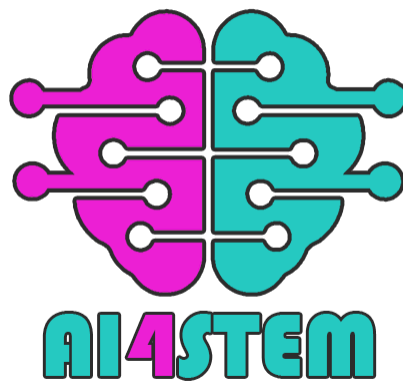




Medegefinancierd door  
de Europese Unie

Gefinancierd door de Europese Unie. De hier geuite ideeën en meningen komen echter uitsluitend voor rekening van de auteur(s) en geven niet noodzakelijkerwijs die van de Europese Unie of het Europese Uitvoerende Agentschap onderwijs en cultuur (EACEA) weer. Noch de Europese Unie, noch het EACEA kan ervoor aansprakelijk worden gesteld.

# AI & computervisietechnologie



Introductie van de 5 grote ideeën op het gebied van  
kunstmatige intelligentie met behulp van het internet der  
dingen in STEM-onderwijs  
T2.4 IoT-projectenontwerp en ontwikkeling van middelen

29.08.2023 | EMPHASYS CENTRE  
PROJECT NUMBER: 2022-1-FR01-KA220-SCH-000085611

# AI4STEM IOT-projecten

## Project: AI & computervisietechnologie

### Copyright

© Copyright het AI4STEM Consortium

2022-1-FR01-KA220-SCH-000085611

Alle rechten voorbehouden.



AI4STEM IOT-projecten Project: AI & computervisietechnologie © 2023 bij [AI4STEM CONSORTIUM](#) is gelicentieerd onder [Naamsvermelding-NietCommercieel-GelijkDelen 4.0 Internationaal](#)

## Inhoudsopgave

1 . Inleiding tot het project .....	4
1.1 De reikwijdte van het Project.....	4
1.2 De doelgroepen.....	4
1.3 Het doel van dit document .....	4
2. Woordenlijst van de eenheid.....	4
3. Inleiding tot de “AI in Vision” .....	5
3.1 Beschrijving .....	5
3.2 Leerdoelen & resultaten .....	5
3.3 Geschatte duur van de Eenheid .....	5
3.4 Activiteit 1 – Introductie van het grote idee van perceptie via IoT: .....	6
3.4.1 Beschrijving .....	6
3.4.2 Hardware .....	6
3.4.3 Installatie.....	7
3.4.4 Experiment 1 .....	13
3.5 Activiteit 2: Introductie van het idee van representatie en redeneren.....	14
3.5.1 Beschrijving .....	14
3.5.2 Hardware .....	14
3.5.3 Opgericht .....	15
3.5.4 Oefening: Train het AI-model om verschillende vormen te herkennen .....	17
3. 6 Activiteit 3: Introductie van het idee van leren door een model voor gezichtsherkenning te trainen .....	18
3.6.1 Beschrijving .....	18
3.6.2 Hardware .....	19
3.6.3 Installatie.....	19
3.6.4 Code .....	20
3.6.5 Oefening : Test of het AI-model u herkent .....	21
3. 7 Activiteit 4: Introductie van het idee van natuurlijke interactie door een getraind model te integreren in een AI-toepassing.....	22
3.7.1 Beschrijving .....	22
3.7.2 Hardware .....	23
3. 7.3 Installatie.....	24

3.7.4 Oefening: Alarmsysteem.....	27
3.8 Activiteit 5: Introductie van het idee van maatschappelijke impact .....	27
3.8.1 Beschrijving .....	27
3.8.2 Hardware .....	27
3.8.3 Installatie.....	28
3.8.4 Het idee van natuurlijke interactie introduceren door een model voor gezichtsherkenning te trainen.....	30
3.9 Aanvullend materiaal en hulpmiddelen.....	30

## 1 . Inleiding tot het project

Voor dit project zullen micro:bit en HuskyLens worden gebruikt om verschillende activiteiten te voltooien. HuskyLens is een slimme vision-sensor of cameramodule die kunstmatige intelligentie (AI) en computervisietechnologie combineert om verschillende taken uit te voeren, zoals **objectherkenning** , **gezichtsherkenning** , **tracking** , **kleurherkenning** en meer. Het integreren van kunstmatige intelligentie (AI) met de HuskyLens-camera en micro:bit is een spannend project waarmee je interactieve en intelligente toepassingen kunt creëren.

### 1.1 De reikwijdte van het Project

De reikwijdte van het project is om micro:bit en HuskyLens te gebruiken om een model voor kunstmatige intelligentie te creëren. Het model wordt **getraind** via de HuskyLens-camera en geeft **output** via de micro:bit-tool.

### 1.2 De doelgroepen

Het project is voornamelijk gericht op de directe betrokkenheid van docenten, voornamelijk uit het hoger basis- en secundair onderwijs.

### 1.3 Het doel van dit document

Het doel van dit document is om te demonstreren hoe je een AI-model kunt trainen en testen met behulp van de HuskyLens-camera en de uitvoer op de micro:bit kunt weergeven.

## 2. Woordenlijst van de eenheid

Woord	Definitie
Micro:bit	De micro:bit is een programmeerbaar computerbord in zakformaat, ontworpen voor het onderwijs. Het beschikt over een LED-matrix, verschillende sensoren en een microcontroller
HuskyLens	De HuskyLens is een vision-sensormodule die objecten, gezichten en gebaren kan herkennen. Het is uitgerust met een camera en ingebouwde verwerkingskracht om AI-gerelateerde taken uit te voeren

## 3. Inleiding tot de “AI in Vision”

### 3.1 Beschrijving

In dit onderdeel maken leerlingen kennis met het proces van het trainen en testen van een AI-model met de HuskyLens-camera. Ze zullen leren hoe ze de HuskyLens-camera kunnen instellen en configureren voor **beeldherkenningstaken**, het AI-model kunnen trainen om **specifieke objecten of patronen te herkennen** en ten slotte de camera kunnen koppelen aan een micro:bit om **de uitvoer van het AI-model weer te geven**. Gedurende de hele unit krijgen leerlingen een praktisch inzicht in computervisie, machinaal leren en hoe ze AI-technologieën kunnen integreren in echte projecten.

### 3.2 Leerdoelen & resultaten

Na succesvolle afronding van dit onderdeel moeten de leerlingen in staat zijn om:

- Inzicht in computervisie
- AI-modeltraining
- HuskyLens-configuratie
- Micro:bit-integratie
- Probleemoplossing
- Praktische toepassing
- Bedien HuskyLens
- Data verzamelen
- Train AI-modellen
- Communiceer met Micro:bit

### 3.3 Geschatte duur van de Eenheid

Dit is een nogal uitgebreid project dat meerdere uren nodig heeft om alle aspecten goed aan te pakken. De volgende duur is indicatief en kan variëren afhankelijk van de leeftijd en het niveau van uw leerlingen.

Activiteit 1: 90 minuten

Activiteit 2: 90 minuten

Activiteit 3: 45 minuten

Activiteit 4: 45 minuten

Activiteit 5: 45 minuten

## 3.4 Activiteit 1 – Introductie van het grote idee van perceptie via IoT:

### 3.4.1 Beschrijving

In deze les verkennen leerlingen de wereld van kunstmatige intelligentie en computervisie om de perceptiemethode te verkennen, door middel van **live volgen van objecten**, een functie die mogelijk wordt gemaakt door de geavanceerde mogelijkheden van de HuskyLens-camera en het micro:bit-platform.

Met behulp van de HuskyLens-camera en het micro:bit-platform ontdekken studenten hoe technologie kan worden gebruikt om live input te krijgen en output aan de gebruiker te geven. Het live tracking-aspect demonstreert niet alleen de real-time mogelijkheden van het systeem, maar benadrukt ook de dynamische aard van technologie bij het reageren op zijn omgeving.

