Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

AI in Vision

Introducing the 5 Big Ideas in Artificial Intelligence using Internet of Things in STEM education

T2.4 IoT Projects Design & Resources Development

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

AI4STEM IoT Projects Design & Resources Development Project: AI in Vision

Copyright

© Copyright the AI4STEM Consortium
2022-1-FR01-KA220-SCH-000085611
All rights reserved.

AI4STEM IoT Projects Design & Resources Development Project: AI in Vision © 2023 by AI4STEM CONSORTIUM is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Table of Contents

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

1.Introduction to the Project

For this project, micro:bit and HuskyLens will be used to complete different activities. HuskyLens is a smart vision sensor or camera module that combines artificial intelligence (AI) and computer vision technology to perform various tasks like object recognition, face recognition, tracking, color recognition and more. Integrating artificial intelligence (AI) with the HuskyLens camera and micro:bit is an exciting project that allows you to create interactive and intelligent applications.

1.1 The scope of the Project

The scope of the project is to use micro:bit and HuskyLens to create an Artificial Intelligence model. The model will be trained through HuskyLens camera and it will give an output through the micro:bit tool.

1.2 The target groups

The project is targeted mainly to the direct involvement of educators, mainly of upper primary and secondary education.

1.3 The purpose of this document

The goal of this document is to demonstrate how to train and test an AI model using the HuskyLens camera and display the output on the micro:bit.

2. Glossary of the Unit

Word	Definition
Micro:bit	The micro:bit is a pocket-sized, programmable computer board designed for education. It features an LED matrix, various sensors, and a microcontroller
HuskyLens	The HuskyLens is a vision sensor module that can recognize objects, faces, and gestures. It's equipped with a camera and onboard processing power to perform AI-related tasks

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3. Introduction to the "AI in Vision"

3.1 Description

This unit will introduce learners to the process of training and testing an AI model with the HuskyLens camera. They will learn how to set up and configure the HuskyLens camera for image recognition tasks, train the AI model to recognize specific objects or patterns, and finally, interface the camera with a micro:bit to display the AI model's output. Throughout the unit, learners will gain a practical understanding of computer vision, machine learning, and how to integrate AI technologies into real-world projects.

3.2 Learning objectives & outcomes

On successful completion of this unit, learners should be able to:

- Understanding of Computer Vision
- AI Model Training
- HuskyLens Configuration
- Micro:bit Integration
- Problem-Solving
- Practical Application
- Operate HuskyLens
- Collect Data
- Train AI Models
- Interact with Micro:bit

3.3 Estimated duration of the Unit

This is a rather extended project needing several hours to properly address all the included aspects. The following duration is indicative and might vary based on your students' age and level.

Activity 1: 90 minutes

Activity 2: 90 minutes

Activity 3: 45 minutes

Activity 4: 45 minutes

Activity 5: 45 minutes

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.4 Activity 1 - Introducing the Big Idea of Perception through IoT:

3.4.1 Description

In this lesson, students will explore the world of artificial intelligence and computer vision in order to explore perception method, through object live tracking, a feature made possible by the cutting-edge capabilities of the HuskyLens camera and the micro:bit platform.

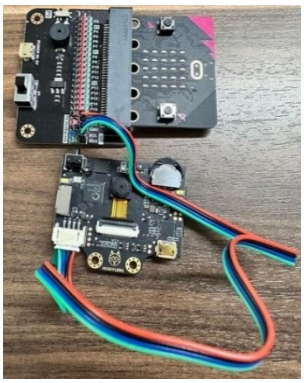
Using the HuskyLens camera and micro:bit platform, students will discover how technology can be used to get live input and gives an output to the user. The live tracking aspect not only demonstrates the real-time capabilities of the system but also emphasizes the dynamic nature of technology in responding to its environment.

In the era of rapid technological advancements, the fusion of Internet of Things (IoT) and cutting-edge image processing technologies has paved the way for groundbreaking applications. One such innovation is the integration of the HuskyLens, a powerful smart camera, with Micro:bit. Together, they form a dynamic system for real-time object tracking, revolutionizing the way we perceive and interact with our surroundings.

As students engage with the HuskyLens and Micro:bit system, they are not only gaining hands-on experience with cutting-edge tools but are also participating in the ongoing narrative of technological progress.

3.4.2 Hardware

- Micro:bit microcontroller
- IO Extender for micro:bit
- HuskyLens



3.4.3 Setup

3.4.3.1 Wiring

- Connect the micro:bit with the IO extender for micro:bit.
- Then use the following table to connect the IO extender for micro:bit with the HuskyLens camera.

IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.4.3.2 Updating Firmware

Before utilizing this product, it is highly advised to install/update the firmware of the HuskyLens to access the newest features and ensure optimal stability. It is suggested to utilize the HuskyLens Uploader on Windows 10 for firmware upload as it offers a graphical user interface (GUI) and user-friendly operation.

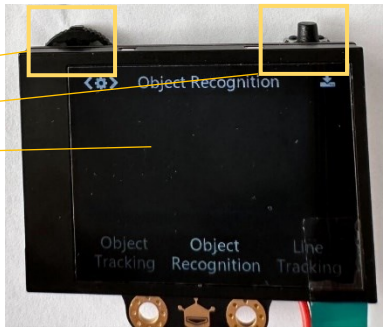
Use the following link to navigate to the DFROBOT website to find the link to install the firmware: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336

After visiting the above-mentioned website, navigate to the "Update Firmware" section. If you have Windows then go to section 4.1 and if you have Linux or Mac, go to section 4.2 and follow the steps to ensure that the latest firmware is updated.

After following the appropriate steps, HuskyLens Uploader USB to UART driver and the latest firmware will be installed.

3.4.3.3 Board Overview

- Function button
- Learning button
- Screen



The HuskyLens is equipped with two buttons: the function button and the learning button. Their primary functions are outlined as follows:

- Use the function button to toggle between different functions by turning it left or right.
- Press the learning button briefly to teach the device about a specific object. Holding down the learning button allows continuous learning of the object from various angles and distances. If the HuskyLens has already learned the object, a brief press of the learning button will cause it to forget.
- To access the second-level menu (parameter settings) within the current function, press and hold the function button. Use left or right turns of the button or a brief press to adjust related parameters.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

HuskyLens Capabilities:

- Face Recognition
- Object Tracking
- Object Recognition
- Line Tracking
- Color Recognition
- Tag Recognition
- Object Classification

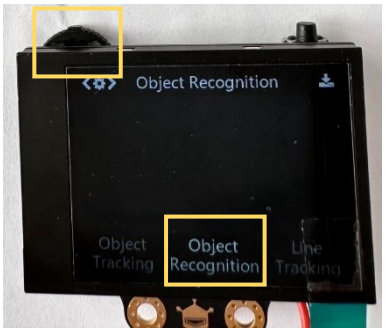
Latest software version:

- Navigate to General Settings -> Version
- Check the following address for instructions on how to update to the newest version (if it exists): https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336

3.4.3.4 Get started

After setting up the wiring between the micro:bit and HuskyLens, the micro:bit must be connected with the PC through a cable. Once the connection is done, the HuskyLens camera alongside with micro:bit will turn on.

