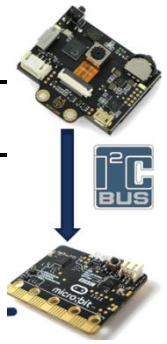
Term spé S.I
Séquence n°6
AE_1 : Activité Expérimentale

Thème sociétal : L'homme assisté

Comment réaliser une reconnaissance de déchets recyclables afin de robotiser une chaîne de tri ?



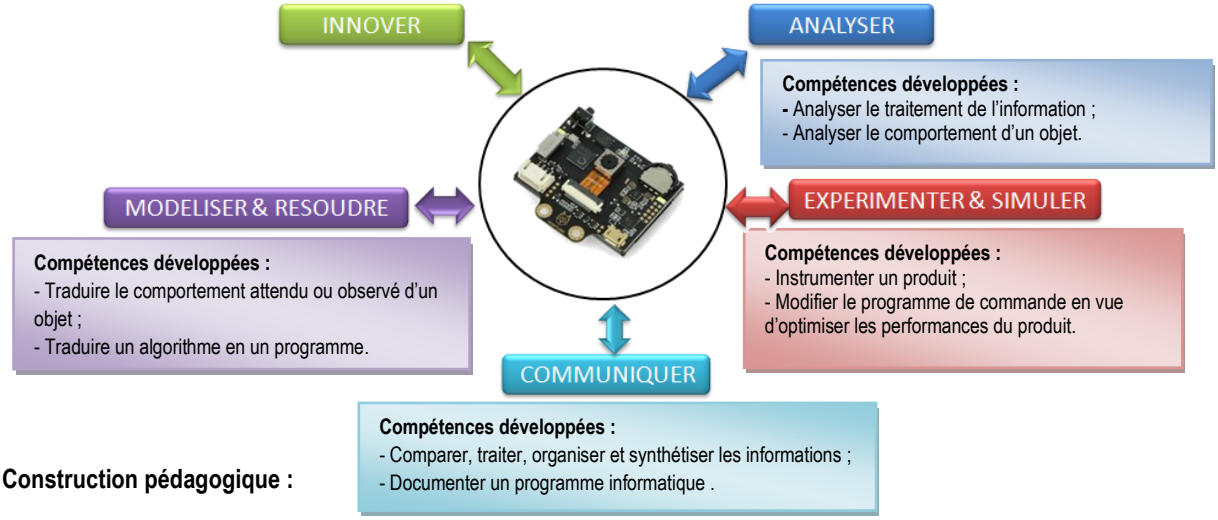
La caméra I.A Huskylens

CONNAISSANCES ASSOCIEES
Algorithme, programme, langage informatique, langage de programmation

PREREQUIS
Cours sur les microprogrammes

Objectif : Paramétrer une caméra I.A, traduire son comportement et traiter les informations afin d'équiper une chaine de production

Compétences terminales visées :



Construction pédagogique :

Thématique(s) générale(s)	L'homme assisté
Thème(s)	Optimisation de la productivité d'une chaine de recyclage
Problématique technique	Comment réaliser une reconnaissance de déchets recyclables afin de robotiser une chaîne de tri ?

	Niveau Term	
Système réel et les matériels : Caméra Huskylens,microbit, PC	Groupe de 2 élèves par poste informatique	**2h**
Poste informatique en réseau Logiciels : Dobot		Fiches ressources : Aucune

Copiez le répertoire «*AE_1_huskylens* »

1- Mise en situation

HuskyLens Gravity est un capteur visuel intelligent, économique, simple d'utilisation basé sur une caméra OV2640 associée à un afficheur 2" IPS et à un processeur Kendryte K210.

- **Applications:** suivi d'objets, reconnaissance faciale, reconnaissance d'objets, suivi de lignes, reconnaissance de couleurs, lecture de QR Codes. Ces applications nécessitent l'utilisation des algorithmes préchargés dans le module.
En plus des programmes préchargés, le HuskyLens est compatible avec les microcontrôleurs Arduino, Micro:bit, Raspberry Pi et LattePanda permettant de la création d'algorithmes et de projets plus complexes ;

- **Fonctionnalités:** Le processeur Kendryte K210 permet l'exécution de tâches de reconnaissances visuelles complexes et également le traitement d'algorithmes d'intelligence artificielle. Ce processeur est composé d'un coeur à 400 Mhz, d'un second coeur à 600 MHz assurant une compatibilité avec le jeu d'instructions RISC-V.