In het tijdperk van snelle technologische vooruitgang heeft de combinatie van Internet of Things (IoT) en geavanceerde beeldverwerkingstechnologieën de weg vrijgemaakt voor baanbrekende toepassingen. Eén van die innovaties is de integratie van de HuskyLens, een krachtige slimme camera, met Micro:bit. Samen vormen ze een dynamisch systeem voor het realtime volgen van objecten, wat een revolutie teweegbrengt in de manier waarop we onze omgeving waarnemen en ermee omgaan.

Terwijl studenten zich bezighouden met het HuskyLens- en Micro:bit-systeem, doen ze niet alleen praktische ervaring op met de allernieuwste tools, maar nemen ze ook deel aan het voortdurende verhaal van de technologische vooruitgang.

### 3.4.2 Hardware

- Micro:bit-microcontroller
- IO-extender voor micro:bit
- HuskyLens



### 3.4.3 Installatie

#### 3.4.3.1 Bedrading

- Verbind de micro:bit met de IO-extender voor micro:bit.
- Gebruik vervolgens de volgende tabel om de IO-extender voor micro:bit te verbinden met de HuskyLens-camera.

IO-extender voor micro:bit-POORT	HuskyLens-POORT
SDA	T
SCL	R
GND	-
3V3	+

#### 3.4.3.2 Firmware bijwerken

Voordat u dit product gebruikt, wordt het ten zeerste aanbevolen om de firmware van de HuskyLens te installeren/bijwerken om toegang te krijgen tot de nieuwste functies en optimale stabiliteit te garanderen. Er wordt voorgesteld om de HuskyLens Uploader op Windows 10 te gebruiken voor het uploaden van firmware, omdat deze een grafische gebruikersinterface (GUI) en gebruiksvriendelijke bediening biedt.

Gebruik de volgende link om naar de DFROBOT-website te navigeren en de link te vinden om de firmware te installeren:

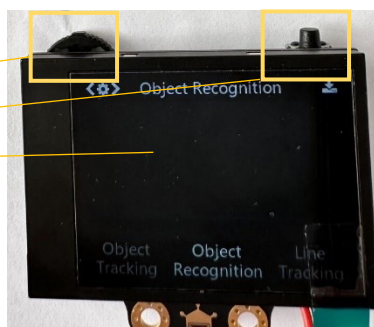
[https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)

Nadat u de bovengenoemde website heeft bezocht, navigeert u naar het gedeelte "Firmware bijwerken". Als u Windows heeft, ga dan naar sectie 4.1 en als u Linux of Mac heeft, ga dan naar sectie 4.2 en volg de stappen om ervoor te zorgen dat de nieuwste firmware wordt bijgewerkt.

Nadat u de juiste stappen heeft gevolgd, worden het HuskyLens Uploader USB to UART-stuurprogramma en de nieuwste firmware geïnstalleerd.

#### 3.4.3.3 Overzicht van de kaart

- Functieknop
- Leerknop
- Scherm



De HuskyLens is uitgerust met twee knoppen: de functieknop en de leerknop. Hun primaire functies worden als volgt beschreven:

- Gebruik de functieknop om tussen verschillende functies te schakelen door deze naar links of rechts te draaien.
- Druk kort op de leerknop om het apparaat over een specifiek object te leren. Door de leerknop ingedrukt te houden, kunt u het object continu leren vanuit verschillende hoeken en afstanden. Als de HuskyLens het object al heeft geleerd, zorgt een korte druk op de leerknop ervoor dat hij het vergeet.
- Om toegang te krijgen tot het menu op het tweede niveau (parameterinstellingen) binnen de huidige functie, houdt u de functieknop ingedrukt. Draai de knop naar links of rechts of druk kort in om de bijbehorende parameters aan te passen.

#### HuskyLens-mogelijkheden:

- Gezichtsherkenning
- Object volgen
- Object herkenning
- Lijn volgen
- Kleurherkenning
- Tagherkenning
- Objectclassificatie

#### Nieuwste softwareversie:

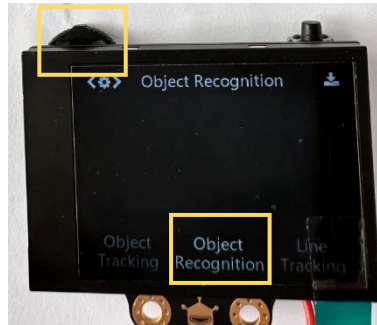
- Navigeer naar Algemene instellingen -> Versie
- Controleer het volgende adres voor instructies over hoe u kunt updaten naar de nieuwste versie (als deze bestaat):

[https://wiki.dfrobot.com/HUSKYLENS\\_V1.0\\_SKU\\_SEN0305\\_SEN0336](https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336)

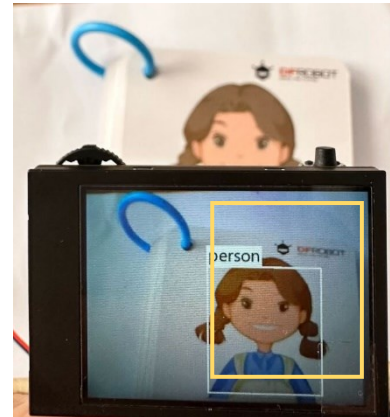
#### 3.4.3.4 Aan de slag

micro:bit en HuskyLens is aangelegd, moet de micro:bit via een kabel met de pc worden verbonden. Zodra de verbinding tot stand is gebracht, worden de HuskyLens-camera en micro:bit ingeschakeld.

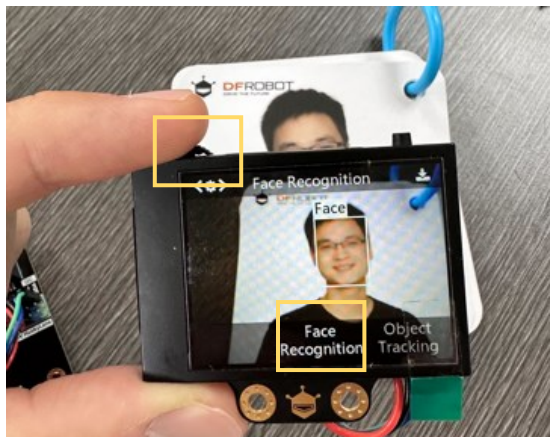
Vervolgens moet de leerling naar het gedeelte Gezichtsherkenning navigeren met behulp van de knop die zich in de linkerbovenhoek van de HuskyLens-camera bevindt. Verplaats het naar rechts totdat je het gedeelte Gezichtsherkenning vindt.



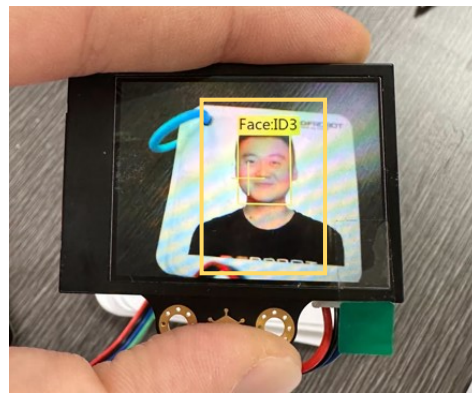
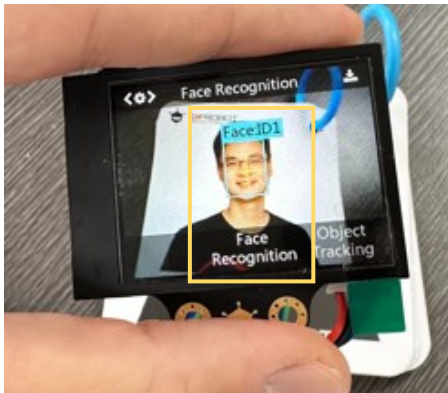
Nadat Gezichtsherkenning is geselecteerd, richt u de camera op een object of persoon om het resultaat te zien. De HuskyLens-camera kan objecten detecteren door middel van live-invoer en kan de gebruiker een uitkomst geven. Dit vertegenwoordigt de perceptie van dit systeem.



