetectie

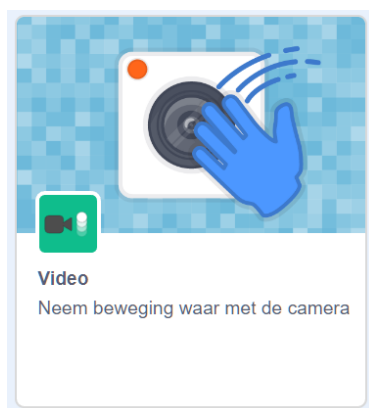
Je gaat nu een vergelijkbaar programma maken. In plaats van te detecteren of de muis zich dicht bij de rover bevindt, detecteert het programma of er beweging is voor de camera van de rover.

Activeer de camera

De programmeerblokken voor de camera en de bewegingsdetectie zitten in een uitbreiding. Om toegang te krijgen, klik je op het pictogram linksonder "**Voeg een uitbreiding toe**".



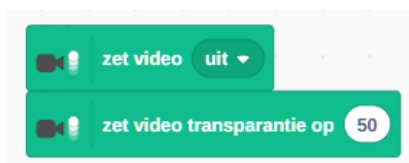
Dit brengt je naar een pagina met alle beschikbare extensies. Blader omlaag op de pagina om de extensie 'Videodetectie' te vinden, en klik het aan.



Je kunt de extensie dan zien onderaan in het instructiegebied (linkse kolom). De browser zal je vragen of hij toegang heeft tot de camera. Sta het toe. Je zult dan zien dat de video die van je webcam komt op de achtergrond in de scène wordt weergegeven.

Camera- en transparantiecontrole

Het is mogelijk om de camera aan en uit te zetten met het **'Zet video ...'** blok, en de transparantie te kiezen met het **'Zet video transparantie op...'** blok. Plaats de twee blokken op elkaar op een willekeurige plaats in het programmeervenster.

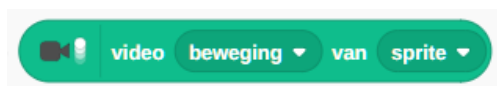


Probeer deze knoppen eens uit. Je kan de waarden "uit" en "50" eenvoudig wijzigen, en vervolgens klik je op het cameraatje links in de groene knop. Je zal dan het effect meteen zien in de scene.

De transparantie is een cijfer tussen 0 (gewoon camerabeeld ondoorzichtig) en 100 (volledig transparant, dus je ziet geen beeld meer). We raden aan om 50 te kiezen, maar je bent natuurlijk vrij om zelf iets uit te proberen.

Bewegingsdetectie

Bewegingsdetectie op een sprite kan aan het programma worden toegevoegd dankzij het blok **'video beweging van sprite'** in de categorie video.



Wijzig het eerder geschreven programma door:

- De blokken **'afstand tot muisaanwijzer'** vervangen door de blokken **'video beweging van sprite'**
- Door de waarde te wijzigen van 100 naar 20 (de drempel voor bewegingsdetectie ligt tussen 0 en 100)
- Door te veranderen wat de rover zegt:

- ▶ Bij beweging minder dan 20: voeg een “zegsec” blokje toe (categorie uiterlijken) met de tekst “ik detecteer niets”, en zet die op 1 seconde.
- ▶ Bij beweging meer dan 20: voeg een “zegsec” blokje toe (categorie uiterlijken) met de tekst “ik detecteer beweging”, en zet die op 1 seconde.

Dan ziet je programma er zo uit:



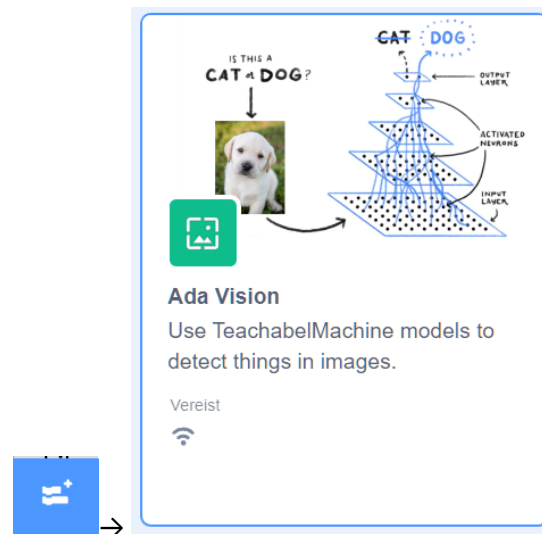
Probeer het maar eens uit (kik op de groene vlag). Je kunt de bewegingsdetectiedrempel (nu 20) wijzigen om de detectiegevoeligheid te wijzigen

8 Beeldherkenning met Teachable Machine-model

De leerlingen gaan de rover nu programmeren met beeldherkenning. Je gebruikt hiervoor een voorspellingsmodel dat je eerder hebt gemaakt met de Teachable Machine.

