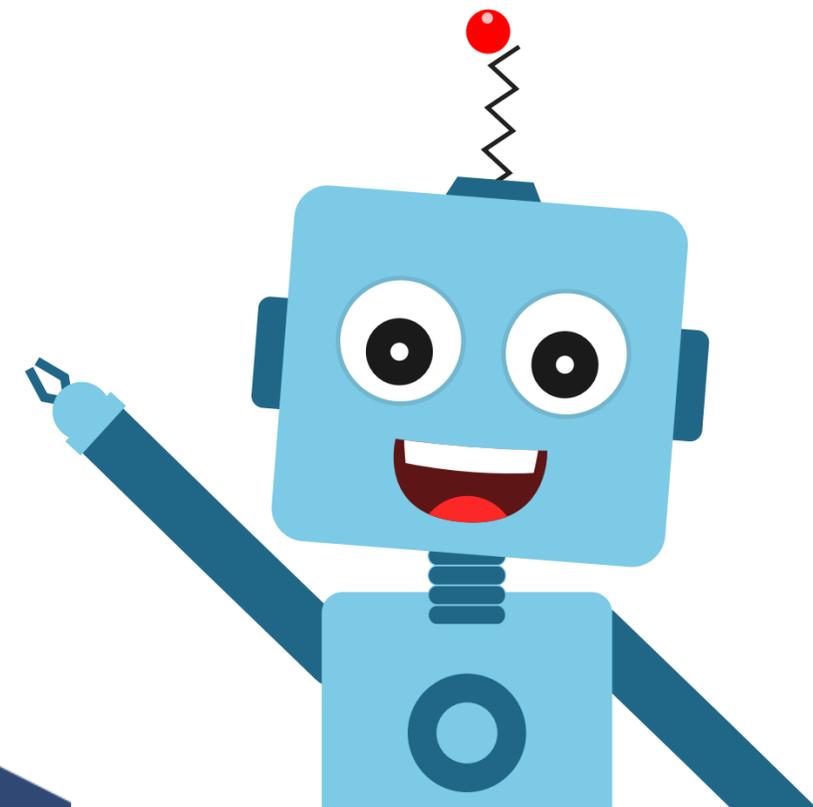


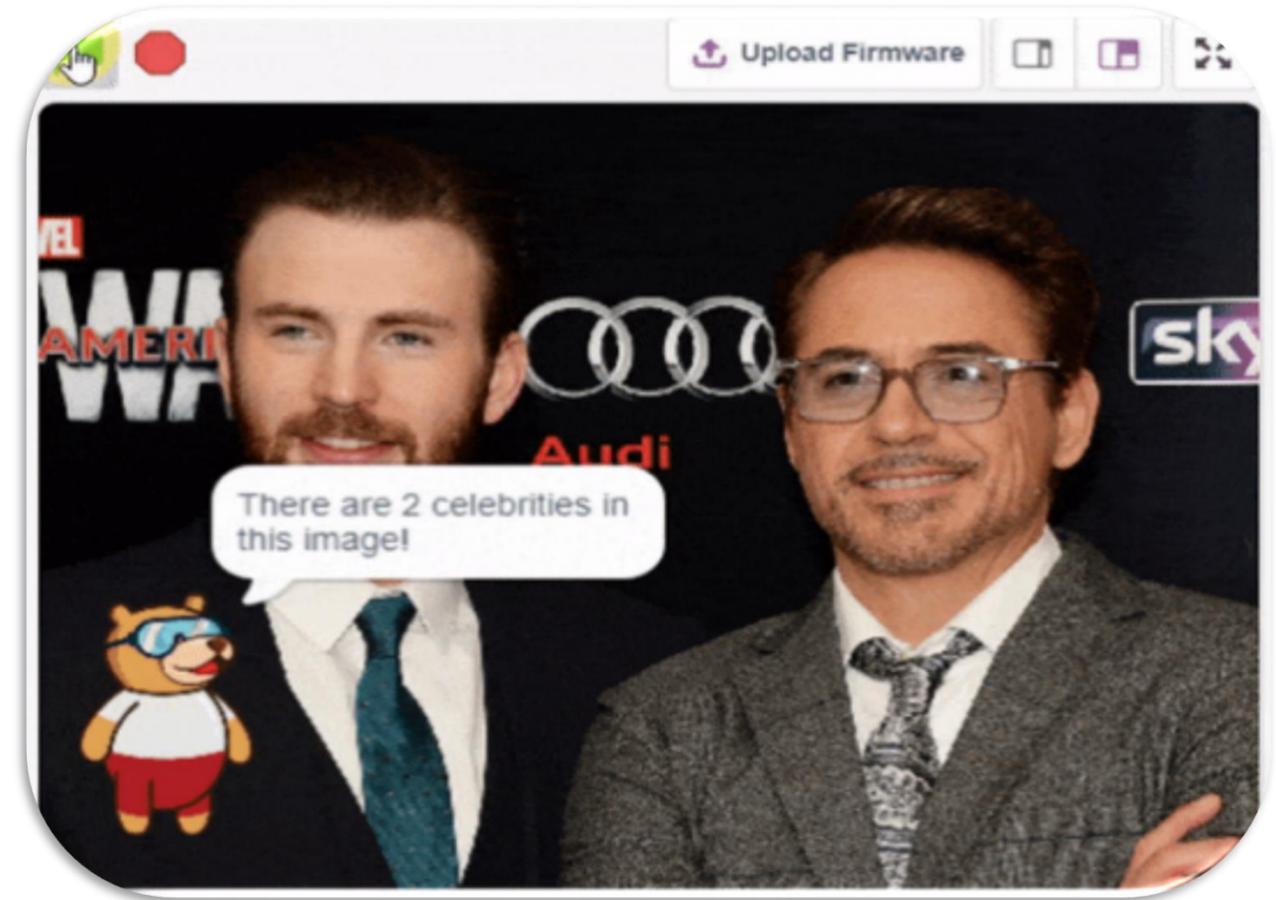