Bij de instelling Gezichtsherkenning heeft u de mogelijkheid om meer dan 1 gezicht op te slaan. Selecteer eerst de gezichtsherkenningsoptie en houd de functieknop lang ingedrukt, zoals hieronder weergegeven.



Klik vervolgens op het Leerveelvoud en vervolgens op Opslaan en terugkeren. Nu is de functie voor het leren van meerdere gezichten geactiveerd en kunnen meerdere gezichten worden gescand en opgeslagen op de HuskyLens-camera.

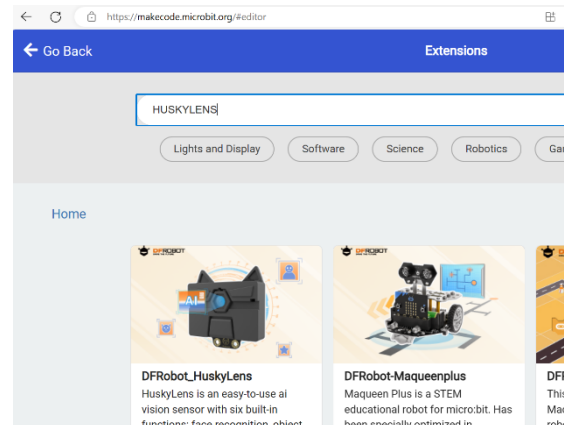
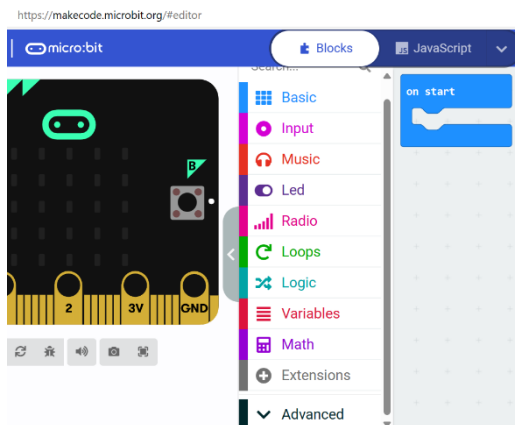


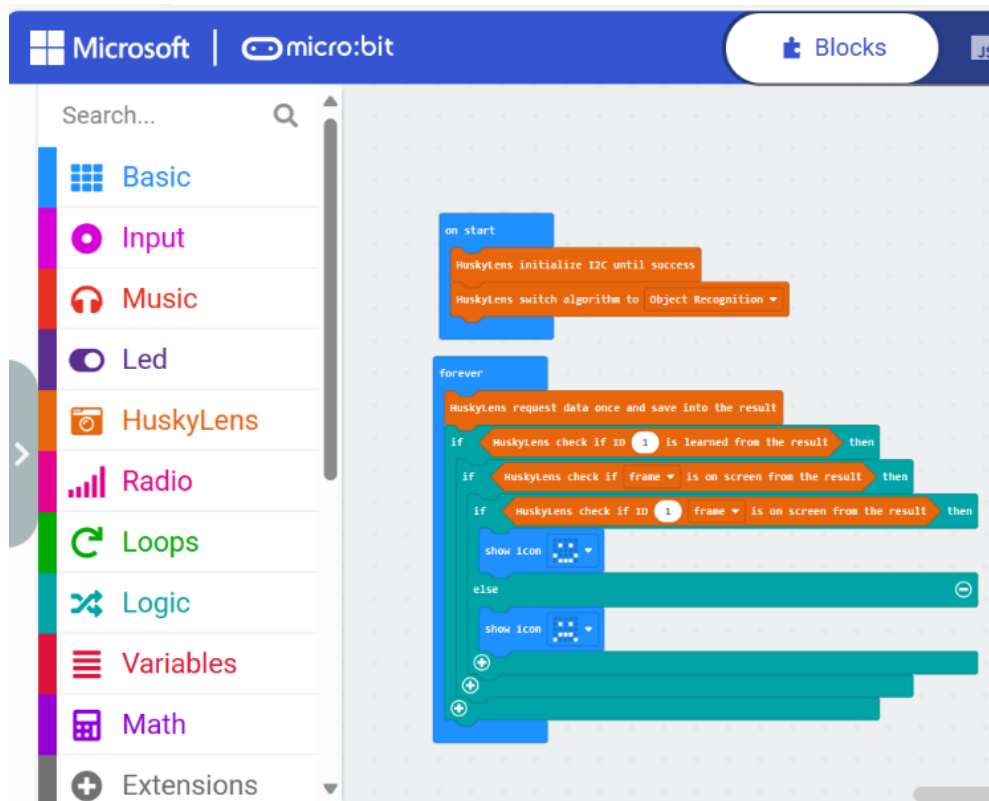
### 3.4.3.5 Code

Na het opzetten van de bedrading tussen de micro:bit en HuskyLens moeten leerlingen doorgaan met het schrijven van code die instructies geeft aan de HuskyLens-camera, waardoor een objectherkenningssysteem kan worden gemaakt. Deze code begeleidt de camera bij het vastleggen en verwerken van objecten om ze effectief te herkennen.

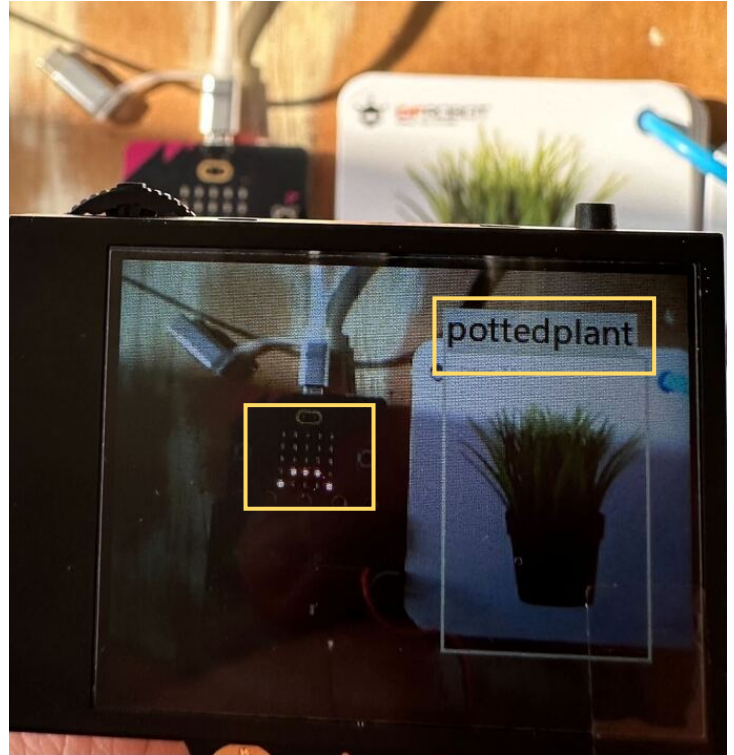
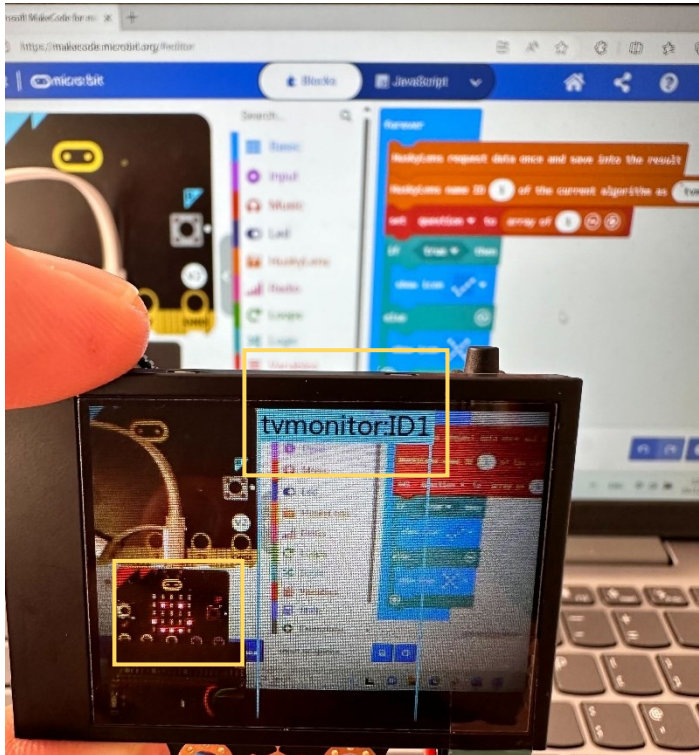
De student moet de software op de volgende website gebruiken om de code te schrijven: <https://makecode.microbit.org/>

Om de onderstaande code te gebruiken, moet een HuskyLens-blok worden toegevoegd. Om dit blok toe te voegen, selecteert u het blok "Extensies" en zoekt u vervolgens naar **DFRobot\_HuskyLens** en selecteert u het. Vervolgens wordt het HuskyLens-blok automatisch toegevoegd aan de bloksectie en kunt u vervolgens doorgaan met de onderstaande code.