L'affichage de la caméra et la gestion du module s'effectuent directement via l'afficheur intégré. La platine comporte un inverseur de sélection 3 positions permettant les réglages du module (sélection du protocole I2C ou UART, ajustement de la luminosité, activation des Leds d'éclairage, rétroéclairage RGB, réglages usine, etc).

Dans cette activité, vous allez prendre en main la caméra I.A, la paramétrer et programmer la carte microbit afin de répondre à la problématique :

Comment réaliser une reconnaissance de déchets recyclables afin de robotiser une chaîne de tri ?

Visionner la vidéo huskylens qui présente la caméra.

1.1- L'intelligence artificielle (©netapp)

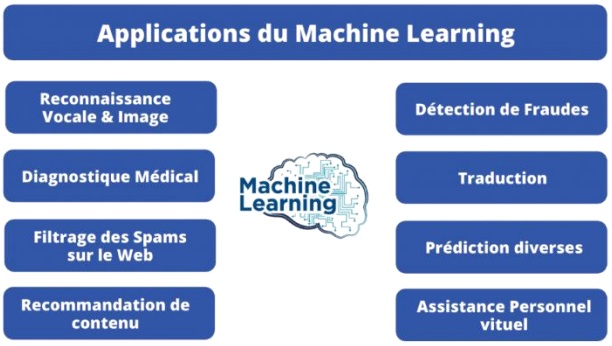
L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains.

Pour créer de l'I.A, vous avez besoin de trois éléments :

- Un système informatique ;
- Des données ;
- Des algorithmes d'IA avancés.

1.1.1- Le machine learning : Apprentissage machine

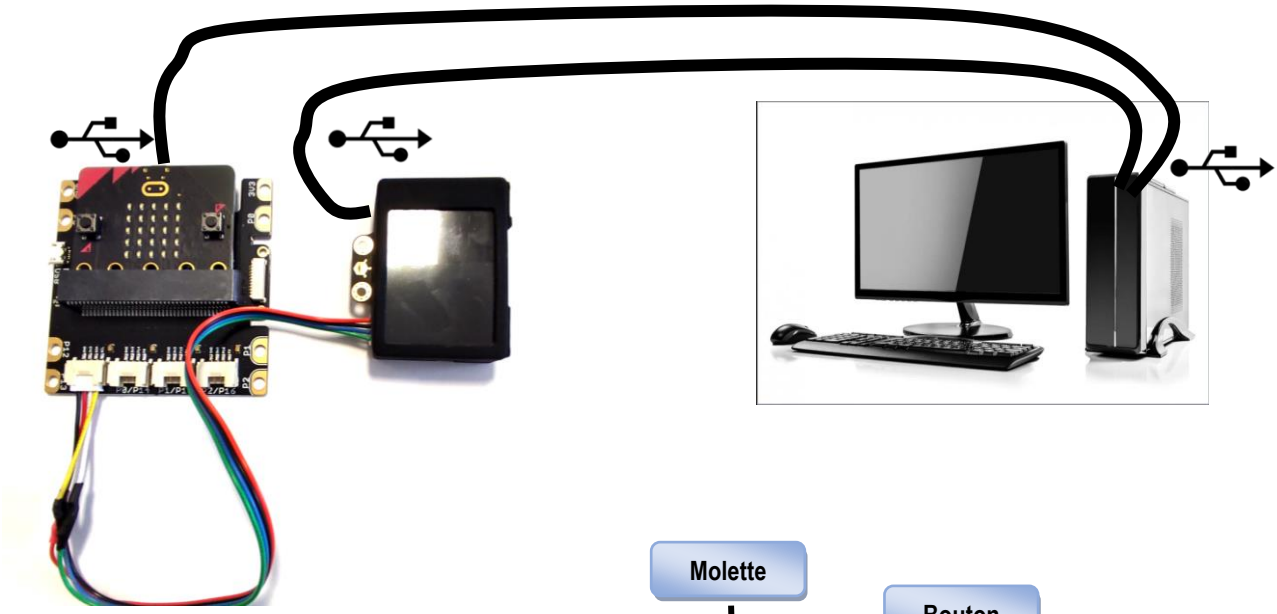
Le machine learning est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune.



