



Cofinancé par
l'Union européenne

Financé par l'Union européenne. Les points de vue et avis exprimés n'engagent toutefois que leur(s) auteur(s) et ne reflètent pas nécessairement ceux de l'Union européenne ou de l'Agence exécutive européenne pour l'éducation et la culture (EACEA). Ni l'Union européenne ni l'EACEA ne sauraient en être tenues pour responsables..

Technologie IA & Vision par ordinateur



Présentation des 5 grandes idées en intelligence artificielle utilisant l'Internet des objets dans l'éducation STEM

T2.4 Conception de projets IoT et développement de ressources

29.08.2023 | EMPHASYS CENTRE
NUMÉRO DE PROJET: 2022-1-FR01-KA220-SCH-000085611

Projets IoT AI4STEM

Projet: Technologie IA & Vision par ordinateur

Copyright

© Copyright au AI4STEM Consortium

2022-1-FR01-KA220-SCH-000085611

Tous droits réservés.



Projets IoT AI4STEM Projet: Technologie IA & Vision par ordinateur © 2023 par [AI4STEM CONSORTIUM](#) est sous licence [Attribution - Utilisation non commerciale - Partage dans les mêmes conditions 4.0 International](#)

Table des matières

1.Introduction au projet.....	Error! Bookmark not defined.
1.1 La portée du projet	Error! Bookmark not defined.
1.2 Les groupes cibles	Error! Bookmark not defined.
1.3 Objet de ce document	Error! Bookmark not defined.
2. Glossaire de l'Unité	Error! Bookmark not defined.
3. Introduction au « Réveil Intelligent »	Error! Bookmark not defined.
3.1 Description	Error! Bookmark not defined.
3.2 Objectifs et résultats d'apprentissage	Error! Bookmark not defined.
3.3 Durée estimée de l'Unité	Error! Bookmark not defined.
3.4 Lesson 1 - Name: Exploring Device Perception.....	Error! Bookmark not defined.
3.4.1 Description	Error! Bookmark not defined.
3.4.2 Matériel.....	Error! Bookmark not defined.
3.4.3 Configuration	Error! Bookmark not defined.
3.4.4 Activité	Error! Bookmark not defined.
3.4.5 Questions	Error! Bookmark not defined.
3.5 Leçon 2 Prise de décision avec des capteurs : représentation et raisonnement	Error! Bookmark not defined.
3.5.1 Introduction – Theorie	Error! Bookmark not defined.
3.5.2 Matériel.....	Error! Bookmark not defined.
3.5.3 Configuration	Error! Bookmark not defined.
3.5.4 Activité	Error! Bookmark not defined.
3.5.5 Questions	Error! Bookmark not defined.
3.6 Leçon 3 Enseigner votre alarme : apprendre à partir des données.....	Error! Bookmark not defined.
3.6.1 Introduction – Théorie	Error! Bookmark not defined.
3.6.2 Matériel.....	Error! Bookmark not defined.
3.6.3 Configuration	Error! Bookmark not defined.
3.6.4 Exercice / Experience 3	Error! Bookmark not defined.
3.6.5 Questions	Error! Bookmark not defined.
3.7 Leçon 4 Alarme télécommandée : Interagir avec les boutons micro:bit	Error! Bookmark not defined.
3.7.1 Introduction – Theorie	Error! Bookmark not defined.
3.7.2 Matériel.....	Error! Bookmark not defined.

3.7.3 Configuration	Error! Bookmark not defined.
3.7.4 Activité	Error! Bookmark not defined.
3.7.5 Questions	Error! Bookmark not defined.
3.8 Leçon 5 Construire des systèmes responsables : impact sociétal des alarmes IoT	Error! Bookmark not defined.
3.8.1 Introduction – Théorie	Error! Bookmark not defined.
3.8.2 Matériel.....	Error! Bookmark not defined.
3.8.3 Configuration	Error! Bookmark not defined.
3.8.4 Exercice / Experience 3	Error! Bookmark not defined.
3.8.5 Questions	Error! Bookmark not defined.

1. Introduction au projet

Pour ce projet, micro:bit et HuskyLens seront utilisés pour réaliser différentes activités. HuskyLens est un capteur de vision intelligent ou un module de caméra qui combine l'intelligence artificielle (IA) et la technologie de vision par ordinateur pour effectuer diverses tâches telles que la reconnaissance d'objets, la reconnaissance faciale, le suivi, la reconnaissance des couleurs, etc. L'intégration de l'intelligence artificielle (IA) avec la caméra HuskyLens et micro:bit est un projet passionnant qui vous permet de créer des applications interactives et intelligentes.

1.1 La portée du projet

La portée du projet est d'utiliser micro:bit et HuskyLens pour créer un modèle d'intelligence artificielle. Le modèle sera formé via la caméra HuskyLens et donnera un résultat via l'outil micro:bit.

1.2 Les groupes cibles

Le projet vise principalement la participation directe des éducateurs, principalement de l'enseignement primaire et secondaire supérieur.