## Resultaat :



## Code-uitleg :

- De HuskyLens-camera die het I2C-protocol gebruikt. Het bereidt de micro:bit voor om met de camera te praten.
- Om de HuskyLens-functie te gebruiken, voegt u eerst de HuskyLens-extensie toe aan uw codepalet. Met deze extensie kunt u code maken die het algoritme naadloos overschakelt naar de gezichtsherkenningsmodus.
- We vertellen de HuskyLens dat hij in de objectherkenningsmodus moet werken. Dit betekent dat het zal proberen objecten te herkennen en te onderscheiden.
- Het volgende deel van de code zorgt voor een continue lus die blijft draaien zolang de micro:bit is ingeschakeld.
- Vervolgens vraagt de micro:bit de HuskyLens om informatie te verstrekken over wat hij momenteel ziet.
- Het programma controleert of de HuskyLens objecten heeft geleerd. Het cijfer 1 verwijst naar het label van het object, en als er iets is geleerd, wordt de code tussen de accolades uitgevoerd.
- De HuskyLens detecteert een object in zijn gezichtsveld.

- Vervolgens wordt gecontroleerd of een specifiek object (in dit geval object 1) is gedetecteerd.
- Als de HuskyLens het object herkent en het als een blok verschijnt, geeft de micro:bit een blij gezichtpictogram weer.
- Als het object niet als blok wordt herkend, wordt er een droevig gezichtpictogram weergegeven.

### 3.4.4 Experiment 1

In deze oefening krijgen leerlingen de kans om de technologie van HuskyLens en micro:bit toe te passen om **verschillende vormen te herkennen**. Deze oefening daagt hun begrip van objectherkenning uit en stelt hen in staat de praktische toepassingen van deze technologie te verkennen.

Het primaire doel van deze oefening is dat leerlingen een programma maken dat de HuskyLens-camera gebruikt om verschillende vormen, zoals een cirkel, driehoek, rechthoek, te identificeren en te onderscheiden en vervolgens de bijbehorende emoji's of symbolen op het display van de micro:bit weer te geven.

## 3.5 Activiteit 2: Introductie van het idee van representatie en redeneren

### 3.5.1 Beschrijving

In deze activiteit verkennen studenten de wereld van kunstmatige intelligentie en computervisie om de gezichtsherkenningmethode te verkennen. Deze activiteit introduceert het idee van **het trainen** van een AI-model om specifieke taken uit te voeren of intelligente beslissingen te nemen. De trainingsfase is cruciaal voor een AI-model, omdat het tijdens deze fase is dat het model **leert** en zich aanpast aan de taak of het probleem dat het moet oplossen. Training is in wezen het proces waarbij het AI-model wordt aangeleerd door het bloot te stellen aan een grote dataset, waardoor het model patronen, correlaties en regels kan leren uit de verstrekte informatie.

Bij deze activiteit wordt een HuskyLens-camera geïnstalleerd waarmee het AI-model rechtstreeks informatie kan verzamelen. Tijdens het trainingsproces past het AI-model zijn interne parameters aan door middel van iteratieve optimalisatie, waardoor het beter in staat is nauwkeurige voorspellingen of classificaties te maken.

Het micro:bit-platform, dat fungeert als het brein van de operatie, wordt de interface waarmee studenten het getrainde AI-model kunnen observeren en ermee kunnen communiceren. Dit praktijkgerichte aspect van de activiteit stelt studenten in staat niet alleen de theoretische onderbouwing van machinaal leren te begrijpen, maar ook de praktische implicaties van het inzetten van dergelijke technologie in scenario's in de echte wereld te waarderen.

Met behulp van de HuskyLens-camera en het micro:bit-platform ontdekken studenten hoe technologie kan worden gebruikt om het AI-model te trainen met bekende menselijke gezichten. Aan het einde van de activiteit begrijpen de leerlingen niet alleen de grondbeginselen van gezichts- en objectherkenning en AI-training, maar krijgen ze ook een bredere waardering voor het potentieel en de uitdagingen van het integreren van AI in het dagelijks leven.

### 3.5.2 Hardware

- Micro:bit
- IO-extender voor micro:bit
- HuskyLens
- Objectafbeeldingen



### 3.5.3 Opgericht

#### 3.5.3.1 Bedrading

- Verbind de micro:bit met de IO-extender voor micro:bit.
- Gebruik vervolgens de volgende tabel om de IO-extender voor micro:bit te verbinden met de HuskyLens-camera.

IO-extender voor micro:bit-POORT	HuskyLens-POORT
SDA	T
SCL	R
GND	-
3V3	+

#### 3.5.3.2 Aan de slag

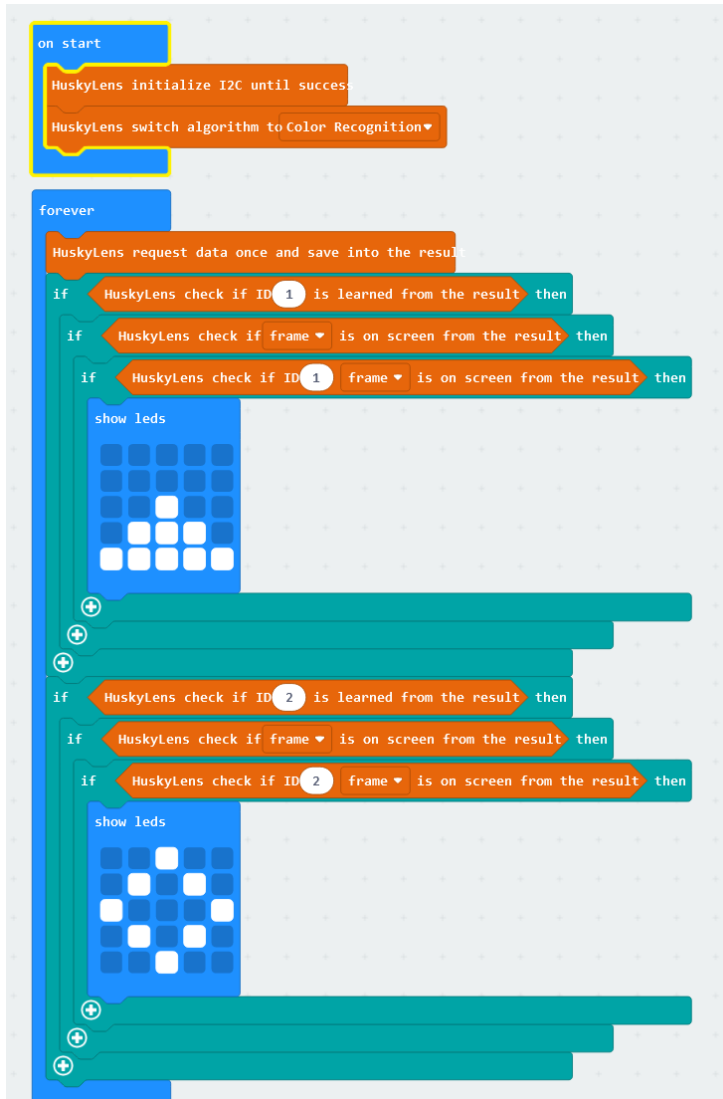
Nadat de bedrading tussen de micro:bit en HuskyLens is aangelegd, moet de micro:bit via een kabel met de pc worden verbonden. Zodra de verbinding tot stand is gebracht, worden de HuskyLens-camera en micro:bit ingeschakeld.