2- Première activité de machine learning

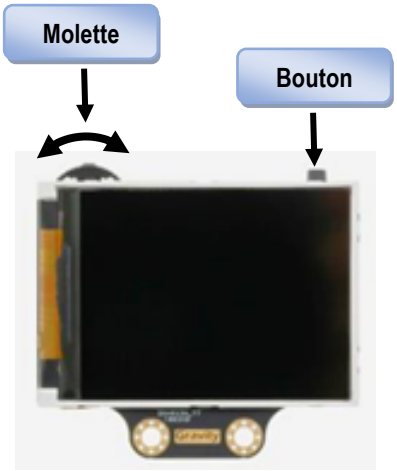
Dans cette partie, vous allez réaliser votre première activité de machine learning à l'aide de la caméra. L'objectif est d'identifier deux images différentes et de programmer la carte Microbit afin d'interagir avec la détection des images.

2.1- Câblage : relier la camera et la carte Microbit au PC à l'aide des câbles USB



2.2- Configuration de la caméra

1- A l'aide de la molette de gauche, aller dans le menu « General Setting » puis « Factory reset » et faire « yes »



2- A l'aide de la molette de gauche, aller dans le menu « objet classification » avec un appui long sur la molette, rentrer dans le menu. Aller dans « Learn multiple » et l'activer. Aller dans le menu « Save & return » et valider par «yes »

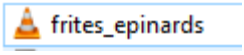
2.3- Reconnaissance d'images

Ouvrir le fichier frites_epinards

Toujours dans le menu « Objet classification », viser les frites avec la caméra et appuyer une fois sur le bouton de droite puis appuyer à nouveau pour valider. Une fenêtre « objet ID1 » doit apparaître. Faire de même avec les épinards une fenêtre ID2 doit apparaître. Enfin, viser le fond blanc et faire de même : une fenêtre ID3 doit apparaître. Si vous déplacez la caméra sur les images, les différentes ID doivent être reconnues.

2.4- Programmation de la carte Microbit

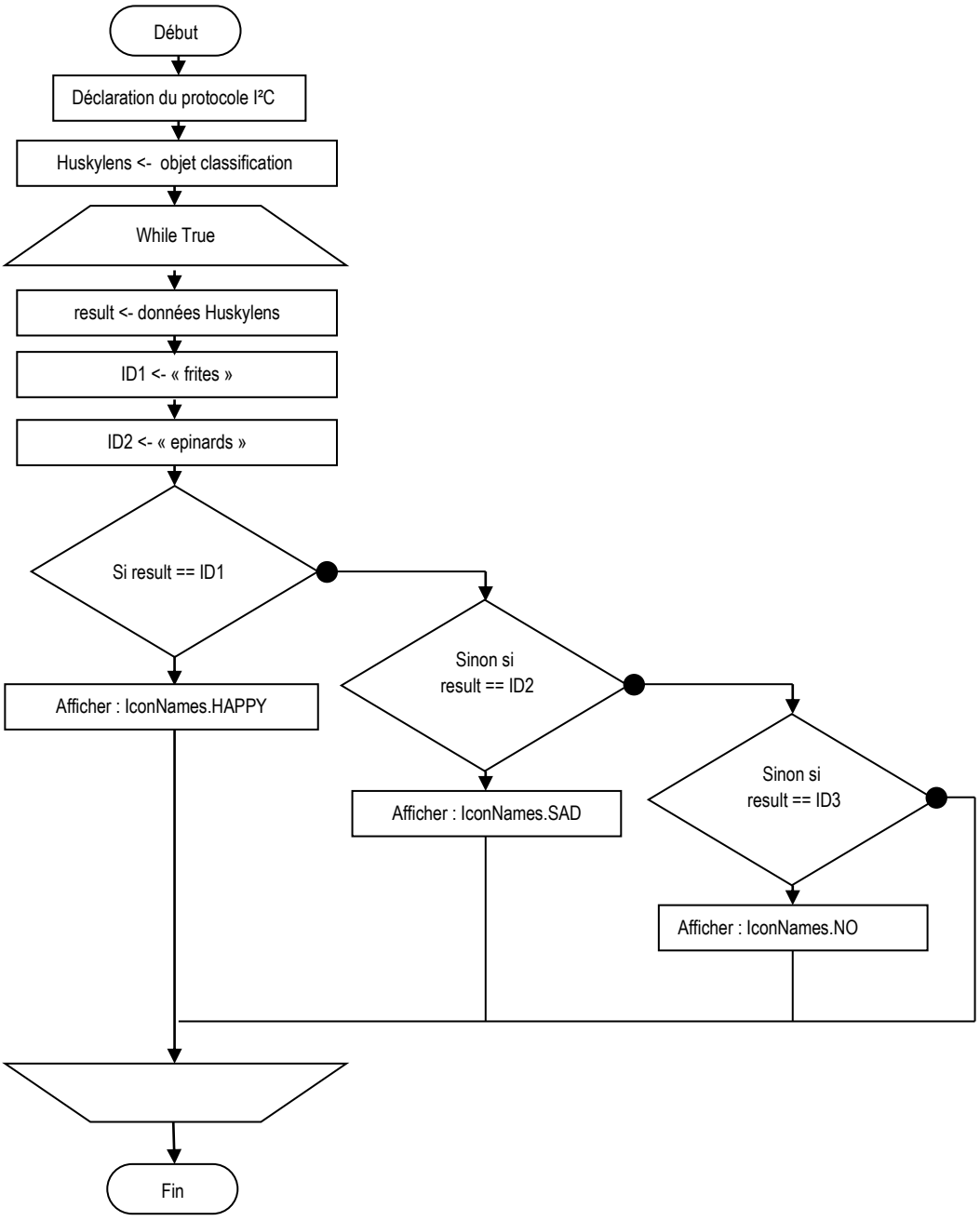
L'objectif de cette partie est d'interagir avec la caméra en fonction de sa détection (c.a.d images frites, épinards ou fond blanc).