1.3 Objet de ce document

L'objectif de ce document est de démontrer comment entraîner et tester un modèle d'IA à l'aide de la caméra HuskyLens et afficher la sortie sur le micro:bit.

2. Glossaire de l'Unité

Mot	Définition
Micro:bit	Le micro:bit est une carte informatique programmable de poche conçue pour l'éducation. Il comprend une matrice LED, divers capteurs et un microcontrôleur
HuskyLens	Le HuskyLens est un module de capteur de vision capable de reconnaître des objets, des visages et des gestes. Il est équipé d'une caméra et d'une puissance de traitement intégrée pour effectuer des tâches liées à l'IA.

3. Introduction à « l'IA en vision »

3.1 Descriptif

Cette unité présentera aux apprenants le processus de formation et de test d'un modèle d'IA avec la caméra HuskyLens. Ils apprendront à installer et à configurer la caméra HuskyLens pour des tâches de reconnaissance d'images, à entraîner le modèle d'IA à reconnaître des objets ou des modèles spécifiques et, enfin, à interfacer la caméra avec un micro:bit pour afficher la sortie du modèle d'IA. Tout au long de l'unité, les apprenants acquerront une compréhension pratique de la vision par ordinateur, de l'apprentissage automatique et de la manière d'intégrer les technologies d'IA dans des projets du monde réel.

3.2 Objectifs et résultats d'apprentissage

À la fin de cette unité, les apprenants devraient être capables de :

- Compréhension de la vision par ordinateur
- Formation sur les modèles d'IA
- Configuration HuskyLens
- Intégration Micro:bit
- Résolution de problème
- Application pratique
- Utiliser HuskyLens
- Collecter des données
- Former des modèles d'IA
- Interagissez avec Micro:bit

3.3 Durée estimée de l'Unité

Il s'agit d'un projet assez long nécessitant plusieurs heures pour aborder correctement tous les aspects inclus. La durée suivante est indicative et peut varier en fonction de l'âge et du niveau de vos élèves.

Activité 1 : 90 minutes

Activité 2 : 90 minutes

Activité 3 : 45 minutes

Activité 4 : 45 minutes

Activité 5 : 45 minutes

3.4 Activité 1 - Présentation de la grande idée de la perception grâce à l'IoT :

3.4.1 Description

Dans cette leçon, les étudiants exploreront le monde de l'intelligence artificielle et de la vision par ordinateur afin d'explorer la méthode de perception, grâce au suivi d'objets en direct, une fonctionnalité rendue possible par les capacités de pointe de la caméra HuskyLens et de la plateforme micro:bit.

À l'aide de la caméra HuskyLens et de la plateforme micro:bit, les étudiants découvriront comment la technologie peut être utilisée pour obtenir une entrée en direct et fournir une sortie à l'utilisateur. L'aspect du suivi en direct démontre non seulement les capacités en temps réel du système, mais souligne également la nature dynamique de la technologie dans sa réponse à son environnement. À l'ère des progrès technologiques rapides, la fusion de l'Internet des objets (IoT) et des technologies de traitement d'images de pointe a ouvert la voie à des applications révolutionnaires. L'une de ces innovations est l'intégration de HuskyLens, une puissante caméra intelligente, avec Micro:bit. Ensemble, ils forment un système dynamique de suivi d'objets en temps réel, révolutionnant la façon dont nous percevons et interagissons avec notre environnement. À mesure que les étudiants s'initient au système HuskyLens et Micro:bit, ils acquièrent non seulement une expérience pratique des outils de pointe, mais participent également au récit continu du progrès technologique.

3.4.2 Matériel

- Microcontrôleur Micro:bit
- Extension IO pour micro:bit
- HuskyLens

3.4.3 Configuration

3.4.3.1 Câblage

- Connectez le micro:bit au prolongateur IO pour micro:bit.
- Utilisez ensuite le tableau suivant pour connecter le prolongateur IO pour micro:bit à la caméra HuskyLens.

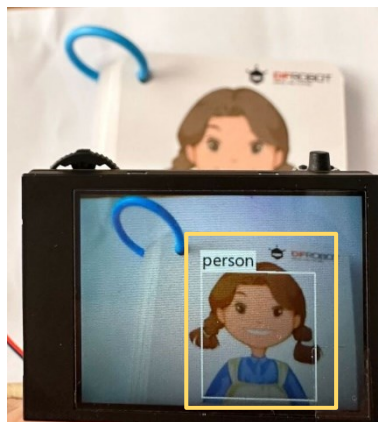
IO Extender for micro:bit PORT	HuskyLens PORT
SDA	T
SCL	R
GND	-
3V3	+

3.4.3.2 Démarrage

Après avoir configuré le câblage entre le micro:bit et HuskyLens, le micro:bit doit être connecté au PC via un câble. Une fois la connexion établie, la caméra HuskyLens ainsi que le micro:bit s'allumeront. Ensuite, l'étudiant doit accéder à la section Reconnaissance d'objets à l'aide du bouton situé dans le coin supérieur gauche de la caméra HuskyLens. Déplacez-le vers la droite jusqu'à trouver la section Reconnaissance d'objets.