Vervolgens moet de leerling naar het gedeelte Gezichtsherkenning navigeren met behulp van de knop die zich in de linkerbovenhoek van de HuskyLens-camera bevindt. Verplaats het naar rechts totdat je het gedeelte Gezichtsherkenning vindt.



#### 3.5.3.3 Code

Na het opzetten van de bedrading tussen de micro:bit en HuskyLens moeten leerlingen doorgaan met het schrijven van code die instructies geeft aan de HuskyLens-camera, waardoor een kleurherkenningssysteem kan worden gemaakt. Deze code begeleidt de camera bij het vastleggen en verwerken van kleuren om vormen effectief te herkennen.

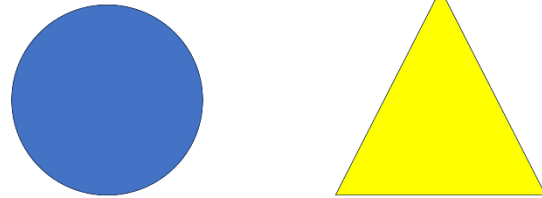
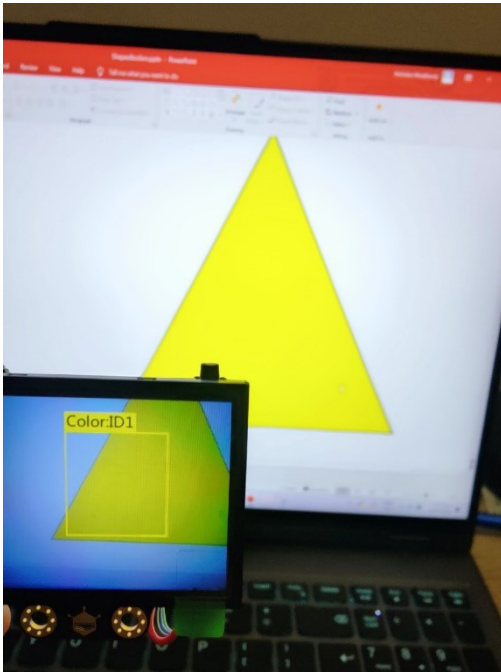


### Code-uitleg :

Om de HuskyLens-functie te gebruiken, voegt u eerst de HuskyLens-extensie toe aan uw codeerpalet. Met deze extensie kunt u code maken die het algoritme naadloos overschakelt naar de kleurherkenningsmodus.

Initialiseert de I2C-communicatie tussen de microcontroller en de HuskyLens. Het blijft proberen totdat de initialisatie succesvol is. Stelt de HuskyLens in om het kleurherkenningsalgoritme te gebruiken. Vraagt eenmalig gegevens op bij de HuskyLens en slaat deze op in het resultaat. Zorgt ervoor dat de kleur met ID 1 wordt geleerd door de HuskyLens. Als kleur-ID 1 wordt herkend, verschijnt er een **driehoekig** LED-patroon op het scherm van de microcontroller.

Zorgt ervoor dat de kleur met ID 2 wordt geleerd door de HuskyLens. Als kleur-ID 2 wordt herkend, geeft u een **cirkelvormig** LED-patroon weer op het scherm van de microcontroller.



### 3.5.4 Oefening: Train het AI-model om verschillende vormen te herkennen

In deze oefening krijgen leerlingen de kans om de technologie van HuskyLens en micro:bit toe te passen om **vormen te herkennen**. Deze oefening daagt hun begrip van de trainingsfase van het AI-model uit.

## 3. 6 Activiteit 3: Introductie van het idee van leren door een model voor gezichtsherkenning te trainen

### 3.6.1 Beschrijving

In deze activiteit verdiepen leerlingen zich in de dynamische fase die volgt op de trainingsfase van het AI-model: de **leerfase**. Deze fase markeert de praktische toepassing van de verworven kennis van het model en de beoordeling van het vermogen ervan om nauwkeurige resultaten te bieden in realtime scenario's, met name in de context van gezichtsherkenning.

De leerfase omvat het testen van het getrainde AI-model, het evalueren van de prestaties ervan en het garanderen dat het de bekende gezichten waarop het is getraind correct kan identificeren en erop kan reageren. Het belangrijkste doel is om te bepalen of het model in staat is de juiste uitkomst te leveren, een cruciale stap in de bevestiging dat het AI-model functioneert zoals bedoeld.

Het micro:bit-platform dient als interface waarmee studenten communiceren met het AI-model. Als gebruikers hebben de leerlingen de taak om te controleren of het model het getrainde gezicht dat door de HuskyLens-camera is vastgelegd, correct herkent. Dit praktijkgerichte aspect van de activiteit biedt een tastbare en praktische demonstratie van hoe AI-modellen overgaan van theoretisch leren naar toepassing in de echte wereld.

Het proces omvat het vastleggen van afbeeldingen met behulp van de HuskyLens-camera en het toestaan dat het AI-model de gezichten in de afbeeldingen analyseert en identificeert. De micro:bit dient dan als feedbackmechanisme en geeft de uitkomst van het herkenningsproces weer. Als het AI-model het getrainde gezicht met succes identificeert, dient dit als een indicator dat de leerfase effectief is geweest en dat het model goed werkt.

Deze activiteit versterkt niet alleen de technische concepten met betrekking tot gezichtsherkenning en AI, maar benadrukt ook het belang van validatie en testen in de echte wereld bij de ontwikkeling en inzet van AI-systemen. Studenten krijgen inzicht in de uitdagingen en overwegingen die betrokken zijn bij het waarborgen van de betrouwbaarheid en nauwkeurigheid van AI-modellen in praktische toepassingen.