Then the student must navigate to the Face Recognition section with the help of the button which is located at the top left corner of the HuskyLens camera. Move it right until find the Face Recognition section.

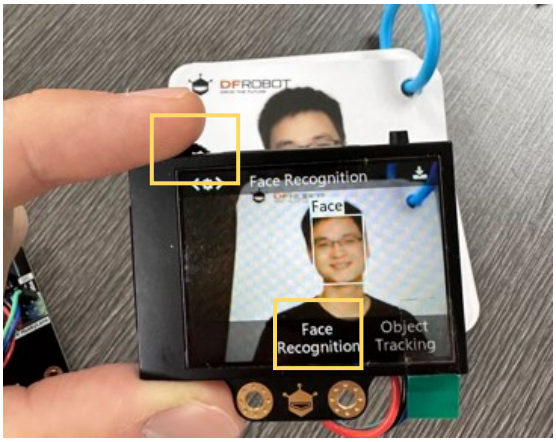


Co-funded by the European Union

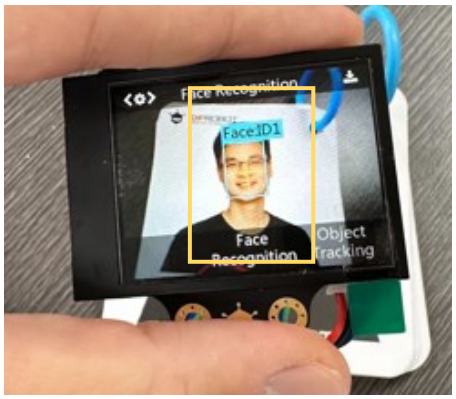
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

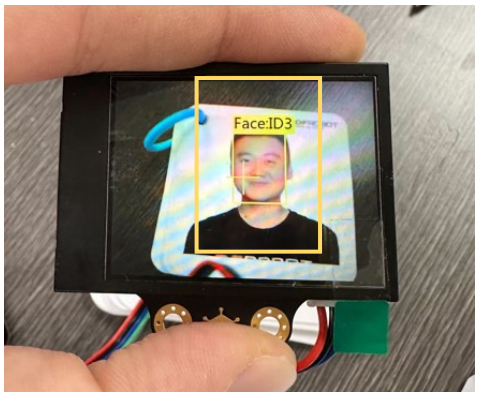
After Face Recognition is selected, point the camera to an object or person to see the outcome. HuskyLens camera can detect objects by tacking live input and it can give an outcome to the user. This represents the perception of this system.

For the Face recognition setting, you have the option to save more than 1 face. First, select the face recognition option and long-press the function button as shown below.



Then click on the Learning Multiple and then Save and Return. Now the learning multiple function is activated and multiple faces can be scanned and saved to the HuskyLens camera.





Co-funded by the European Union

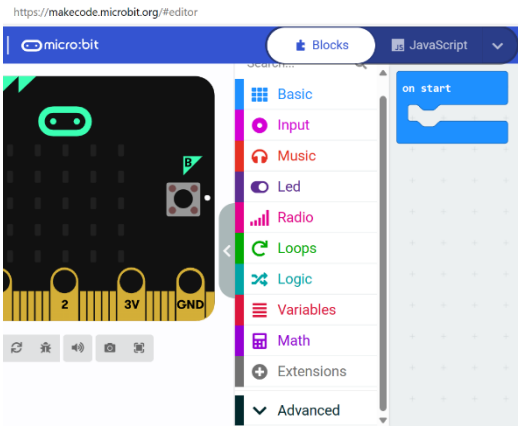
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

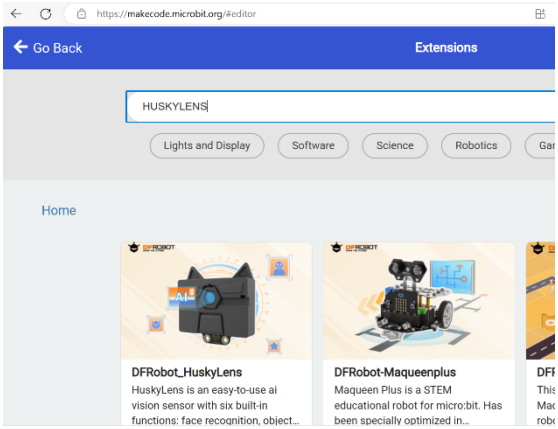
3.4.3.5 Code

After setting up the wiring between the micro:bit and HuskyLens, students should proceed to write code that provides instructions to the HuskyLens camera, enabling the creation of an Object Recognition system. This code will guide the camera in capturing and processing objects to recognize them effectively.

The student must use the software on the following website to write the code: https://makecode.microbit.org/