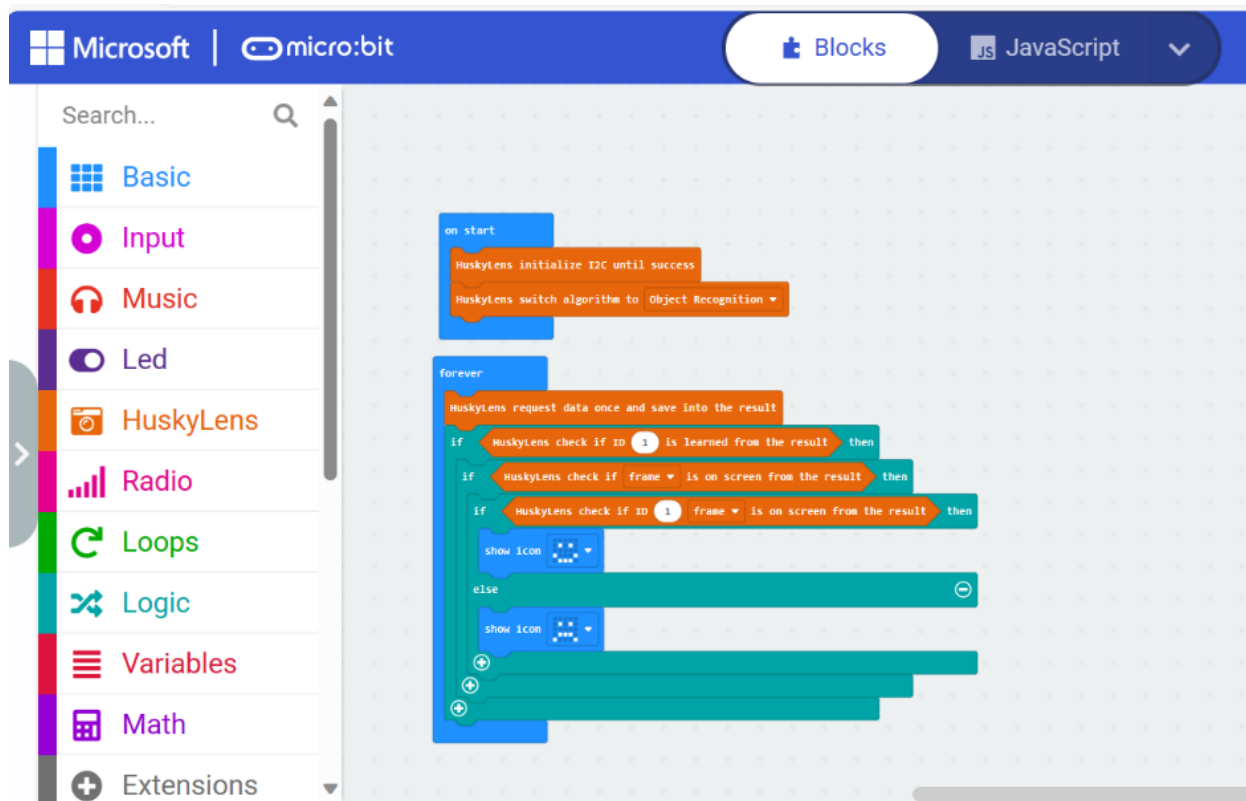


Une fois la reconnaissance d'objet sélectionnée, pointez la caméra vers un objet ou une personne pour voir le résultat. La caméra HuskyLens peut détecter des objets en captant les entrées en direct et peut donner un résultat à l'utilisateur. Cela représente la perception de ce système.