### 3.6.2 Hardware

- Micro:bit
- IO-extender voor micro:bit
- HuskyLens
- Kleurenafbeeldingen

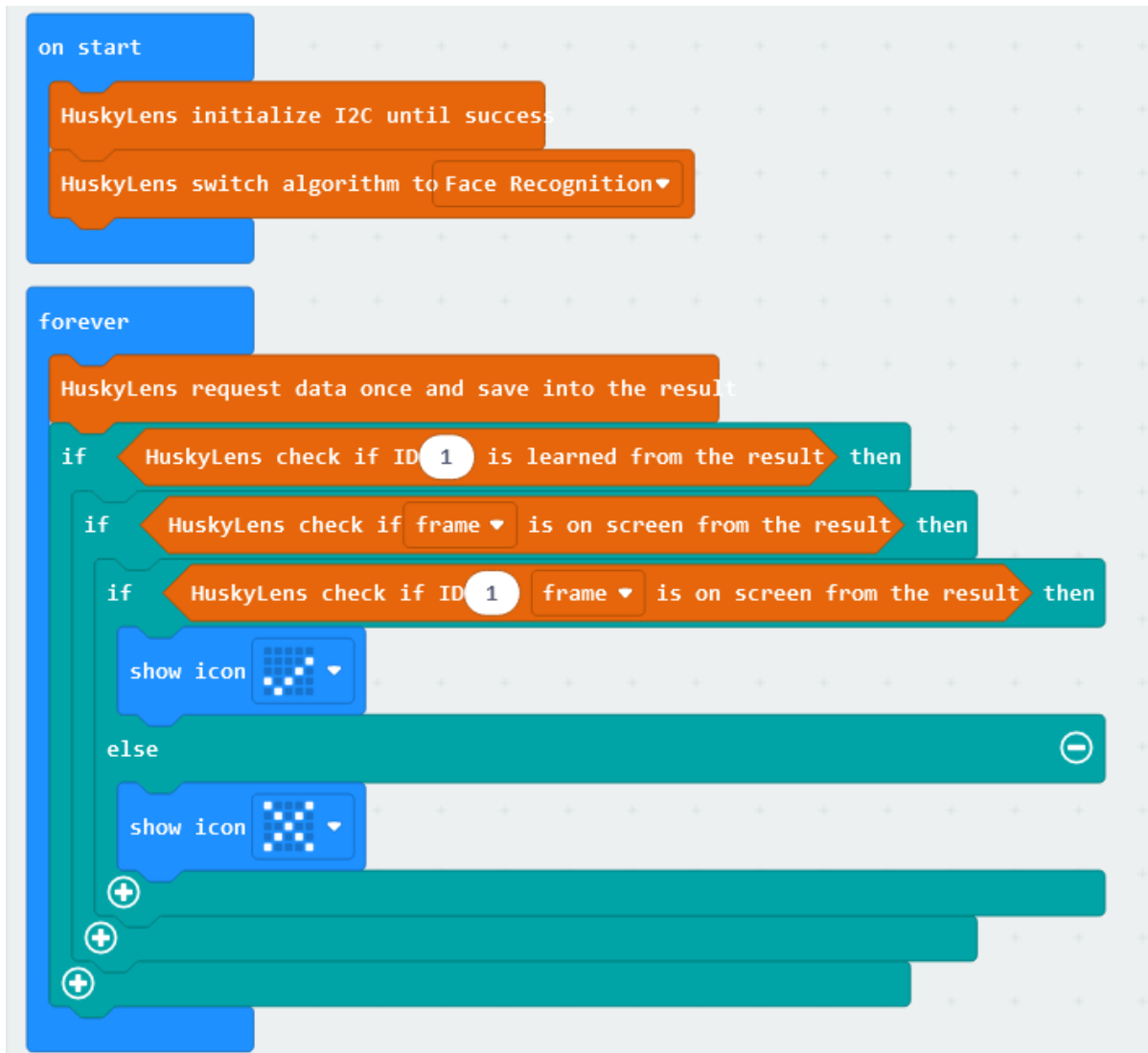


### 3.6.3 Installatie

- Verbind de micro:bit met de IO-extender voor micro:bit.
- Gebruik vervolgens de volgende tabel om de IO-extender voor micro:bit te verbinden met de HuskyLens-camera.

IO-extender voor micro:bit-POORT	HuskyLens- POORT
SDA	T
SCL	R
GND	-
3V3	+

### 3.6.4 Code



#### Code-uitleg :

Om de HuskyLens-functie te gebruiken, voegt u eerst de HuskyLens-extensie toe aan uw codeerpalet. Met deze extensie kunt u code maken die het algoritme naadloos overschakelt naar de gezichtsherkenningsmodus.

Deze code vraagt om invoergegevens, waardoor de leerling de HuskyLens-camera moet gebruiken om het gezicht van een persoon vast te leggen. De gebruiker heeft het AI-model met een specifiek gezicht getraind op de vorige activiteit.

Als de camera het herkende gezicht detecteert, zal de micro:bit reageren met een vrolijk lachend gezicht of een correct antwoordsignaal als uitvoer, terwijl een willekeurig of niet-herkend gezicht zal resulteren in een droevig gezicht of een verkeerd signaal. Dit boeiende

proces overbruggt de kloof tussen technologie en menselijke interactie, waardoor het een fascinerende leerervaring wordt.

- De HuskyLens-camera die het I2C-protocol gebruikt. Het bereidt de micro:bit voor om met de camera te praten.
- We vertellen de HuskyLens om in de kleurherkenningsmodus te werken. Dit betekent dat het objecten zal proberen te herkennen en te onderscheiden op basis van hun kleuren.
- Het volgende deel van de code zorgt voor een continue lus die blijft lopen zolang de micro:bit is ingeschakeld.
- Vervolgens vraagt de micro:bit de HuskyLens om informatie te verstrekken over wat hij momenteel ziet.
- Het programma controleert of de HuskyLens objecten heeft geleerd. Het cijfer 1 verwijst naar het label van het object, en als er iets is geleerd, wordt de code tussen de accolades uitgevoerd.
- De HuskyLens detecteert een object in zijn gezichtsveld.
- Vervolgens wordt gecontroleerd of een specifiek object (in dit geval object 1) is gedetecteerd.
- Als de HuskyLens het object herkent en het verschijnt als een blok, geeft de micro:bit een blij gezichtpictogram weer.
- Als het object niet als blok wordt herkend, wordt er een droevig gezichtpictogram weergegeven.



### 3.6.5 Oefening : Test of het AI-model u herkent

Het HuskyLens- en micro:bit-systeem gebruiken om het AI-model te trainen met een nieuw gezicht met de naam van de persoon erop en vervolgens het gezicht van de nieuwe persoon

vast te leggen met de HuskyLens-camera om te zien of het het kan herkennen en of het het juiste label erop heeft persoon.

Deze oefening zal de gebruiker testen of hij het AI-model kan trainen en testen.

### 3. 7 Activiteit 4: Introductie van het idee van natuurlijke interactie door een getraind model te integreren in een AI-toepassing

#### 3.7.1 Beschrijving

In deze boeiende voortzetting van het AI-verkenningstraject gaan leerlingen door naar de **testfase**, een cruciale stap die volgt op de leerfase. De testfase dient als een uitgebreide beoordeling van de prestaties van het AI-model, waarbij het vermogen wordt geëvalueerd om te generaliseren en nauwkeurig te reageren op verschillende gevallen. Door de lens van gezichtsherkenning met behulp van de HuskyLens-camera en het micro:bit-platform zullen studenten zich verdiepen in de fijne kneepjes van de testfase.

De testfase is de stap na de leerfase. In deze fase wordt het AI-model in meerdere cases getest om te kijken of het goed werkt. Na deze fase kunnen passende aanpassingen worden gedaan. De testfase is een essentieel controlepunt bij de ontwikkeling van een AI-model, omdat het scenario's uit de echte wereld simuleert en de aanpasbaarheid en betrouwbaarheid van het model beoordeelt. Na de leerfase, waarin het model vertrouwd is geraakt met specifieke gezichten, introduceert de testfase variabiliteit door zowel bekende als onbekende menselijke gezichten vast te leggen.

Bij deze activiteit leggen leerlingen meerdere gezichten vast met de HuskyLens-camera, inclusief gezichten waarop het AI-model is getraind (bekende gezichten) en andere gezichten die het niet is tegengekomen tijdens de trainings- en leerfasen (onbekende gezichten). De micro:bit geeft vervolgens voor elk vastgelegd gezicht de uitkomst van het herkenningsproces weer.

Met behulp van de HuskyLens-camera en het micro:bit-platform ontdekken studenten hoe technologie kan worden gebruikt om het AI-model te testen met bekende en onbekende menselijke gezichten.