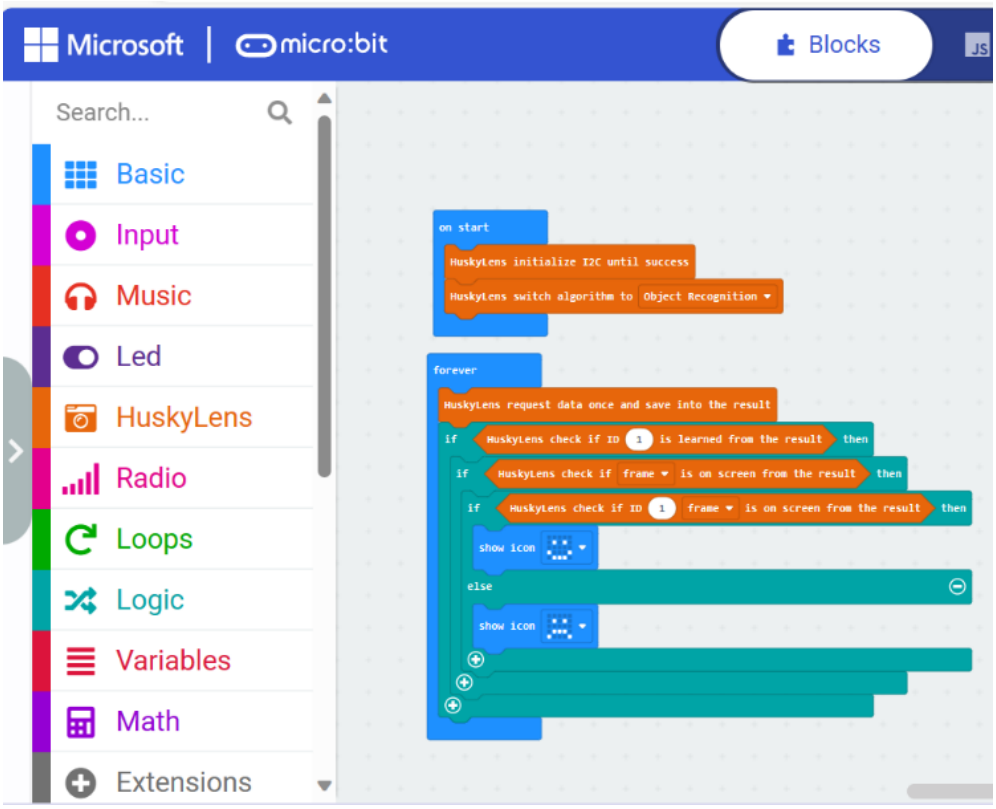
To use the code below, a HuskyLens block must be added. To add this block, select the "Extensions" block and then search for DFRobot_HuskyLens and select it. Then the HuskyLens block will automatically be added to the block section and you can then proceed with the code below.





Co-funded by the European Union

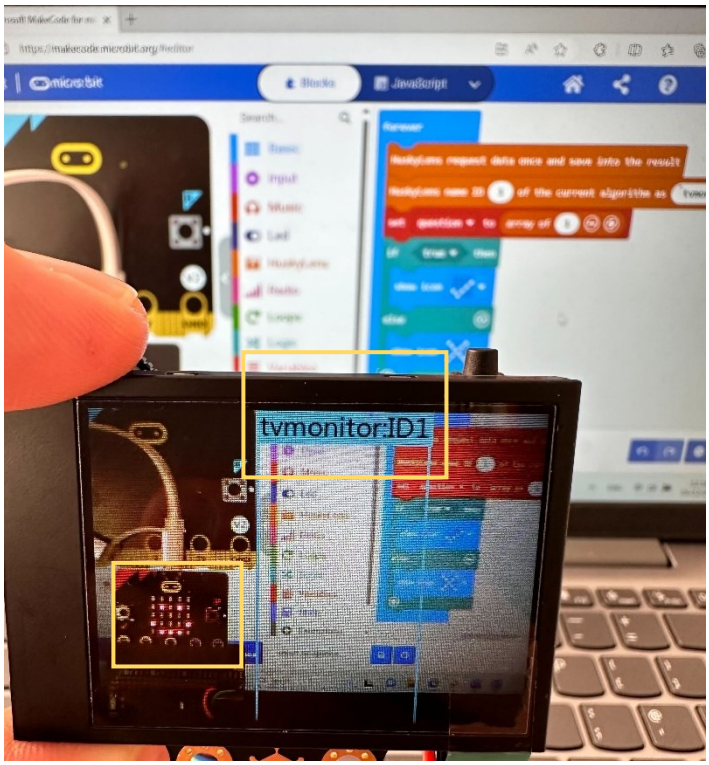
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

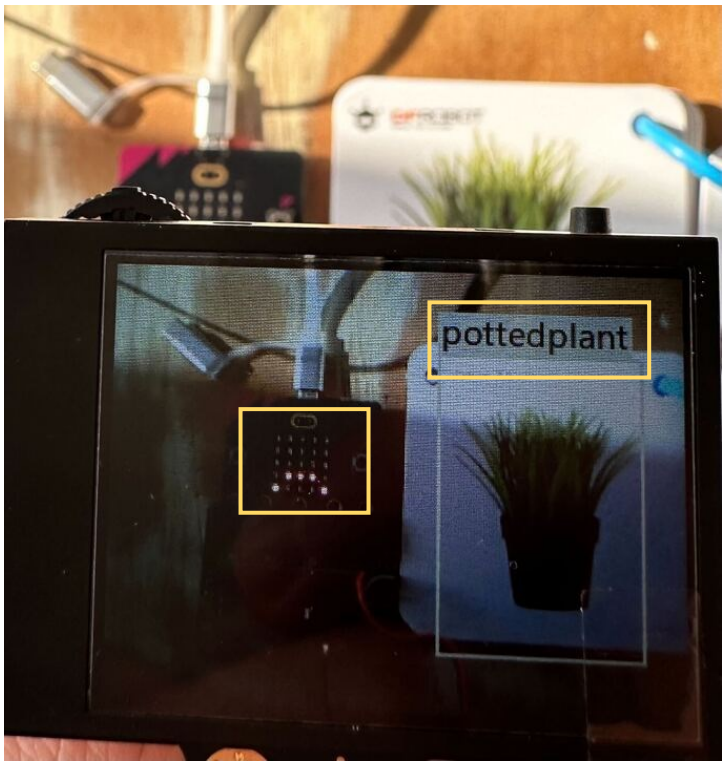


Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Outcome:





Code explanation:

- The HuskyLens camera using the I2C protocol. It prepares the micro:bit to talk to the camera.
- To utilize the HuskyLens feature, begin by adding the HuskyLens extension to your coding palette. With this extension, you can create code that seamlessly switches the algorithm to Face Recognition mode.
- We are telling the HuskyLens to work in object recognition mode. This means it will try to recognize and differentiate objects.
- Next part of the code sets up a continuous loop that will run as long as the micro:bit is powered on.
- Then the micro:bit asks the HuskyLens to provide information about what it's currently seeing.
- The program checks if the HuskyLens has learned any objects. The number 1 refers to the object's label, and if it's learned something, the code inside the curly braces will run.
- The HuskyLens is detecting an object in its view.
- Next it checks if a specific object (in this case, object 1) is detected.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

- If the HuskyLens recognizes the object and it's appearing as a block, the micro:bit displays a happy face icon.
- If the object isn't recognized as a block, it displays a sad face icon.