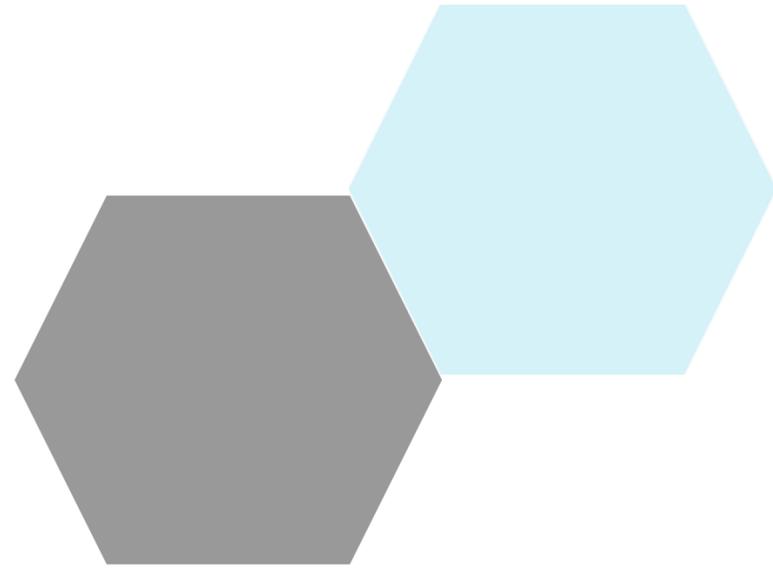
Computer Vision object detection part-2