### 3.7.2 Hardware

- Micro:bit
- IO-extender voor micro:bit
- HuskyLens
- Kleurenafbeeldingen



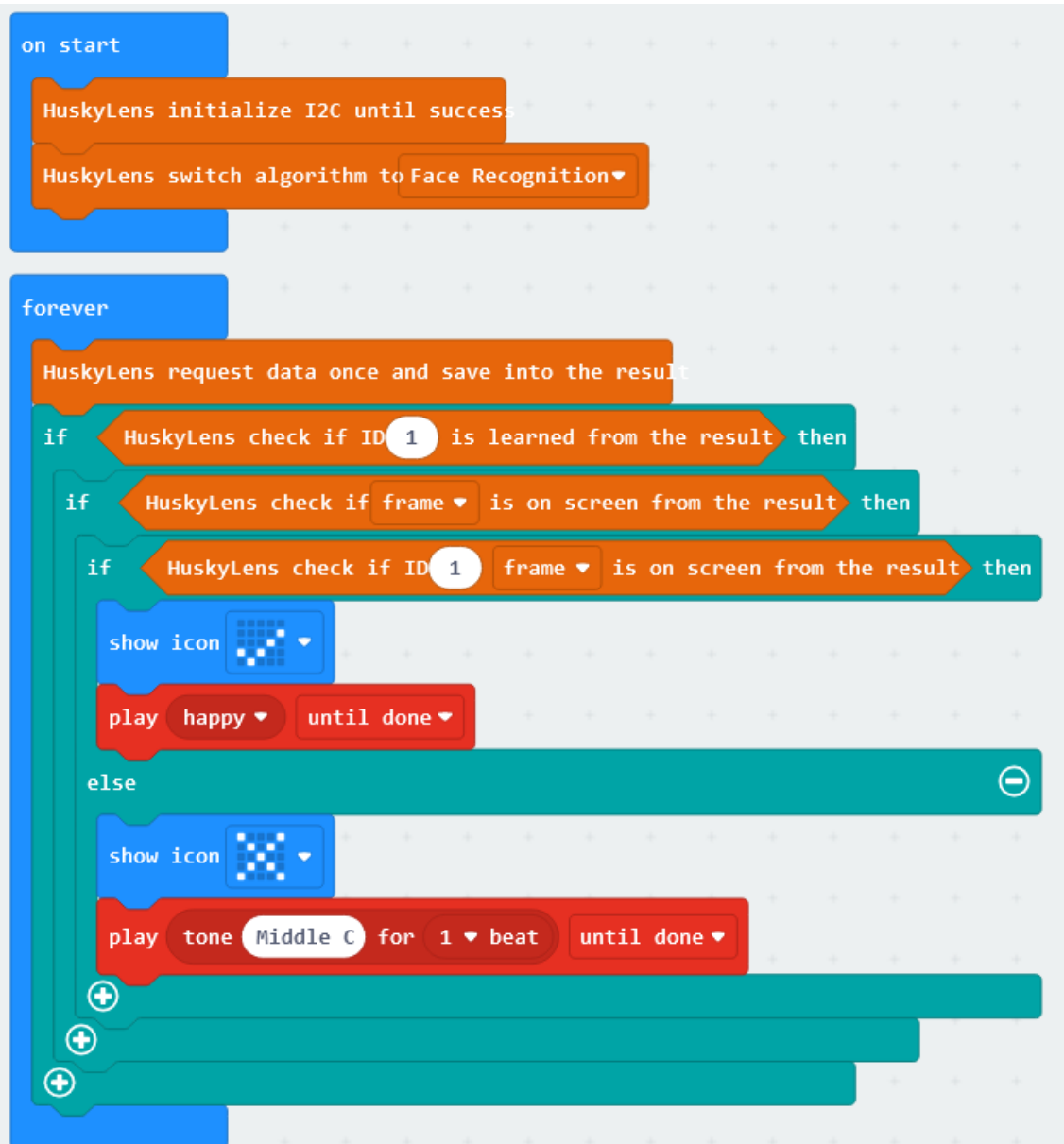
### 3. 7.3 Installatie

- Verbind de micro:bit met de IO-extender voor micro:bit.
- Gebruik vervolgens de volgende tabel om de IO-extender voor micro:bit te verbinden met de HuskyLens-camera.

IO-extender voor micro:bit-POORT	HuskyLens-POORT
SDA	T
SCL	R
GND	-
3V3	+

#### 3.7.3.1 Code

Na het opzetten van de bedrading tussen de micro:bit en HuskyLens moeten leerlingen doorgaan met het schrijven van code die instructies geeft aan de HuskyLens-camera, waardoor een gezichtsherkenningssysteem kan worden gemaakt. Deze code begeleidt de camera bij het vastleggen en verwerken van gezichten om ze effectief te herkennen.



De

student moet de software op de volgende website gebruiken om de code te schrijven:

<https://makecode.microbit.org/>

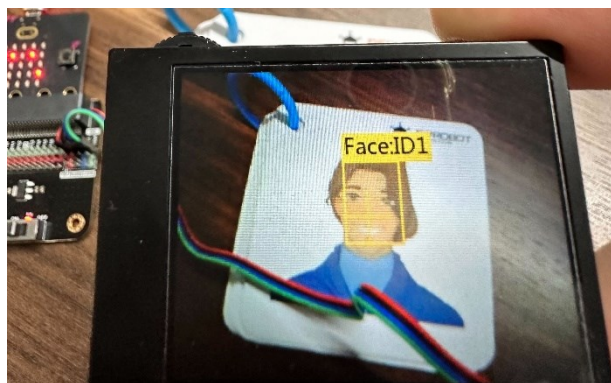
### Code-uitleg :

Om de HuskyLens-functie te gebruiken, voegt u eerst de HuskyLens-extensie toe aan uw codeerpalet. Met deze extensie kunt u code maken die het algoritme naadloos overschakelt naar de gezichtsherkenningsmodus.

Deze code vraagt om invoergegevens, waardoor de leerling de HuskyLens-camera moet gebruiken om het gezicht van een persoon vast te leggen. De gebruiker heeft het AI-model met een specifiek gezicht getraind op de vorige activiteit.

Als de camera het herkende gezicht detecteert, zal de micro:bit reageren met een vrolijk smileygezicht als uitvoer en een vrolijk geluid, terwijl een willekeurig of niet-herkend gezicht zal resulteren in een droevig gezicht en een droevig geluid. Dit boeiende proces overbrugt de kloof tussen technologie en menselijke interactie, waardoor het een fascinerende leerervaring wordt.

- De HuskyLens-camera die het I2C-protocol gebruikt. Het bereidt de micro:bit voor om met de camera te praten.
- We vertellen de HuskyLens om in de kleurherkenningsmodus te werken. Dit betekent dat het objecten zal proberen te herkennen en te onderscheiden op basis van hun kleuren.
- Het volgende deel van de code zorgt voor een continue lus die blijft lopen zolang de micro:bit is ingeschakeld.
- Vervolgens vraagt de micro:bit de HuskyLens om informatie te verstrekken over wat hij momenteel ziet.
- Het programma controleert of de HuskyLens objecten heeft geleerd. Het cijfer 1 verwijst naar het label van het object, en als er iets is geleerd, wordt de code tussen de accolades uitgevoerd.
- De HuskyLens detecteert een object in zijn gezichtsveld.
- Vervolgens wordt gecontroleerd of een specifiek object (in dit geval object 1) is gedetecteerd.
- Als de HuskyLens het object herkent en het verschijnt als een blok, geeft de micro:bit een blij gezichtpictogram weer.
- Als het object niet als blok wordt herkend, wordt er een droevig gezichtpictogram weergegeven.



### 3.7.4 Oefening: Alarmsysteem

Gebruik het HuskyLens- en micro:bit-systeem om het AI-model te trainen met een nieuw gezicht met de naam van de persoon erop en leg vervolgens het gezicht van de nieuwe persoon vast met de HuskyLens-camera om te zien of deze het kan herkennen en of de camera een persoon detecteert die dat wel is niet in het systeem om geluid te maken. Voor deze oefening maakt de gebruiker een alarmsysteem aan.

## 3.8 Activiteit 5: Introductie van het idee van maatschappelijke impact

### 3.8.1 Beschrijving

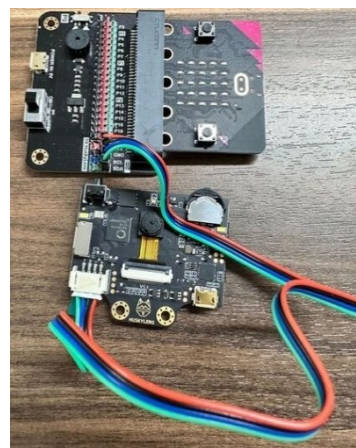
Deze les introduceert het concept van maatschappelijke impact door een model voor meervoudige gezichtsherkenning te trainen. Studenten zullen onderzoeken hoe ze gezichtsgegevens kunnen coderen en verwerken om een model te trainen dat meerdere gezichten tegelijkertijd kan herkennen.