3.4.4 Experiment 1

In this exercise, students will have the opportunity to apply the technology of HuskyLens and micro:bit to recognize different shapes. This exercise challenges their understanding of object recognition and allows them to explore the practical applications of this technology.

The primary goal of this exercise is for students to create a program that utilizes the HuskyLens camera to identify and distinguish different shapes, such as a circle, triangle, rectangle and then display corresponding emojis or symbols on the micro:bit's display.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.5 Activity 2: Introducing the idea of Representation and Reasoning

3.5.1 Description

In this activity, students will explore the world of artificial intelligence and computer vision in order to explore face recognition method. This activity will introduce the idea of training an AI model to perform specific tasks or make intelligent decisions. The training phase is crucial for an AI model because it is during this stage that the model learns and adapts to the task or problem it is designed to solve. Training is essentially the process of teaching the AI model by exposing it to a large dataset, allowing the model to learn patterns, correlations, and rules from the provided information.

In this activity, a HuskyLens camera is installed and it enables the AI model to gather information directly. During the training process, the AI model adjusts its internal parameters through iterative optimization, enhancing its ability to make accurate predictions or classifications.

The micro:bit platform, acting as the brains of the operation, becomes the interface through which students can observe and interact with the trained AI model. This hands-on aspect of the activity allows students to not only understand the theoretical underpinnings of machine learning but also to appreciate the practical implications of deploying such technology in real-world scenarios.

Using the HuskyLens camera and micro:bit platform, students will discover how technology can be used to train the AI model with familiar human faces. By the end of the activity, students not only grasp the fundamentals of face and object recognition and AI training but also gain a broader appreciation for the potential and challenges of integrating AI into everyday life.

3.5.2 Hardware

- Micro:bit
- IO Extender for micro:bit
- HuskyLens
- Object images

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.5.3 Setup

3.5.3.1 Wiring

- Connect the micro:bit with the IO extender for micro:bit.
- Then use the following table to connect the IO extender for micro:bit with the HuskyLens camera.

IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

3.5.3.2 Get started

After setting up the wiring between the micro:bit and HuskyLens, the micro:bit must be connected with the PC through a cable. Once the connection is done, the HuskyLens camera alongside with micro:bit will turn on.

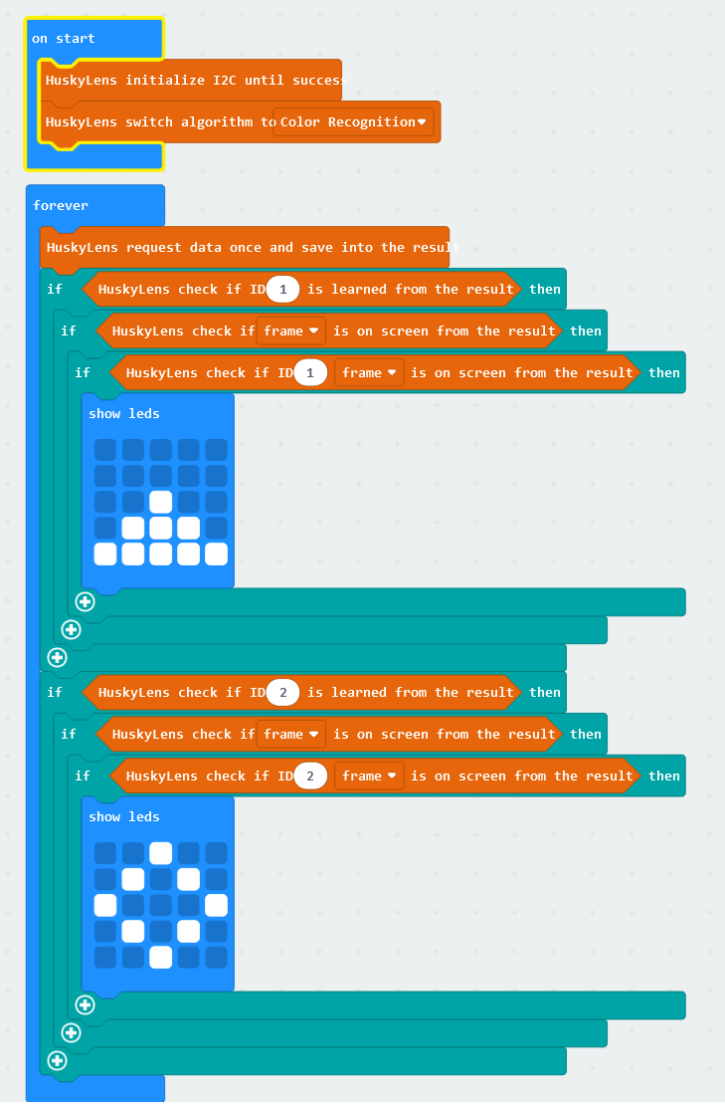
Then the student must navigate to Face Recognition section with the help of the button which is located at the top left corner of the HuskyLens camera. Move it right until find Face Recognition section.

3.5.3.3 Code

After setting up the wiring between the micro:bit and HuskyLens, students should proceed to write code that provides instructions to the HuskyLens camera, enabling the creation of a colour recognition system. This code will guide the camera in capturing and processing colors to recognize shapes effectively.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Code explanation:

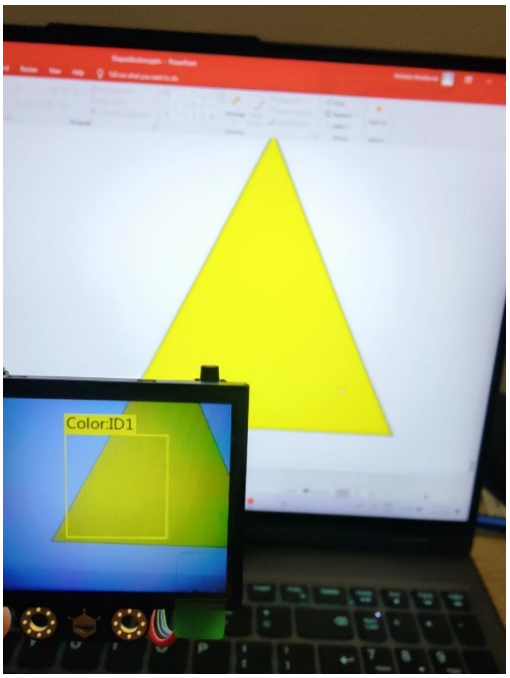
To utilize the HuskyLens feature, begin by adding the HuskyLens extension to your coding palette. With this extension, you can create code that seamlessly switches the algorithm to Color Recognition mode.