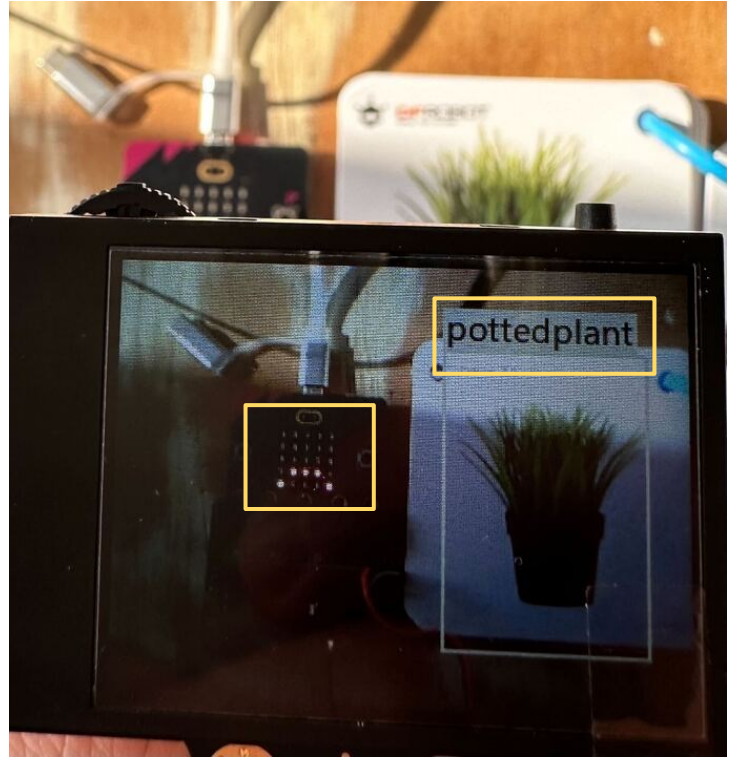
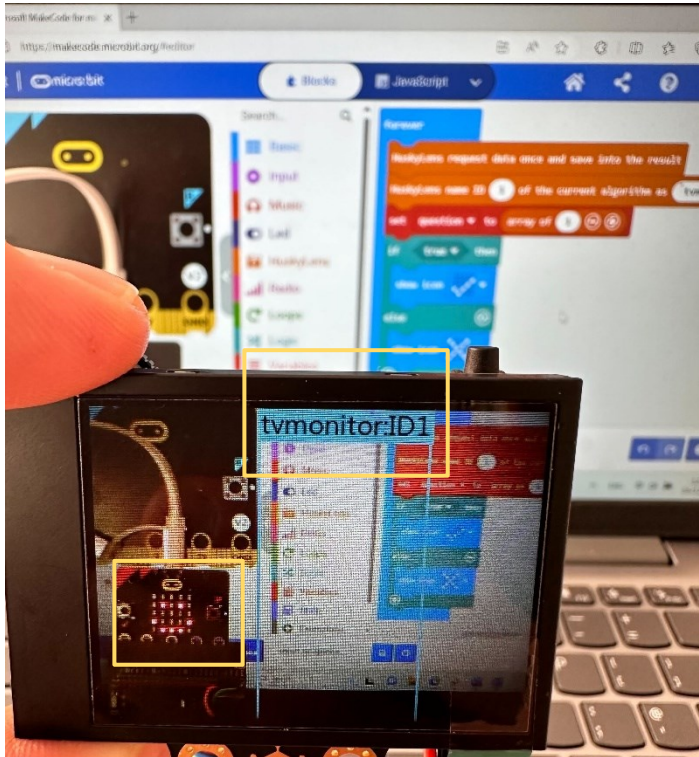


3.4.3.3 Code

Après avoir configuré le câblage entre le micro:bit et HuskyLens, les étudiants doivent écrire du code qui fournit des instructions à la caméra HuskyLens, permettant la création d'un système de reconnaissance d'objets. Ce code guidera la caméra dans la capture et le traitement des objets pour les reconnaître efficacement. L'étudiant doit utiliser le logiciel du site suivant pour écrire le code : <https://makecode.microbit.org/>



Résultat:



Explication du code :

- La caméra HuskyLens utilisant le protocole I2C. Il prépare le micro:bit à parler à la caméra.
- Pour utiliser la fonctionnalité HuskyLens, commencez par ajouter l'extension HuskyLens à votre palette de codage. Avec cette extension, vous pouvez créer du code qui fait passer de manière transparente l'algorithme en mode de reconnaissance faciale.
- Nous disons au HuskyLens de fonctionner en mode de reconnaissance d'objets. Cela signifie qu'il essaiera de reconnaître et de différencier les objets.
- La partie suivante du code établit une boucle continue qui fonctionnera tant que le micro:bit sera allumé.
- Ensuite, le micro:bit demande au HuskyLens de fournir des informations sur ce qu'il voit actuellement.
- Le programme vérifie si le HuskyLens a appris des objets. Le chiffre 1 fait référence à l'étiquette de l'objet, et s'il a appris quelque chose, le code entre les accolades s'exécutera.
- Le HuskyLens détecte un objet dans son champ de vision.
- Ensuite, il vérifie si un objet spécifique (dans ce cas, l'objet 1) est détecté.

- Si le HuskyLens reconnaît l'objet et qu'il apparaît sous forme de bloc, le micro:bit affiche une icône de visage heureux.
- Si l'objet n'est pas reconnu comme un bloc, il affiche une icône de visage triste.

3.4.4 Expérience 1

Dans cet exercice, les étudiants auront l'occasion d'appliquer la technologie HuskyLens et micro:bit pour reconnaître différentes formes. Cet exercice remet en question leur compréhension de la reconnaissance d'objets et leur permet d'explorer les applications pratiques de cette technologie. L'objectif principal de cet exercice est que les étudiants créent un programme qui utilise la caméra HuskyLens pour identifier et distinguer différentes formes, telles qu'un cercle, un triangle, un rectangle, puis afficher les émojis ou les symboles correspondants sur l'écran du micro:bit.