On souhaite que lorsque la caméra détecte les frites, un Smiley content  s'affiche sur la carte Microbit, lorsque que la caméra détecte des épinards un Smiley « pas content » s'affiche sur la carte Microbit  et enfin lorsque la caméra détecte le fond blanc une croix s'affiche sur la carte. Visionner la vidéo 

afin d'observer le fonctionnement désiré.

On donne l'algorigramme suivant qui permet de réaliser notre projet :

Algorigramme **littéral** du projet



Compléter le code Python du programme correspondant à l'algorigramme de la page précédente ainsi que les ligne de commentaire (#) :

```
huskylens.init_i2c() #.........................
huskylens.init_mode(protocolAlgorithm.OBJECTCLASSIFICATION)

while True: #.................
    huskylens.request()
    huskylens.write_name(ID1, "..........") #.....................
    huskylens.write_name(ID2, "..........")

    if ..........:
        basic.show_icon(..........)
```

3- Programmation de la carte Microbit

Lancer le raccourci suivant : programme frites l'interface suivante doit s'ouvrir :

1- Cliquez sur modifier ;

2- Puis cliquez sur télécharger, récupérez le fichier .hexa et collez-le dans la carte Microbit ;

3- Testez à l'aide » de la caméra.

Appelez le professeur pour vérification !!!

4- Identification du tri de déchets plastiques et métalliques (apprentissage de forme)

Dans cette partie, vous allez réaliser à l'aide de la caméra I.A, l'identification d'une bouteille plastique et d'une cannette métallique. Ce travail va nous permettre dans l'activité expérimentale n°2, de trier les déchets plastiques et métalliques dans des conteneurs prévus à cet effet.

4.1- Visionner la vidéo detec_hys_bout_canette afin d'observer le fonctionnement désiré.

4.2- Apprentissage de forme

Dans le menu « Objet classification », viser la bouteille d'eau avec la caméra et appuyer en continu sur le bouton de droite tout en déplaçant la caméra afin d'enregistrer toutes les formes de la bouteille (Visionner la vidéo learning_bouteille afin de bien comprendre l'apprentissage de forme) puis appuyer à nouveau pour valider. Une fenêtre « objet ID1 » doit apparaître. Faire de même avec la canette métallique, une fenêtre ID2 doit apparaître. Enfin viser le fond blanc (c.a.d votre table) et faire de même, une fenêtre ID3 doit apparaître. Si vous déplacez la caméra sur les objets, les différentes ID doivent être reconnues.

5- Programmation de la carte Microbit

Le programme Python est donné ci-dessous :

```
huskylens.init_i2c() #.............................................
huskylens.init_mode(protocolAlgorithm.OBJECTCLASSIFICATION) #selection du mode
def on_forever():
    huskylens.request() # Fonction request (cette fonction va identifier l'objet à
la caméra)
    huskylens.write_name(1, "bouteille") #.....................................
    huskylens.write_name(2, "canette")
    huskylens.write_name(3, "fond")
    if huskylens.is_appear(1, HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        basic.show_leds("""
            . # . . .
                        . # . . .
                        . # # # .
                        . # . # .
                        . # # # .
        """)
    elif huskylens.is_appear(2, HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        basic.show_leds("""
            . . . . .
                        . # # # .
                        . # . . .
                        . # . . .
                        . # # # .
        """)
```

