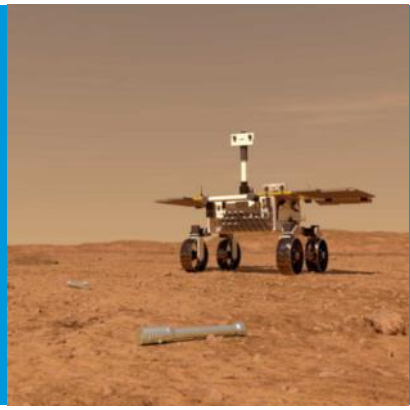
Belgium esero

Fiche pour l'enseignant

Reconnaissance d'images

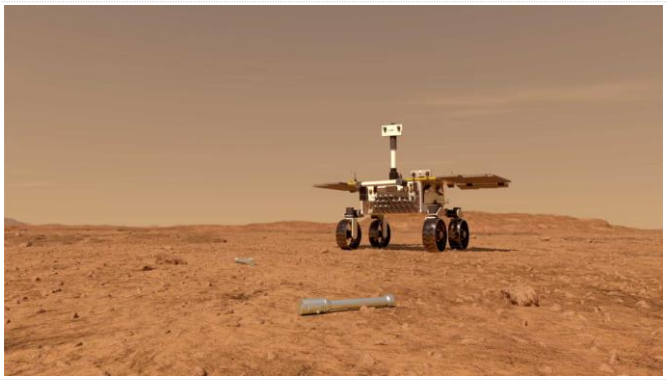
Phase 2 / Avec Python



Une fois que les élèves ont entraîné un modèle de reconnaissance d'images avec la Teachable Machine (voir fiche « Reconnaissance d'images, phase 1 – Entraînement du modèle »), cette activité montre comment l'utiliser dans un programme en Python pour transmettre des instructions au programme en montrant des images de flèches à la caméra.

esa belspo La Scientothèque L'ÉGALITÉ DES CHANCES PAR LES SCIENCES

Intelligence Artificielle



FetchBot : Rover martien intelligent

Age : 14-18 ans

Reconnaissance d'images avec Python

Objectifs :

- ✓ Déplacer la tortue Python avec de la reconnaissance d'images

Notions abordées : Intelligence artificielle, classification d'images, programmation avec Python, condition, boucle.

- Phase 1 : La tortue Python
- Phase 2 : Prendre une image avec la caméra
- Phase 3 : Classification de l'image
- Phase 4 : Contrôle de la tortue avec la reconnaissance d'images

Durée : 4h00

Dispositif pédagogique : par groupes de 2

Matériel

- Un laptop/tablette par groupe de 2 avec connexion à Internet.

Prérequis

1. Connaissances de bases de Python.
2. Avoir créé un modèle de reconnaissance d'images avec la *Teachable Machine* de Google (Activité 1 - Reconnaissance d'images avec la *Teachable Machine*)

Références & liens utiles

- Documentation pour la tortue Python : https://docs.python.org/fr/3/library/turtle.html

Droits d'auteur

Le contenu de cette fiche pédagogique est publié sous licence Creative Commons Attribution - Pas d'utilisation commerciale - Partage dans les mêmes conditions (CC-BY-NC-SA) :

Attribution [BY] (*Attribution*) : l'œuvre peut être librement utilisée, à la condition de l'attribuer à l'auteur en citant son nom : La Scientothèque. Cela ne signifie pas que l'auteur est en accord avec l'utilisation qui est faite de ses œuvres.

Pas d'utilisation commerciale [NC] (*Noncommercial*) : le titulaire de droits peut autoriser tous les types d'utilisation ou au contraire restreindre aux utilisations non commerciales (les utilisations commerciales restant soumises à son autorisation). Elle autorise à reproduire, diffuser, et à modifier une œuvre, tant que l'utilisation n'est pas commerciale.

Partage dans les mêmes conditions [SA] (*ShareAlike*) : le titulaire des droits peut autoriser à l'avance les modifications ; peut se superposer l'obligation (SA) pour les œuvres dites dérivées d'être proposées au public avec les mêmes libertés que l'œuvre originale (sous les mêmes options Creative Commons).

Description détaillée

Un protocole à destination des jeunes est disponible en ligne. L'activité peut donc être réalisée en autonomie par les élèves.

Phase 1 : La tortue Python

Vos élèves vont commencer par utiliser la tortue Python. La tortue Python est une sorte de robot qu'ils vont pouvoir déplacer dans une fenêtre graphique.