3.5 Activité 2 : Introduire l'idée de représentation et de raisonnement

3.5.1 Description

Dans cette activité, les étudiants exploreront le monde de l'intelligence artificielle et de la vision par ordinateur afin d'explorer la méthode de reconnaissance faciale. Cette activité introduira l'idée de former un modèle d'IA pour effectuer des tâches spécifiques ou prendre des décisions intelligentes. La phase de formation est cruciale pour un modèle d'IA car c'est au cours de cette étape que le modèle apprend et s'adapte à la tâche ou au problème qu'il est conçu pour résoudre. La formation est essentiellement le processus d'apprentissage du modèle d'IA en l'exposant à un vaste ensemble de données, permettant au modèle d'apprendre des modèles, des corrélations et des règles à partir des informations fournies. Dans cette activité, une caméra HuskyLens est installée et permet au modèle d'IA de recueillir directement des informations. Au cours du processus de formation, le modèle d'IA ajuste ses paramètres internes grâce à une optimisation itérative, améliorant ainsi sa capacité à effectuer des prédictions ou des classifications précises. La plateforme micro:bit, agissant comme le cerveau de l'opération, devient l'interface à travers laquelle les étudiants peuvent observer et interagir avec le modèle d'IA formé. Cet aspect pratique de l'activité permet aux étudiants non seulement de comprendre les fondements théoriques de l'apprentissage automatique, mais également d'apprécier les implications pratiques du déploiement d'une telle technologie dans des scénarios du monde réel.

À l'aide de la caméra HuskyLens et de la plateforme micro:bit, les étudiants découvriront comment la technologie peut être utilisée pour entraîner le modèle d'IA avec des visages humains familiers. À la fin de l'activité, les étudiants maîtrisent non seulement les principes fondamentaux de la reconnaissance faciale et de la formation en IA, mais acquièrent également une appréciation plus large du potentiel et des défis liés à l'intégration de l'IA dans la vie quotidienne.

3.5.2 Matériel

- Micro : bits
- Extension IO pour micro:bit
- HuskyLens
- Images d'objets



3.5.3 Configuration

3.5.3.1 Wiring

- Connectez le micro:bit au prolongateur IO pour micro:bit.
- Utilisez ensuite le tableau suivant pour connecter le prolongateur IO pour micro:bit à la caméra HuskyLens.

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

3.5.3.2 Démarrage

Après avoir configuré le câblage entre le micro:bit et HuskyLens, le micro:bit doit être connecté au PC via un câble. Une fois la connexion établie, la caméra HuskyLens ainsi que le micro:bit s'allumeront. Ensuite, l'étudiant doit accéder à la section Reconnaissance faciale à l'aide du bouton situé dans le coin supérieur gauche de la caméra HuskyLens. Déplacez-le vers la droite jusqu'à trouver la section Reconnaissance faciale.



3.5.3.3 Code

Après avoir configuré le câblage entre le micro:bit et HuskyLens, les étudiants doivent écrire du code qui fournit des instructions à la caméra HuskyLens, permettant la création d'un système de reconnaissance faciale. Ce code guidera la caméra dans la capture et le traitement des images pour reconnaître efficacement les visages.

Explication du code :

Pour utiliser la fonctionnalité HuskyLens, commencez par ajouter l'extension HuskyLens à votre palette de codage. Avec cette extension, vous pouvez créer du code qui fait passer de manière transparente l'algorithme en mode de reconnaissance faciale.

Ce code demandera des données d'entrée, obligeant l'étudiant à utiliser la caméra HuskyLens pour capturer le visage d'une personne.

Pour capturer un nouveau visage, appuyez simplement sur le bouton d'apprentissage, en suivant les repères visuels fournis dans les images qui l'accompagnent. S'il y a déjà un visage

enregistré, appuyer une fois sur le bouton d'apprentissage lancera une invite pour oublier le visage précédent.

Une fois confirmé, un réticule apparaîtra sur l'écran. Déplacez le réticule vers le visage que vous souhaitez capturer et appuyez à nouveau sur le bouton d'apprentissage pour l'enregistrer en tant que données apprises. Il s'agit de la phase de formation du modèle d'IA.

- La caméra HuskyLens utilisant le protocole I2C. Il prépare le micro:bit à parler à la caméra.
- Nous disons au HuskyLens de fonctionner en mode de reconnaissance faciale. Cela signifie qu'il essaiera de reconnaître et de différencier les objets.
- La partie suivante du code établit une boucle continue qui fonctionnera tant que le micro:bit sera allumé.
- Ensuite, le micro:bit demande au HuskyLens de fournir des informations sur ce qu'il voit actuellement.
- Pour entraîner le modèle IA, pointez la caméra de HuskyLens vers le visage et attendez qu'une croix avec l'étiquette « visage » apparaisse sur l'écran de HuskyLens.
- Ensuite, s'il s'agit du visage avec lequel vous souhaitez entraîner l'IA, cliquez sur le bouton dans le coin supérieur droit pendant 3 secondes et la couleur de l'étiquette changera.