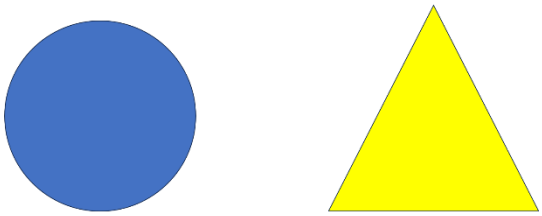
Initializes the I2C communication between the microcontroller and the HuskyLens. It will keep trying until the initialization is successful. Sets the HuskyLens to use the Color Recognition algorithm. Requests data from the HuskyLens once and stores it in the result. Ensures that the color with ID 1 is learned by the HuskyLens. If color ID 1 is recognized, a triangle LED pattern is on the microcontroller's screen.

Ensures that the color with ID 2 is learned by the HuskyLens. If color ID 2 is recognized, display a circle LED pattern on the microcontroller's screen.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.





3.5.4 Exercise: Train the AI model to recognise different shapes

In this exercise, students will have the opportunity to apply the technology of HuskyLens and micro:bit to recognize shapes. This exercise challenges their understanding of the training phase of the AI model.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.6 Activity 3: Introducing the idea of Learning by training a model for Face Recognition

3.6.1 Description

In this activity, students will delve into the dynamic phase that follows the training phase of the AI model— the learning phase. This phase marks the practical application of the model's acquired knowledge and the assessment of its ability to provide accurate outcomes in real-time scenarios, specifically in the context of face recognition.

The learning phase involves putting the trained AI model to the test, evaluating its performance, and ensuring that it can correctly identify and respond to the familiar faces it has been trained on. The key objective is to determine whether the model is capable of delivering the right outcome, a critical step in affirming that the AI model is functioning as intended.

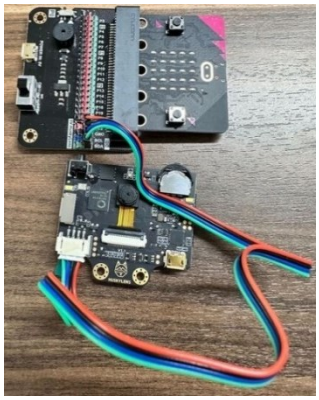
The micro:bit platform serves as the interface through which students interact with the AI model. As users, students are tasked with checking if the model correctly recognizes the trained face captured by the HuskyLens camera. This hands-on aspect of the activity provides a tangible and practical demonstration of how AI models transition from theoretical learning to real-world application.

The process involves capturing images using the HuskyLens camera and allowing the AI model to analyze and identify the faces within the images. The micro:bit then serves as the feedback mechanism, displaying the outcome of the recognition process. If the AI model successfully identifies the trained face, it serves as an indicator that the learning phase has been effective, and the model is working properly.

This activity not only reinforces the technical concepts related to face recognition and AI but also emphasizes the importance of validation and real-world testing in the development and deployment of AI systems. Students gain insights into the challenges and considerations involved in ensuring the reliability and accuracy of AI models in practical applications.

3.6.2 Hardware

- Micro:bit
- IO Extender for micro:bit
- HuskyLens
- Color images



Co-funded by the European Union

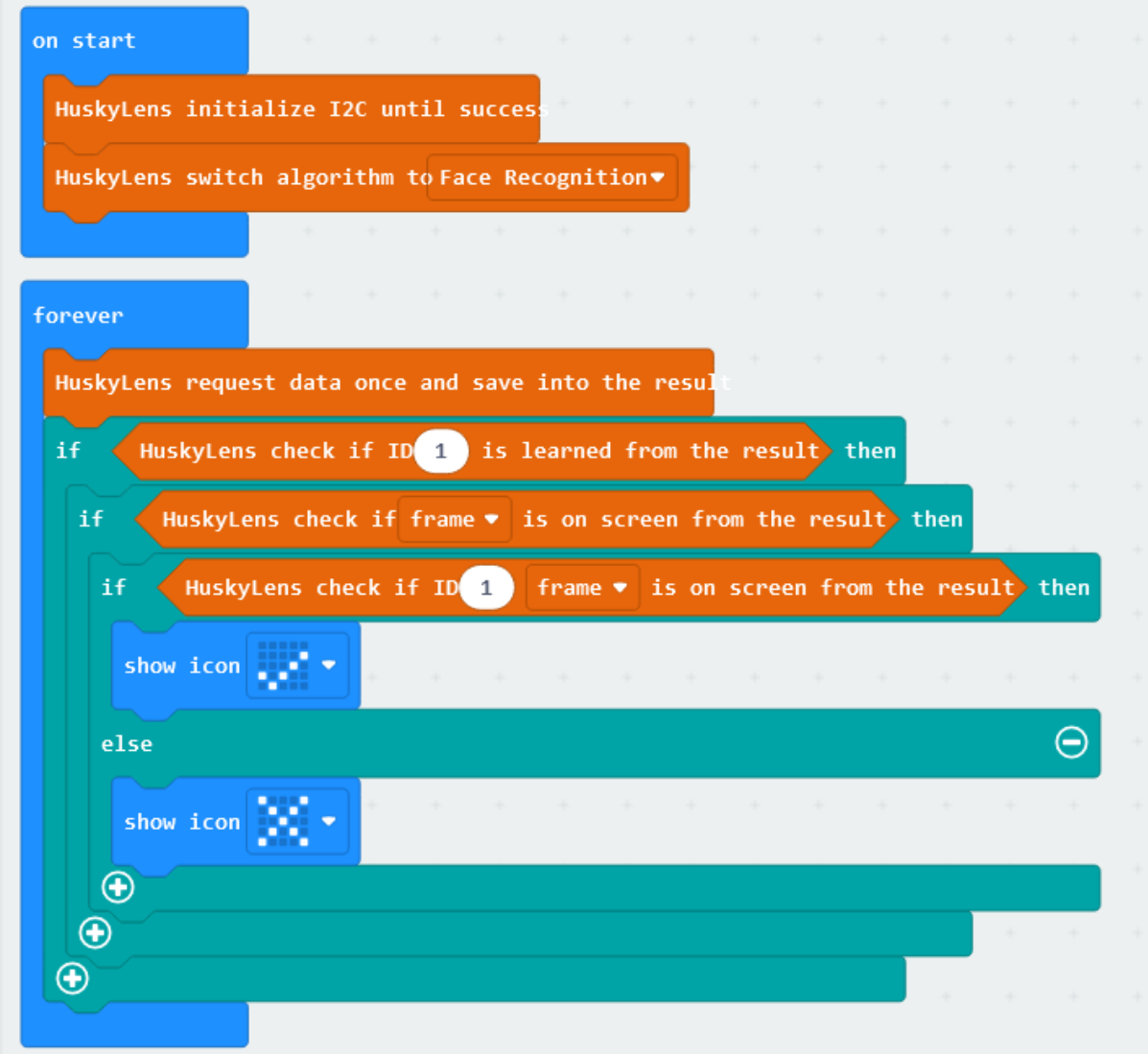
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.6.3 Setup

- Connect the micro:bit with the IO extender for micro:bit.
- Then use the following table to connect the IO extender for micro:bit with the HuskyLens camera.

IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

3.6.4 Code



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Code explanation:

To utilize the HuskyLens feature, begin by adding the HuskyLens extension to your coding palette. With this extension, you can create code that seamlessly switches the algorithm to Face Recognition mode.

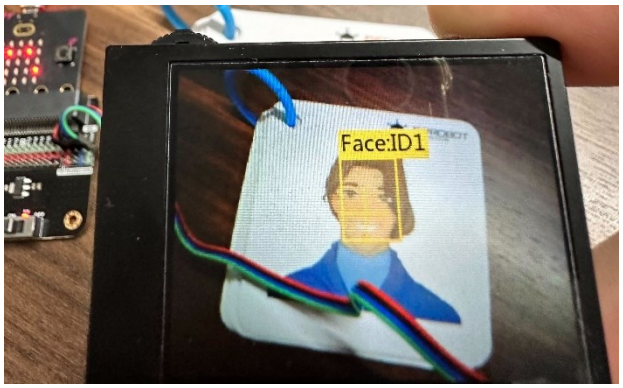
This code will prompt for input data, requiring the student to use the HuskyLens camera to capture a person's face. The user trained the AI model with a specific face on the previous activity.

Moving forward, when the camera detects the recognized face, the micro:bit will respond with a cheerful smiley face or a correct answer signal as output, while a random or unrecognized face will result in a sad face or wrong signal output. This engaging process bridges the gap between technology and human interaction, making it a fascinating learning experience.

- The HuskyLens camera using the I2C protocol. It prepares the micro:bit to talk to the camera.
- We are telling the HuskyLens to work in color recognition mode. This means it will try to recognize and differentiate objects based on their colors.
- The next part of the code sets up a continuous loop that will run as long as the micro:bit is powered on.
- Then the micro:bit asks the HuskyLens to provide information about what it's currently seeing.
- The program checks if the HuskyLens has learned any objects. The number 1 refers to the object's label, and if it's learned something, the code inside the curly braces will run.
- The HuskyLens is detecting an object in its view.
- Next it checks if a specific object (in this case, object 1) is detected.
- If the HuskyLens recognizes the object and it appears as a block, the micro:bit displays a happy face icon.
- If the object isn't recognized as a block, it displays a sad face icon.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



3.6.4 Exercise: Test if the AI model recognizes you

Using the HuskyLens and micro:bit system to train the AI model with a new face with the person's name on it and then capture the new person's face with the HuskyLens camera to see if it can recognize it and if it has the right label on that person.

This exercise will test the user if can train and test the AI model.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.7 Activity 4: Introducing the idea of Natural Interaction by integrating a trained model to an AI application

3.7.1 Description

In this engaging continuation of the AI exploration journey, students will progress to the testing phase, a critical step that follows the learning phase. The testing phase serves as a comprehensive assessment of the AI model's performance, evaluating its ability to generalize and respond accurately to various cases. Through the lens of face recognition using the HuskyLens camera and micro:bit platform, students will delve into the intricacies of the testing phase.

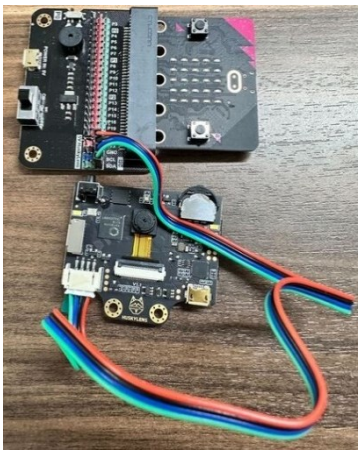
The testing phase is the step after the learning phase. In this phase, the AI model is tested in multiple cases to see if it's working properly. After this phase, appropriate adjustments can be done. The testing phase is an essential checkpoint in the development of an AI model, as it simulates real-world scenarios and assesses the model's adaptability and reliability. After the learning phase, during which the model has become familiar with specific faces, the testing phase introduces variability by capturing both familiar and unfamiliar human faces.

In this activity, students will capture multiple faces using the HuskyLens camera, including those that the AI model has been trained on (familiar faces) and others that it has not encountered during the training and learning phases (unfamiliar faces). The micro:bit will then display the outcome of the recognition process for each captured face.

Using the HuskyLens camera and micro:bit platform, students will discover how technology can be used to test the AI model with familiar and unfamiliar human faces.

3.7.2 Hardware

- Micro:bit
- IO Extender for micro:bit
- HuskyLens
- Color images



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.7.3 Setup

- Connect the micro:bit with the IO extender for micro:bit.
- Then use the following table to connect the IO extender for micro:bit with the HuskyLens camera.