1.1. Créez la tortue

Pour permettre à Python d'utiliser la tortue, vous devez d'abord importer la librairie 'turtle' (tortue en anglais), et ensuite exécuter la commande `screen = turtle.getscreen()` pour ouvrir la fenêtre graphique.

Ouvrez l'éditeur Python Thonny, et créez un nouveau fichier en cliquant sur l'icône '+'. Copiez le code ci-dessous :

```
# Import turtle library
import turtle
# Open screen and display turtle
screen = turtle.getscreen()
# Create a turtle object
my_turtle = turtle.Turtle()
```

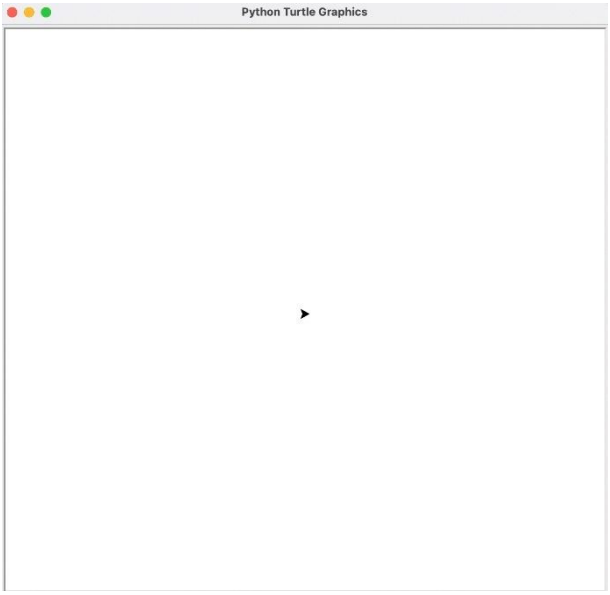
Remarque : Tous les textes après les '#" sont des commentaires, et vous n'êtes pas obligé de les recopier. Ils aident cependant à comprendre ce que fait le code.

Sauvegardez le fichier sous le nom '1-turtle.py'.

1.2. Testez le code

Les élèves sont maintenant prêts à exécuter le code. Dites-leur de cliquer sur le bouton 'Run' de l'éditeur Thonny. Si le code ne s'exécute pas correctement, il se peut qu'il y ait une erreur dans le code qui a été écrit.

En principe, les élèves devraient maintenant voir une fenêtre s'ouvrir, avec la tortue qui est placée au centre. La tortue est représentée par une tête de flèche.



1.3. Dirigez la tortue

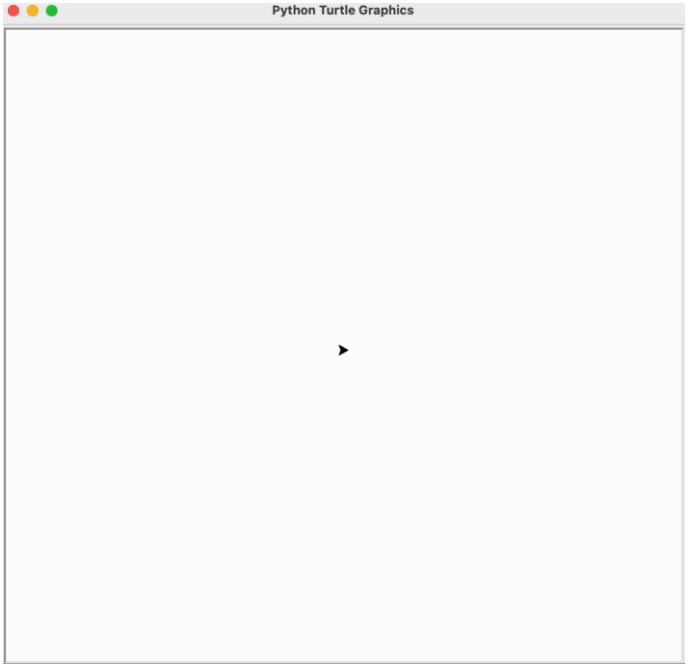
Vous pouvez diriger la tortue avec les commandes :

- `my_turtle.forward(x)` : Pour avancer de x pixels
- `my_turtle.backward(x)` : Pour reculer de x pixels
- `my_turtle.right(x)` : tourner à droite de x degrés
- `my_turtle.left(x)` : tourner à gauche de x degrés

Par exemple, pour avancer de 100 pixels, tourner de 90 degrés, puis avancer de 100 pixels, vous pouvez utiliser le code suivant :

```
# Import turtle library
import turtle
# Open screen and display turtle
screen = turtle.getscreen()
# Create a turtle object
my_turtle = turtle.Turtle()

my_turtle.forward(100)
my_turtle.right(90)
my_turtle.forward(100)
```

Copiez ce code dans l'éditeur Thonny et exécutez-le. Vous devriez voir la tortue suivre le trajet correspondant :



Demandez aux élèves quel code ils pourraient utiliser pour faire faire un carré à la tortue.