3.5.4 Exercice : Entraînez le modèle IA avec votre visage

Dans cet exercice, les étudiants auront l'occasion d'appliquer la technologie HuskyLens et micro:bit pour reconnaître le visage de l'utilisateur. Cet exercice remet en question leur compréhension de la phase de formation du modèle d'IA.

3.6 Présentation de l'idée d'apprentissage en formant un modèle pour la reconnaissance faciale

3.6.1 Description

Dans cette activité, les étudiants plongeront dans la phase dynamique qui suit la phase de formation du modèle d'IA : la phase d'apprentissage. Cette phase marque l'application pratique des connaissances acquises par le modèle et l'évaluation de sa capacité à fournir des résultats précis dans des scénarios en temps réel, notamment dans le contexte de la reconnaissance faciale.

La phase d'apprentissage consiste à tester le modèle d'IA formé, à évaluer ses performances et à s'assurer qu'il peut identifier et répondre correctement aux visages familiers sur lesquels il a été formé. L'objectif clé est de déterminer si le modèle est capable de produire le bon résultat, une étape cruciale pour affirmer que le modèle d'IA fonctionne comme prévu. La plateforme micro:bit sert d'interface à travers laquelle les étudiants interagissent avec le modèle d'IA. En tant qu'utilisateurs, les étudiants sont chargés de vérifier si le modèle reconnaît correctement le visage entraîné capturé par la caméra HuskyLens. Cet aspect pratique de l'activité fournit une démonstration tangible et pratique de la façon dont les modèles d'IA passent de l'apprentissage théorique à l'application dans le monde réel. Le processus consiste à capturer des images à l'aide de la caméra HuskyLens et à permettre au modèle d'IA d'analyser et d'identifier les visages dans les images. Le micro:bit sert alors de mécanisme de rétroaction, affichant le résultat du processus de reconnaissance. Si le modèle d'IA parvient à identifier le visage entraîné, cela indique que la phase d'apprentissage a été efficace et que le modèle fonctionne correctement. Cette activité renforce non seulement les concepts techniques liés à la reconnaissance faciale et à l'IA, mais souligne également l'importance de la validation et des tests réels dans le développement et le déploiement de systèmes d'IA. Les étudiants acquièrent un aperçu des défis et des considérations nécessaires pour garantir la fiabilité et la précision des modèles d'IA dans des applications pratiques.

3.6.2 Matériel

- Micro : bits
- Extension IO pour micro:bit
- HuskyLens
- Images en couleur



3.6.3 Configuration

- Connectez le micro:bit au prolongateur IO pour micro:bit.
- Utilisez ensuite le tableau suivant pour connecter le prolongateur IO pour micro:bit à la caméra HuskyLens.

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

Explication du code :

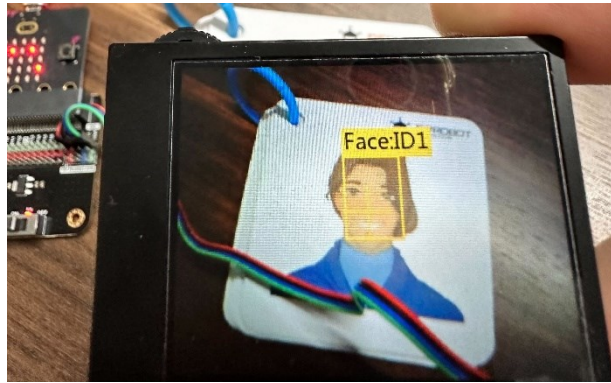
Pour utiliser la fonctionnalité HuskyLens, commencez par ajouter l'extension HuskyLens à votre palette de codage. Avec cette extension, vous pouvez créer du code qui fait passer de manière transparente l'algorithme en mode de reconnaissance faciale.

Ce code demandera des données d'entrée, obligeant l'étudiant à utiliser la caméra HuskyLens pour capturer le visage d'une personne. L'utilisateur a entraîné le modèle d'IA avec un visage spécifique sur l'activité précédente.

À l'avenir, lorsque la caméra détecte le visage reconnu, le micro:bit répondra avec un visage souriant joyeux en sortie, tandis qu'un visage aléatoire ou non reconnu entraînera une sortie de visage triste. Ce processus engageant comble le fossé entre la technologie et l'interaction humaine, ce qui en fait une expérience d'apprentissage fascinante.

- La caméra HuskyLens utilisant le protocole I2C. Il prépare le micro:bit à parler à la caméra.
- Nous demandons au HuskyLens de fonctionner en mode de reconnaissance des couleurs. Cela signifie qu'il essaiera de reconnaître et de différencier les objets en fonction de leurs couleurs.
- La partie suivante du code met en place une boucle continue qui fonctionnera tant que le micro:bit sera allumé.
- Ensuite, le micro:bit demande au HuskyLens de fournir des informations sur ce qu'il voit actuellement.
- Le programme vérifie si le HuskyLens a appris des objets. Le chiffre 1 fait référence à l'étiquette de l'objet, et s'il a appris quelque chose, le code entre les accolades s'exécutera.
- Le HuskyLens détecte un objet dans son champ de vision.