De AdaVision-extensie toevoegen en het model laden

Voor beeldherkenning heb je de AdaVision-extensie nodig, opnieuw te vinden als je de knop “voeg een uitbreiding toe” gaat aanklikken. Deze extensie laat toe dat je een beeldherkenningsmodel gaat uploaden, en dat je vervolgens gaat vaststellen wat er op het beeld te zien is.



Je zal zien dat de blokjes van deze extensie in het Engels zijn.

Het eerste blok in deze categorie gaan we gebruiken om ons beeldherkenningsmodel te laden:



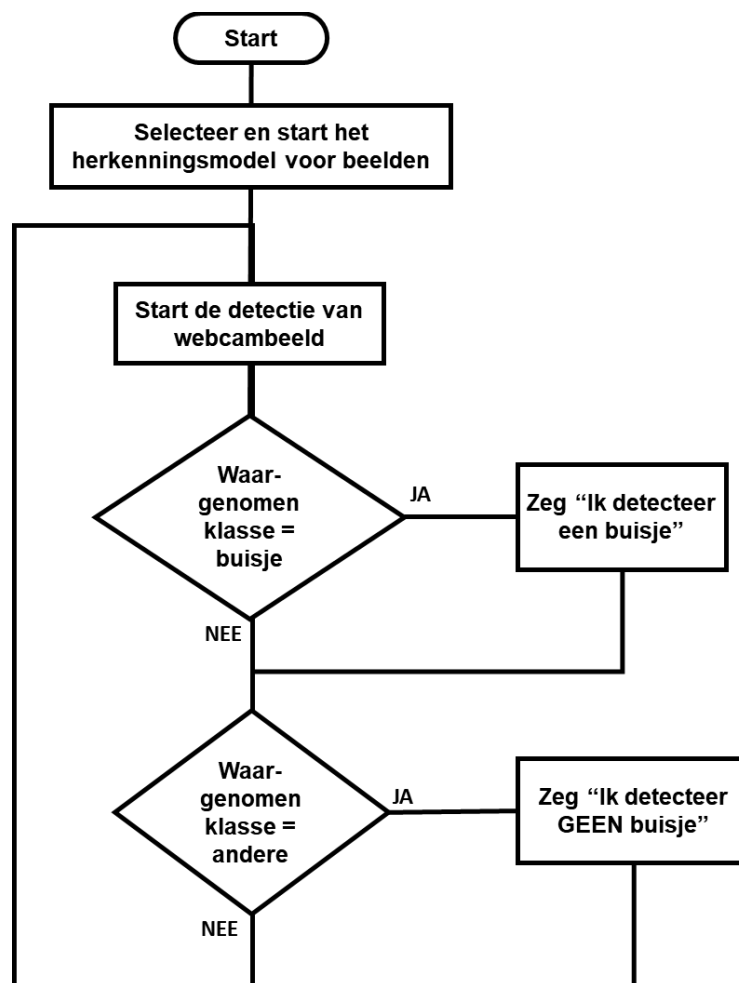
select and init the model which URL is <https://teachablemachine.withgoogle.com/models/Se2BsLUIB/>

We gaan dit blok in het programma zetten, juist onder “**wanneer op groene vlag wordt geklikt**”

Sample buisje: detectie

Het programma voor buisdetectie lijkt sterk op dat voor bewegingsdetectie:

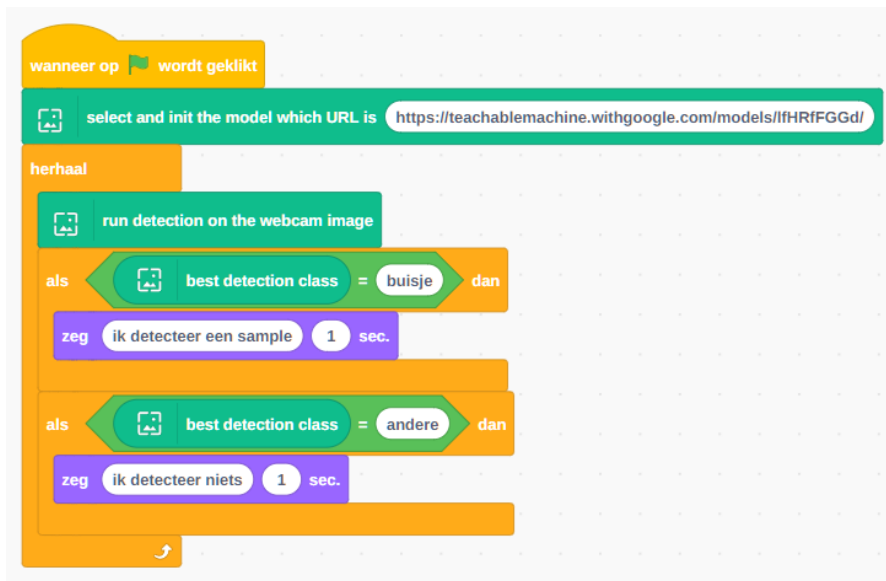
In plaats van te testen of er beweging voor de camera is, test je of er een buis aanwezig is. Dit geeft het volgende stroomschema:



Wijzig het schema voor bewegingsdetectie als volgt:

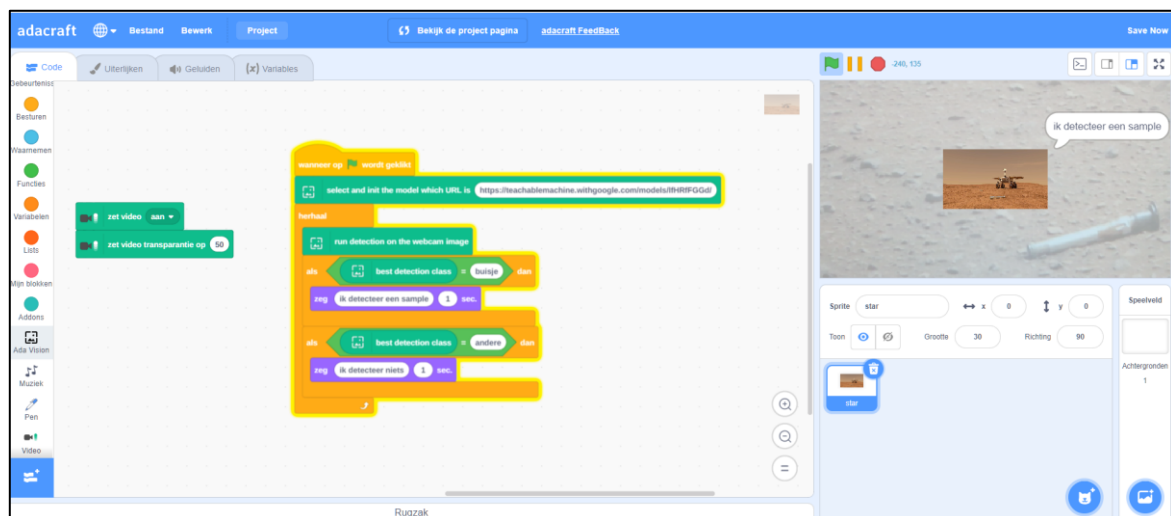
- Voeg een blok "**Run detection on the webcam image**" toe na het blok "**Herhaal**".
- Neem in categorie 'Functies' een vergelijkingsblok "**...= 50**" en zet het in de "**als ... dan ...**" blok, om te testen of de herkende afbeelding een 'buisje' is, of 'andere'. De waarde 50 moet dan uiteraard vervangen worden door de naam van de klasse.
- In het linkse vak van de vergelijkingsblok plaats je het blok "Best detection class". In het rechtse vak van dit blok schrijf je de naam van de klasse 'buisje'.
- Dezelfde aanpassingen doe je ook voor de andere klasse 'andere'.
- Wijzig de tekst die de rover zegt. Bijvoorbeeld:
 - ▶ "Ik heb een buis ontdekt" wanneer de gedetecteerde klasse "buisje" is
 - ▶ "Ik detecteer niets" wanneer de gedetecteerde klasse "andere" is.

Dan ziet je programma er zo uit:



Let op: In het blok “**Select init the model which URL is...**”, moet de URL van je eigen model uit teachable machines komen.

Start vervolgens het programma. Je rover kan nu detecteren of er zich een buis voor bevindt.



Fase 2b : Programmeren met Python

< Wordt binnenkort aangevuld >

Bijkomende informatie

Om Scratch te leren:

Ressources Scratch de la Communauté de l'Apprentissage de l'Informatique

<https://cai.community/ressource/premiers-pas-et-plus-avec-scratch/>

“Je découvre Scratch”, vidéo de 30 minutes de la Scientothèque

https://www.youtube.com/watch?v=9zSp4_0lg1o&t=284s

Andere:

Fondation La Main à la Pâte

<https://fondation-lamap.org/en/node/1684>

CodeClub - Scratch Module 1

<https://projects.raspberrypi.org/nl-NL/codeclub/scratch-module-1>

Fondation Raspberry - Introduction à Scratch

<https://projects.raspberrypi.org/fr-FR/pathways/scratch-intro>