Phase 2 : Prendre une image avec la caméra

Le code suivant permet de prendre une image avec la caméra. Exécutez-le pour voir apparaître l'image prise de la Webcam.

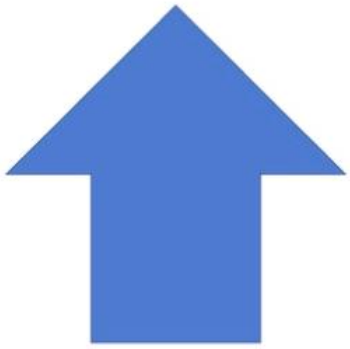
```
# Necessary import
import cv2 # cv2 is used to take image from the camera
import myfunctions # myfunctions helps for taking picture and making
predictions
from PIL import Image # PIL is used to deal with images
import time # time is used for making the computer wait

# Get camera object
camera_object = cv2.VideoCapture(0)

# Take a picture from the camera
picture_rgb = myfunctions.take_picture(camera_object)

# Display picture
picture_as_image_object = Image.fromarray(picture_rgb)
picture_as_image_object.show(picture_as_image_object)
```

Phase 3 : Classification de l'image

Pour classifier l'image, les élèves auront besoin d'un modèle de reconnaissance d'image créé avec la *Teachable Machine* (cf. activité 1 pour créer ce modèle).

3.1. Entraînement du classificateur

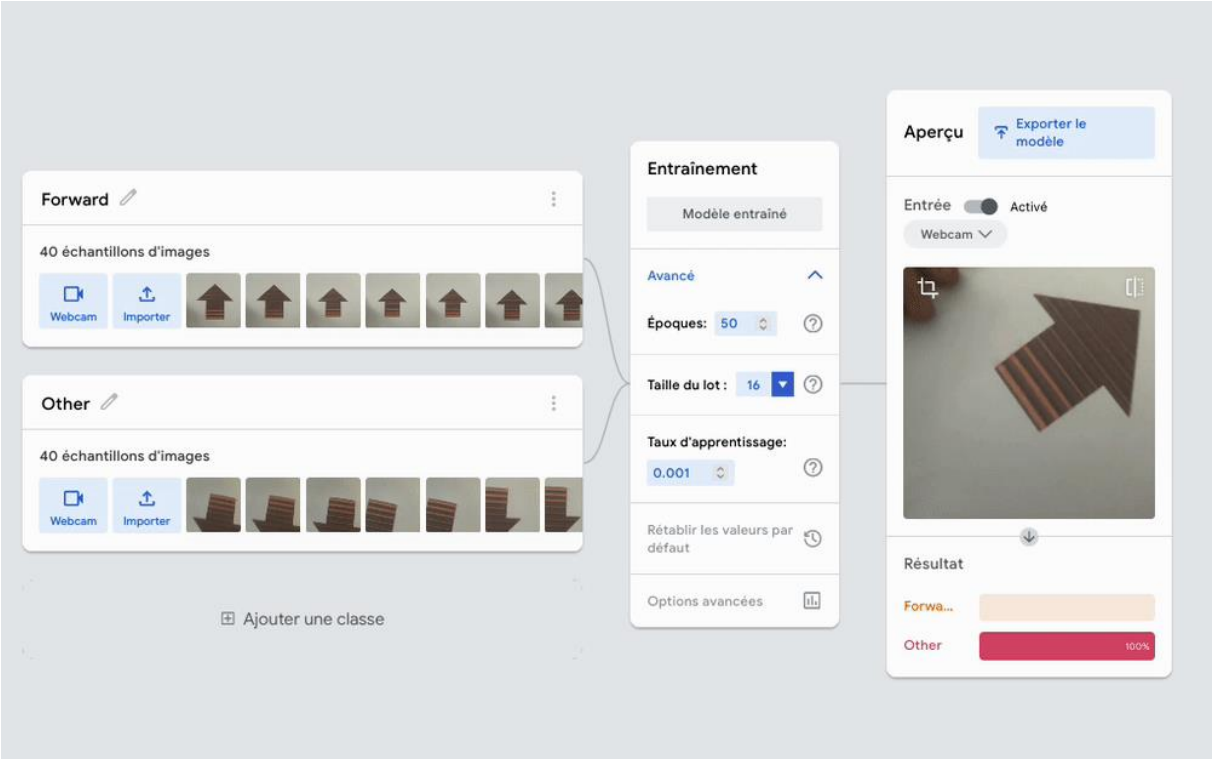
Pour contrôler la tortue, les élèves peuvent par exemple utiliser une flèche imprimée sur un papier, comme la flèche ci-dessous :