- Ensuite, il vérifie si un objet spécifique (dans ce cas, l'objet 1) est détecté.
- Si le HuskyLens reconnaît l'objet et qu'il apparaît comme un bloc, le micro:bit affiche une icône de visage heureux.
- Si l'objet n'est pas reconnu comme un bloc, il affiche une icône de visage triste.



3.6.4 Testez si le modèle IA vous reconnaît

Utiliser le système HuskyLens et micro:bit pour entraîner le modèle d'IA avec un nouveau visage portant le nom de la personne, puis capturer le visage de la nouvelle personne avec la caméra HuskyLens pour voir s'il peut le reconnaître et s'il porte la bonne étiquette dessus personne.

Cet exercice testera l'utilisateur s'il peut entraîner et tester le modèle d'IA.

3.7 Activité 4 : Présentation de l'idée d'interaction naturelle en intégrant un modèle entraîné à une application d'IA

3.7.1 Description

Dans cette poursuite engageante du parcours d'exploration de l'IA, les étudiants passeront à la phase de test, une étape critique qui suit la phase d'apprentissage. La phase de test sert à une évaluation complète des performances du modèle d'IA, évaluant sa capacité à généraliser et à répondre avec précision à divers cas. Grâce à la reconnaissance faciale utilisant la caméra HuskyLens et la plateforme micro:bit, les étudiants plongeront dans les subtilités de la phase de test. La phase de test est l'étape qui suit la phase d'apprentissage. Au cours de cette phase, le modèle d'IA est testé dans plusieurs cas pour voir s'il fonctionne correctement. Après cette phase, des ajustements appropriés peuvent être effectués. La phase de test est un point de contrôle essentiel dans le développement d'un modèle d'IA, car elle simule des scénarios réels et évalue l'adaptabilité et la fiabilité du modèle. Après la phase d'apprentissage, au cours de laquelle le modèle s'est familiarisé avec des visages spécifiques, la phase de test introduit de la variabilité en capturant à la fois des visages humains familiers et inconnus. Dans cette activité, les étudiants captureront plusieurs visages à l'aide de la caméra HuskyLens, y compris ceux sur lesquels le modèle d'IA a été formé (visages familiers) et d'autres qu'il n'a pas rencontrés pendant les phases de formation et d'apprentissage (visages inconnus). Le micro:bit affichera alors le résultat du processus de reconnaissance pour chaque visage capturé. À l'aide de la caméra HuskyLens et de la plateforme micro:bit, les étudiants découvriront comment la technologie peut être utilisée pour tester le modèle d'IA avec des visages humains familiers et inconnus.

3.7.2 Matériel

- Micro : bits
- Extension IO pour micro:bit
- HuskyLens
- Images en couleur



3.7.3 Configuration

- Connectez le micro:bit au prolongateur IO pour micro:bit.
- Utilisez ensuite le tableau suivant pour connecter le prolongateur IO pour micro:bit à la caméra HuskyLens.

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

3.7.3.1 Code

Après avoir configuré le câblage entre le micro:bit et HuskyLens, les étudiants doivent écrire du code qui fournit des instructions à la caméra HuskyLens, permettant la création d'un système de reconnaissance faciale. Ce code guidera la caméra dans la capture et le traitement des visages pour les reconnaître efficacement.

L'étudiant doit utiliser le logiciel du site suivant pour écrire le code : <https://makecode.microbit.org/>

Explication du code :

Pour utiliser la fonctionnalité HuskyLens, commencez par ajouter l'extension HuskyLens à votre palette de codage. Avec cette extension, vous pouvez créer du code qui fait passer de manière transparente l'algorithme en mode de reconnaissance faciale.

Ce code demandera des données d'entrée, obligeant l'étudiant à utiliser la caméra HuskyLens pour capturer le visage d'une personne. L'utilisateur a entraîné le modèle d'IA avec un visage spécifique sur l'activité précédente.

À l'avenir, lorsque la caméra détecte le visage reconnu, le micro:bit répondra avec un visage souriant joyeux en sortie, tandis qu'un visage aléatoire ou non reconnu entraînera une sortie de visage triste et un son triste. Ce processus engageant comble le fossé entre la technologie et l'interaction humaine, ce qui en fait une expérience d'apprentissage fascinante.

- La caméra HuskyLens utilisant le protocole I2C. Il prépare le micro:bit à parler à la caméra.

- Nous demandons au HuskyLens de fonctionner en mode de reconnaissance des couleurs. Cela signifie qu'il essaiera de reconnaître et de différencier les objets en fonction de leurs couleurs.
- La partie suivante du code met en place une boucle continue qui fonctionnera tant que le micro:bit sera allumé.
- Ensuite, le micro:bit demande au HuskyLens de fournir des informations sur ce qu'il voit actuellement.
- Le programme vérifie si le HuskyLens a appris des objets. Le chiffre 1 fait référence à l'étiquette de l'objet, et s'il a appris quelque chose, le code entre les accolades s'exécutera.
- Le HuskyLens détecte un objet dans son champ de vision.
- Ensuite, il vérifie si un objet spécifique (dans ce cas, l'objet 1) est détecté.
- Si le HuskyLens reconnaît l'objet et qu'il apparaît comme un bloc, le micro:bit affiche une icône de visage heureux.
- Si l'objet n'est pas reconnu comme un bloc, il affiche une icône de visage triste.