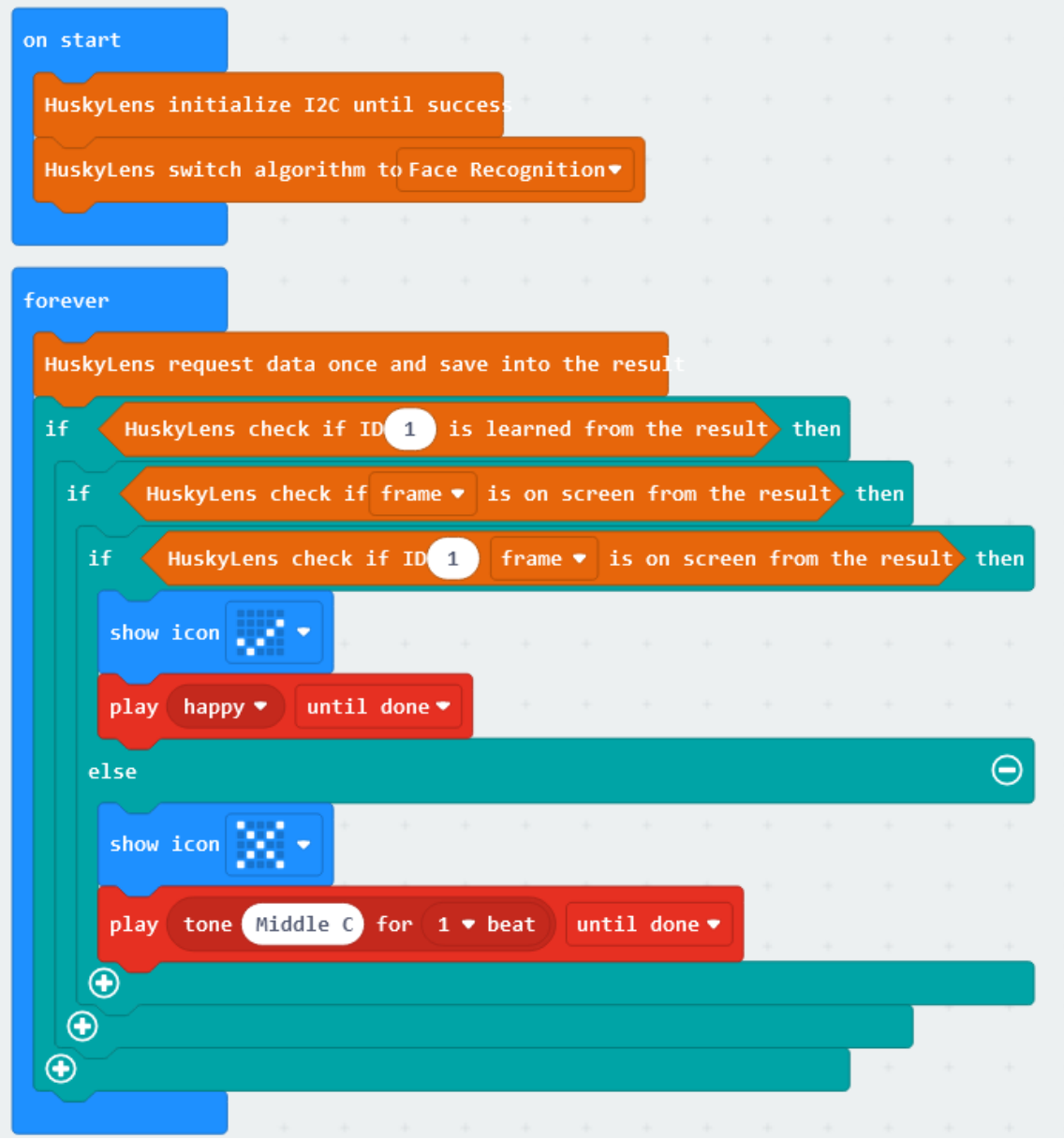
IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

3.7.3.1 Code

After setting up the wiring between the micro:bit and HuskyLens, students should proceed to write code that provides instructions to the HuskyLens camera, enabling the creation of an Face Recognition system. This code will guide the camera in capturing and processing faces to recognize them effectively.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



The student must use the software on the following website to write the code: https://makecode.microbit.org/

Code explanation:

To utilize the HuskyLens feature, begin by adding the HuskyLens extension to your coding palette. With this extension, you can create code that seamlessly switches the algorithm to Face Recognition mode.

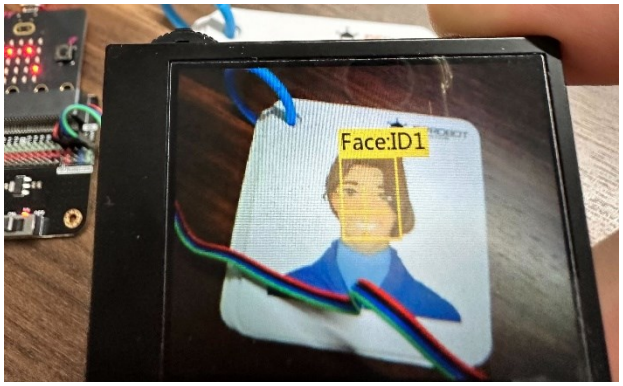
This code will prompt for input data, requiring the student to use the HuskyLens camera to capture a person's face. The user trained the AI model with a specific face on the previous activity.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Moving forward, when the camera detects the recognized face, the micro:bit will respond with a cheerful smiley face as output and a happy sound, while a random or unrecognized face will result in a sad face output and a sad sound. This engaging process bridges the gap between technology and human interaction, making it a fascinating learning experience.

- The HuskyLens camera using the I2C protocol. It prepares the micro:bit to talk to the camera.
- We are telling the HuskyLens to work in color recognition mode. This means it will try to recognize and differentiate objects based on their colors.
- The next part of the code sets up a continuous loop that will run as long as the micro:bit is powered on.
- Then the micro:bit asks the HuskyLens to provide information about what it's currently seeing.
- The program checks if the HuskyLens has learned any objects. The number 1 refers to the object's label, and if it's learned something, the code inside the curly braces will run.
- The HuskyLens is detecting an object in its view.
- Next it checks if a specific object (in this case, object 1) is detected.
- If the HuskyLens recognizes the object and it appears as a block, the micro:bit displays a happy face icon.
- If the object isn't recognized as a block, it displays a sad face icon.



3.7.4 Exercise: Alarm system

Using the HuskyLens and micro:bit system to train the AI model with a new face with the person's name on it and then capture the new person's face with the HuskyLens camera to see if it can recognize it and if the camera detects a person that is not in the system to make a sound. The user will create an alarm system for this exercise.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.8 Activity 5: Introducing the Idea of Societal Impact

3.8.1 Description

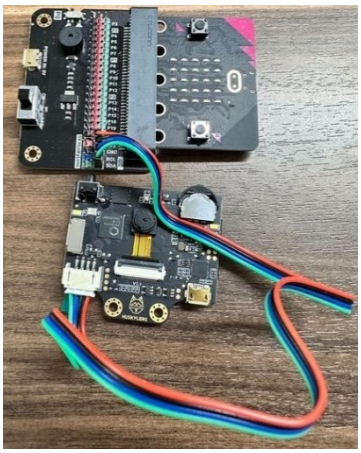
This lesson introduces the concept of societal impact by training a model for multiple face recognition. Students will explore how to encode and process facial data to train a model capable of recognizing multiple faces simultaneously.