Ze leren hoe ze deze technologie kunnen implementeren en bespreken de maatschappelijke implicaties van AI-systemen die gezichten in verschillende situaties kunnen detecteren en identificeren. Het vermogen van AI om gezichtsherkenning uit te voeren heeft aanzienlijke maatschappelijke gevolgen, waaronder verbeterde beveiliging, privacykwesaties en ethische overwegingen met betrekking tot surveillance en datagebruik.

Deze les is bedoeld om leerlingen een uitgebreid inzicht te geven in zowel de technische als de maatschappelijke aspecten van gezichtsherkenningstechnologie.

### 3.8.2 Hardware

- Micro:bit
- IO-extender voor micro:bit
- HuskyLens
- Afbeeldingen met gezichten



### 3. 8.3 Installatie

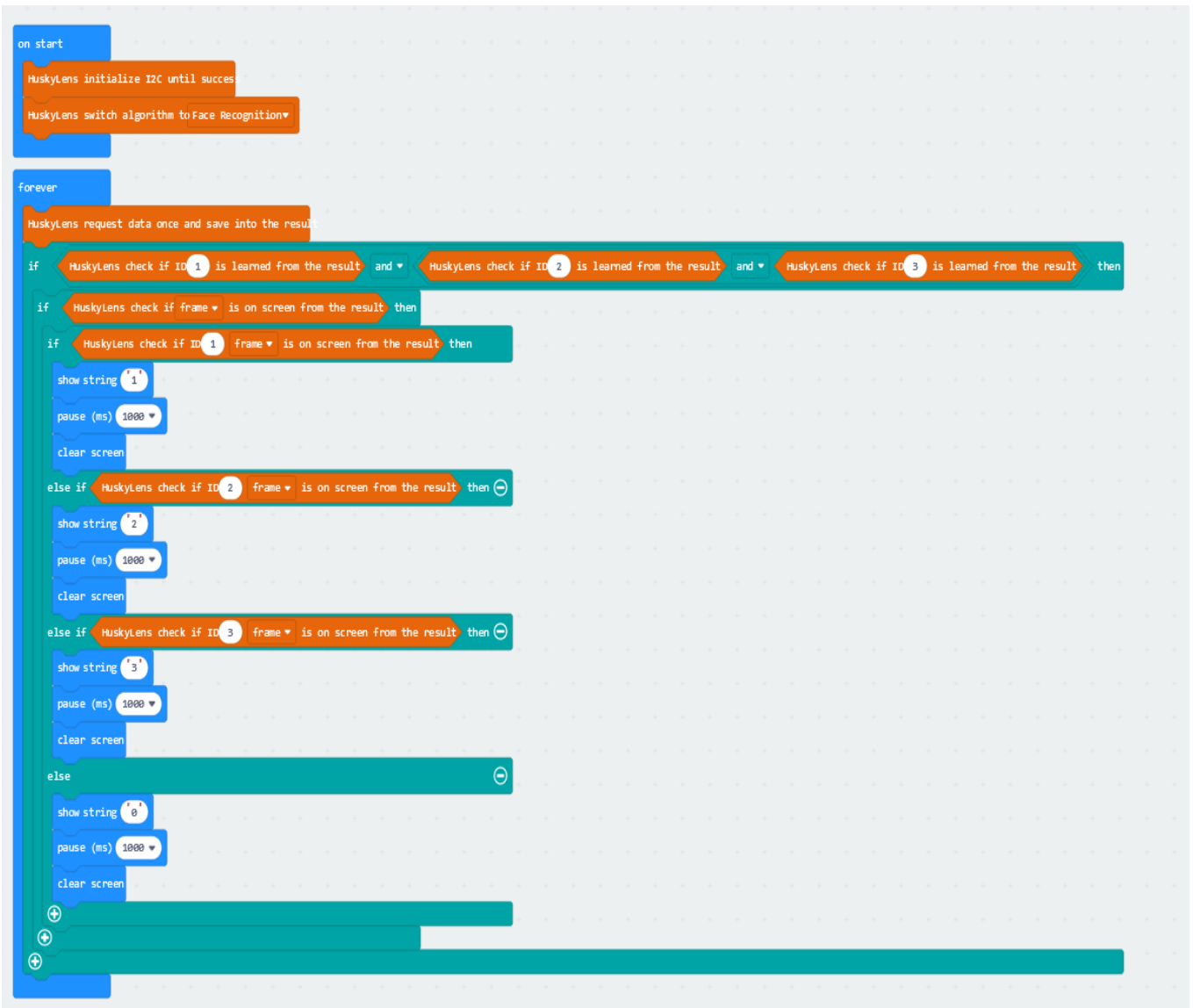
- Verbind de micro:bit met de IO-extender voor micro:bit.
- Gebruik vervolgens de volgende tabel om de IO-extender voor micro:bit te verbinden met de HuskyLens-camera.

IO-extender voor micro:bit- POORT	HuskyLens- POORT
SDA	T
SCL	R
GND	-
3V3	+

#### 3.8.3.1 Code

Na het opzetten van de bedrading tussen de micro:bit en HuskyLens moeten leerlingen doorgaan met het schrijven van code die instructies geeft aan de HuskyLens-camera, waardoor een gezichtsherkenningssysteem kan worden gemaakt. Deze code begeleidt de camera bij het vastleggen en verwerken van gezichten om ze effectief te herkennen. Het is belangrijk dat u bij deze activiteit de instelling voor het maken van meerdere afbeeldingen gebruikt om de volgende activiteit te creëren.

De student moet de software op de volgende website gebruiken om de code te schrijven:  
<https://makecode.microbit.org/>



### Code-uitleg :

Om de HuskyLens-functie te gebruiken, voegt u eerst de HuskyLens-extensie toe aan uw codeerpalet. Met deze extensie kunt u code maken die het algoritme naadloos overschakelt naar de gezichtsherkenningsmodus.

Initialiseert de I2C-communicatie tussen de microcontroller en de HuskyLens. Het blijft proberen totdat de initialisatie succesvol is. Stelt de HuskyLens in om het gezichtsherkenningssysteem te gebruiken. Vraagt eenmalig gegevens op bij de HuskyLens en slaat deze op in het resultaat. Zorgt ervoor dat gezichten met ID's 1, 2 en 3 door de HuskyLens worden geleerd.

De code draait continu in een lus, vraagt gegevens op van de HuskyLens en controleert of een van de drie gezichten wordt herkend. Afhankelijk van welk gezicht wordt gedetecteerd, wordt het overeenkomstige nummer (1, 2 of 3) weergegeven op het scherm van de microcontroller. Als er geen gezicht wordt herkend, wordt "0" weergegeven. Dit levert real-time feedback op basis van gezichtsherkenning, waardoor de interactie boeiend en informatief wordt voor de gebruiker.

### 3.8.4 Het idee van natuurlijke interactie introduceren door een model voor gezichtsherkenning te trainen

Studenten trainen de HuskyLens met een gezicht en schrijven code om het te detecteren en een geluid af te spelen.

Voor deze activiteit is het belangrijk om de volgende stappen te overwegen: Gebruik de optie Gezichtsherkenning, druk lang op de functieknoop in Gezichtsherkenning, selecteer Meerdere en neem 3 gezichten op.

## 3.9 Aanvullend materiaal en hulpmiddelen

Type bron	Titel	Onderwerp	Koppeling
<b>software</b>	Maakcode	Blokcodering	<a href="https://makecode.microbit.org/">https://makecode.microbit.org/</a>
<b>Artikel</b>	Gebruik van Microbit en Huskylens	Hoe Huskylens te gebruiken	<a href="https://www.instructables.com/Microbit-Visual-Object-Tracking-With-Huskylens/">https://www.instructables.com/Microbit-Visual-Object-Tracking-With-Huskylens/</a>