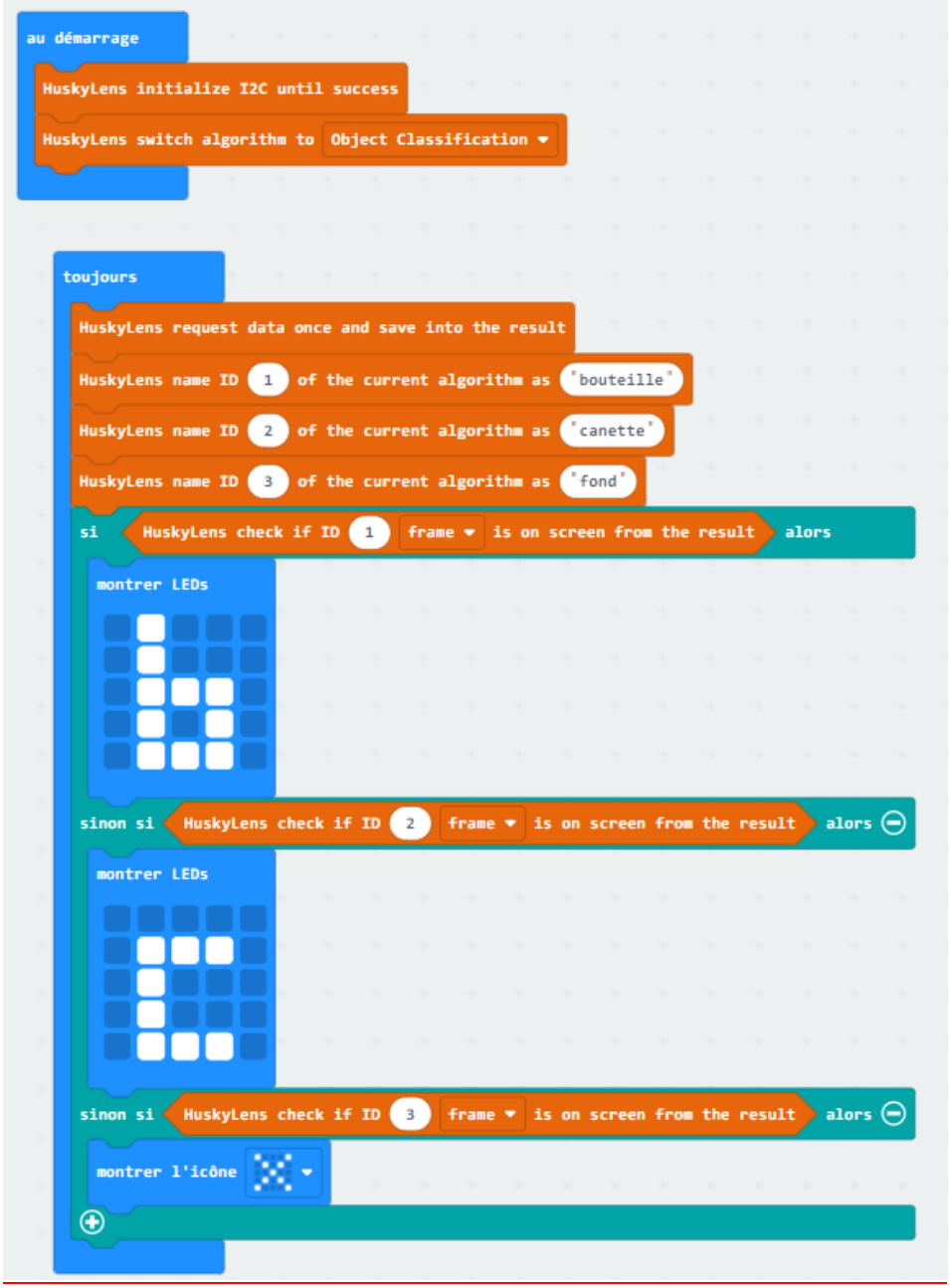
```
    elif huskylens.is_appear(3, HUSKYLENSResultType_t.HUSKYLENS_RESULT_BLOCK):
        basic.show_icon(IconNames.NO)
basic.forever(on_forever)
```

5.1- Complétez les commentaires et expliquez en quelques lignes le fonctionnement de ce programme :

5.2- Réalisation du programme

Réaliser à l'aide du lien suivant https://makecode.microbit.org/# le programme bloc ci-dessous, puis cliquer sur télécharger, récupérer le fichier .hexa et le coller dans la carte Microbit. Tester à l'aide » de la caméra.

Appelez le professeur pour vérification !!!



5.3- Réalisez une brève conclusion sur la possibilité d'identifier différents types de déchets et de les trier :