They will learn how to implement this technology and discuss the societal implications of AI systems that can detect and identify faces in various situations. The ability of AI to perform face recognition has significant societal impacts, including enhanced security, privacy concerns, and ethical considerations regarding surveillance and data usage.

This lesson aims to provide students with a comprehensive understanding of both the technical and societal aspects of face recognition technology.

3.8.2 Hardware

- Micro:bit
- IO Extender for micro:bit
- HuskyLens
- Images with faces



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.8.3 Setup

- Connect the micro:bit with the IO extender for micro:bit.
- Then use the following table to connect the IO extender for micro:bit with the HuskyLens camera.

IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

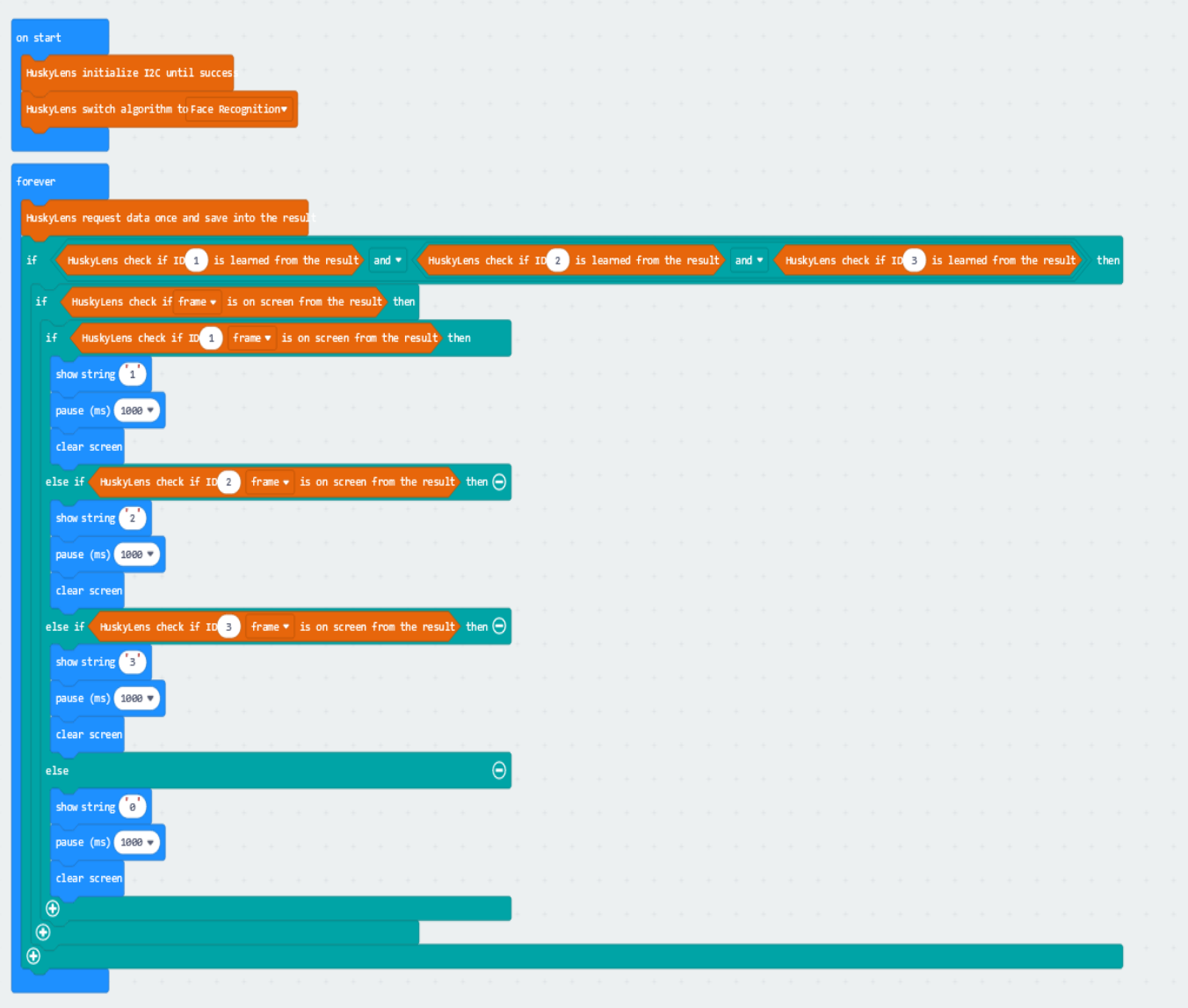
3.8.3.1 Code

After setting up the wiring between the micro:bit and HuskyLens, students should proceed to write code that provides instructions to the HuskyLens camera, enabling the creation of an Face Recognition system. This code will guide the camera in capturing and processing faces to recognize them effectively. It is important for this activity to use the setting for capturing multiple images in order to create the following activity.

The student must use the software on the following website to write the code: https://makecode.microbit.org/

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Code explanation:

To utilize the HuskyLens feature, begin by adding the HuskyLens extension to your coding palette. With this extension, you can create code that seamlessly switches the algorithm to Face Recognition mode.

Initializes the I2C communication between the microcontroller and the HuskyLens. It will keep trying until the initialization is successful. Sets the HuskyLens to use the Face Recognition algorithm. Requests data from the HuskyLens once and stores it in the result. Ensures that faces with IDs 1, 2, and 3 are learned by the HuskyLens.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

The code continuously runs in a loop, requesting data from the HuskyLens and checking if any of the three faces are recognized. Depending on which face is detected, it displays the corresponding number (1, 2, or 3) on the microcontroller's screen. If no face is recognized, it displays "0". This provides real-time feedback based on face recognition, making the interaction engaging and informative for the user.

3.8.4 Introducing the idea of Natural Interaction by training a model for Face Recognition

Students will train the HuskyLens with a face and write code to detect it and play a sound.

For this activity, it is important to consider the following steps: Use the Face Recognition option, long press the function button in Face Recognition, select Multiple, and record 3 faces.

3.9 Additional Material and Resources

Type of Resource	Title	Topic	Link
software	Makecode	Block coding	https://makecode.microbit.org/
Article	Usage of Microbit and Huskylens	How to use Huskylens	https://www.instructables.com/Microbit-Visual-Object-Tracking-With-Huskylens/

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.