3.7.4 Exercice: Système d'alarme

Utiliser le système HuskyLens et micro:bit pour entraîner le modèle d'IA avec un nouveau visage portant le nom de la personne, puis capturer le visage de la nouvelle personne avec la caméra HuskyLens pour voir s'il peut le reconnaître et si la caméra détecte une personne qui est pas dans le système pour émettre un son. L'utilisateur créera un système d'alarme sur cet exercice.

3.8 Activité 5 : Présentation de l'idée d'impact sociétal

3.8.1 Description

À l'ère de l'intelligence artificielle, la combinaison de la caméra HuskyLens avec le microcontrôleur micro:bit crée une plate-forme puissante et accessible qui a le potentiel d'être transformatrice à l'ère de l'intelligence artificielle. Cette combinaison facilite non seulement les expériences d'apprentissage pratique, mais sert également de catalyseur pour l'innovation et l'engagement communautaire.

Le potentiel collaboratif de cette combinaison améliore l'engagement communautaire, ouvrant la voie à des projets collaboratifs, des ateliers et des événements qui contribuent à un paysage technologique plus inclusif et dynamique.

3.8.2 Aspects selon lesquels micro:bit avec HuskyLens a un impact social

HuskyLens et micro:bit sont des outils pouvant être utilisés à des fins pédagogiques et interactives, notamment dans les domaines de la robotique et de la vision par ordinateur. Lorsqu'ils sont combinés, ils peuvent avoir plusieurs impacts sociaux :

Impact éducatif :

- Éducation STEM : l'intégration de HuskyLens avec micro:bit peut améliorer l'éducation STEM. Les étudiants peuvent en apprendre davantage sur la robotique, la vision par ordinateur et la programmation de manière interactive et engageante.
- Apprentissage pratique : l'utilisation de ces outils permet aux étudiants de vivre une expérience d'apprentissage pratique, favorisant une compréhension plus approfondie de la technologie et des concepts de programmation.
- Innovation et créativité : Encourager l'innovation : en fournissant une plate-forme d'expérimentation et d'exploration, ces outils peuvent encourager les étudiants à penser de manière créative et à proposer des solutions innovantes à des problèmes du monde réel.
- Apprentissage par projet : les étudiants peuvent travailler sur des projets impliquant la conception et la construction de systèmes robotiques dotés de capacités de vision par ordinateur, favorisant ainsi l'apprentissage par projet.
- Technologie accessible : Démocratiser la technologie : ces outils sont relativement abordables et accessibles, ce qui rend l'enseignement technologique plus inclusif. Les étudiants d'horizons divers peuvent avoir accès et bénéficier d'opportunités d'apprentissage en robotique et en vision par ordinateur.
- Engagement communautaire: Projets communautaires : l'utilisation de HuskyLens et de micro:bit peut s'étendre au-delà de la salle de classe, permettant aux étudiants de s'impliquer dans des projets communautaires. Par exemple, ils pourraient développer

des solutions à des problèmes locaux en utilisant la robotique et la vision par ordinateur.

Promouvoir la collaboration :

- Travail d'équipe et collaboration : travailler sur des projets impliquant ces outils peut promouvoir le travail d'équipe et la collaboration entre les étudiants. Ils devront peut-être combiner leurs compétences en codage, en électronique et en résolution de problèmes pour atteindre les objectifs du projet.
- Inspirer les futurs technologues : Exposition précoce à la technologie : initier les étudiants à la technologie dès leur plus jeune âge peut susciter un intérêt pour les domaines STEM. L'utilisation de ces outils peut contribuer au développement des futurs technologues et innovateurs.
- Développement de compétence: Compétences en codage et en résolution de problèmes : l'intégration de micro:bit et de HuskyLens offre aux étudiants la possibilité de développer des compétences en codage et en résolution de problèmes, qui sont de plus en plus importantes dans le monde technologique d'aujourd'hui.
- Expositions et installations interactives : Démonstrations publiques : les écoles ou les établissements d'enseignement peuvent utiliser des projets impliquant ces outils pour des démonstrations publiques, mettant en valeur les capacités de la technologie et encourageant l'intérêt pour la science et l'ingénierie.

3.9 Matériel et ressources supplémentaires

Type de ressource	Titre	Sujet	Lien
Logiciel	Makecode	Codage en bloc	https://makecode.microbit.org/
Article	Utilisation de Microbit et Huskylens	Comment utiliser Huskylens	https://www.instructables.com/Microbit-Visual-Object-Tracking-With-Huskylens/