Créez ensuite un classificateur avec deux classes dans la *Teachable machine*, que vous nommerez :

- Forward : Avec des exemples de flèche vers le haut (qui serviront à faire avancer la tortue)
- Other : Avec des exemples où la flèche ne va pas vers le haut

Prenez une quarantaine d'images pour chacune des deux classes, et entraînez le classificateur. Testez-le ensuite, le classificateur devrait reconnaître quand la flèche est orientée vers le haut (classe 'forward'), comme ci-dessous :



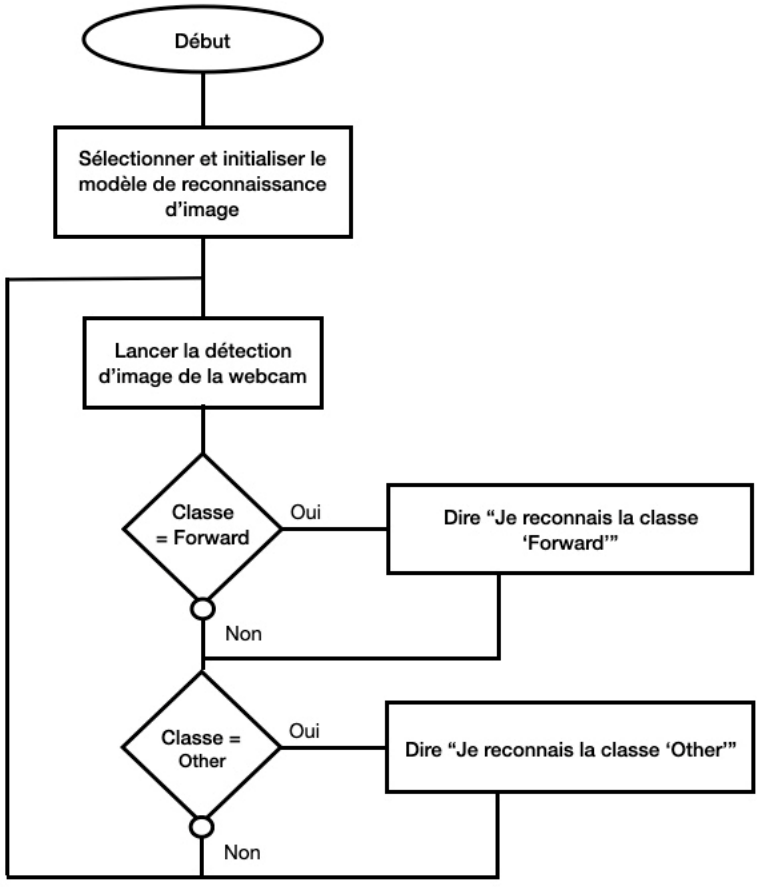
Exportez ensuite le modèle pour Python (Voir activité 1), et copiez le fichier model.tflite dans le répertoire de vos fichiers Python.

3.2. Classification de l'image

Pour la classification d'image, nous allons reprendre le code pour prendre une image avec la caméra, et y ajouter le fait de prédire à quelle classe appartient l'image. Trois ajouts sont à faire

1. Charger (initialiser) le modèle de reconnaissance d'images
2. Lancer la détection d'image prise de la webcam
3. Afficher "Je reconnais la classe 'Forward'" si la classe est 'Forward', et afficher "Je reconnais la classe 'Other'" si la classe est 'Other'

Enfin, on placera une boucle pour répéter la détection d'image et l'affichage. L'algorithme est donné ci dessous:



Et voici la traduction en langage Python:

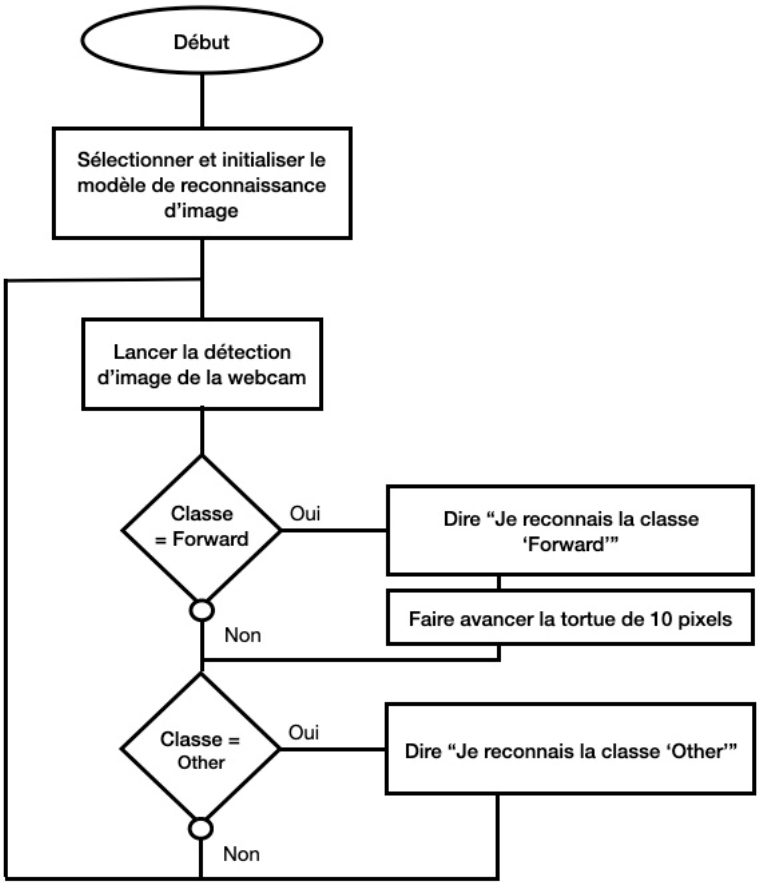
```
# Necessary import
import cv2 # cv2 is used to take image from the camera
import myfunctions # myfunctions helps for taking picture and making
predictions
```

```
import matplotlib.pyplot as plt # matplotlib is used to deal with
images
from PIL import Image # PIL is used to deal with images
import time # time is used for making the computer wait

# Get camera object
camera_object = cv2.VideoCapture(0)

# Initialize model
interpreter = myfunctions.initialize_model(model_path='model.tflite')

# Infinite loop
while True:

    # Wait for one second
    time.sleep(1)

    # Take image from the camera
    picture_rgb = myfunctions.take_picture(camera_object)

    # Predict image class
    prediction, probability = myfunctions.model_prediction(interpreter,
picture_rgb)

    # If prediction is class 0, class is 'Forward'
    if prediction == 0:

        print("Je reconnais la classe 'Forward'")

    # If prediction is class 1, class is 'Other'
    if prediction == 1:

        print("Je reconnais la classe 'Other'")
```

Vous pouvez copier le code ci-dessus dans Thonny, et le lancer. Vous verrez à chaque nouvelle seconde s'afficher la classe reconnue par le modèle.

Phase 4 : Contrôle de la tortue avec la reconnaissance d'images

Vous pouvez maintenant modifier le programme pour faire avancer la tortue de 10 pixels quand la classe est 'Forward'. Il suffit de rajouter la commande `my_turtle.forward(10)` lorsque la classe détectée est forward. Voici l'algorithme correspondant:



Et la traduction en langage Python:

```
Necessary import
import turtle # For using Python turtle
import cv2 # cv2 is used to take image from the camera
import myfunctions # myfunctions helps for taking picture and making
predictions
import matplotlib.pyplot as plt # matplotlib is used to deal with
images
from PIL import Image # PIL is used to deal with images
import time # time is used for making the computer wait

# Open screen and display turtle
screen = turtle.getscreen()
# Create a turtle object
my_turtle = turtle.Turtle()

# Get camera object
camera_object = cv2.VideoCapture(0)

# Initialize model
interpreter = myfunctions.initialize_model(model_path='model.tflite')

# Infinite loop
while True:
```

```
    # Wait for one second
    time.sleep(1)

    # Take image from the camera
    picture_rgb = myfunctions.take_picture(camera_object)


    # Predict image class
    prediction, probability = myfunctions.model_prediction(interpreter,
picture_rgb)

    # If prediction is class 0, class is 'Forward'
    if prediction == 0:

        my_turtle.forward(10)

# Release camera
camera_object.release()
```

Vous pouvez copier le code ci-dessus dans Thonny, et le lancer. Vous verrez à chaque nouvelle seconde s'afficher la classe reconnue par le modèle, et la tortue avancer de 10 pixels quand la classe est 'Forward'.

Phase 5 : Aller plus loin

- Ajoutez des classes avec des images de flèches vers la droite ou la gauche pour faire tourner la tortue à droite ou à gauche.
- Faites un classificateur qui utilise l'image de tube sur sol martien, faites avancer la tortue si aucun tube n'est détecté et faites-la s'arrêter lorsqu'un tube est détecté.