Session 18

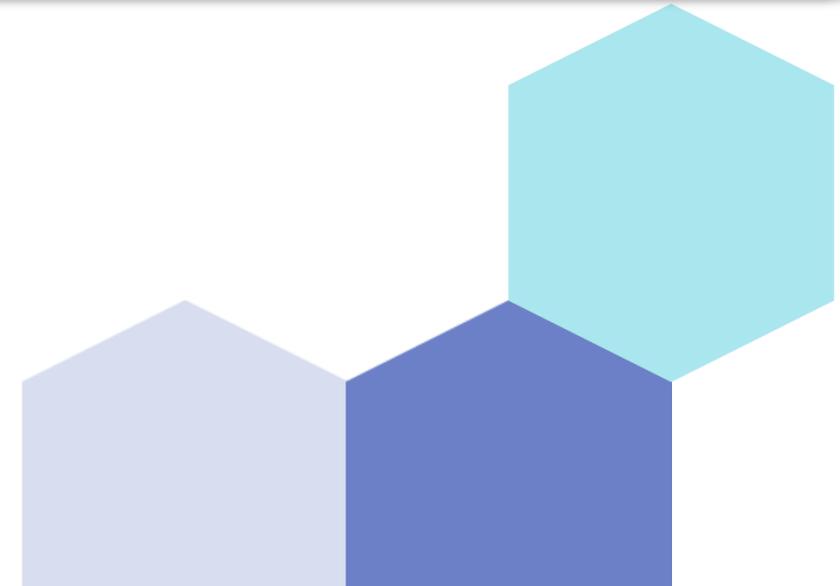


Topics covered

- Introduction to computer vision
- Application of computer vision
- Activity celebrity detection



Computer Vision



Introduction to Computer Vision

PictoBlox Python offer computer vision capabilities that deals with how computers can be made to gain a high-level understanding of digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human vision can do. Using this library, you can create projects to identify celebrities, brands, landmarks, and objects in images.

By analyzing the images for image features, you can recognize the following things:

- **Brands:** Brand detection uses a database of thousands of global logos to identify commercial brands in images. The Computer Vision service detects if there are brand logos in a given image; if so, it returns the brand name; else, it returns NULL.



Brands



Celebrity



Input Image

- **Celebrity:** Celebrity detection uses a database to identify celebrities in images. The Computer Vision service detects if there is a celebrity in a given image; if so, it returns their name; else, it returns NULL.
- **Objects:** Computer vision detects if there are objects in a given image; if so, it returns their name; else, it returns NULL

Introduction to Computer Vision

- **Landmarks:** Landmark detection uses a database of thousands of global landmarks to identify them in images, e.g., the Taj Mahal.
- **Image Description:** Human-readable sentence that describes the contents of the image
- **Image Tags:** Computer vision returns the taxonomy-based categories detected in an image. Computer Vision can categorize an image broadly or specifically according to the 86 categories.



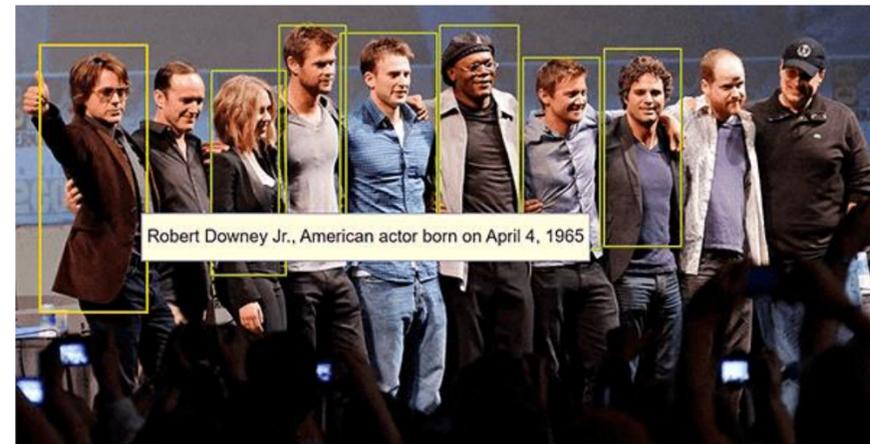
Landmarks

In this activity, we will write a code that will perform Celebrity Detection on the PictoBlox stage. We will upload an image containing one or more celebrities as a PictoBlox backdrop. The code we run to analyze this stage will detect the following things from the image:

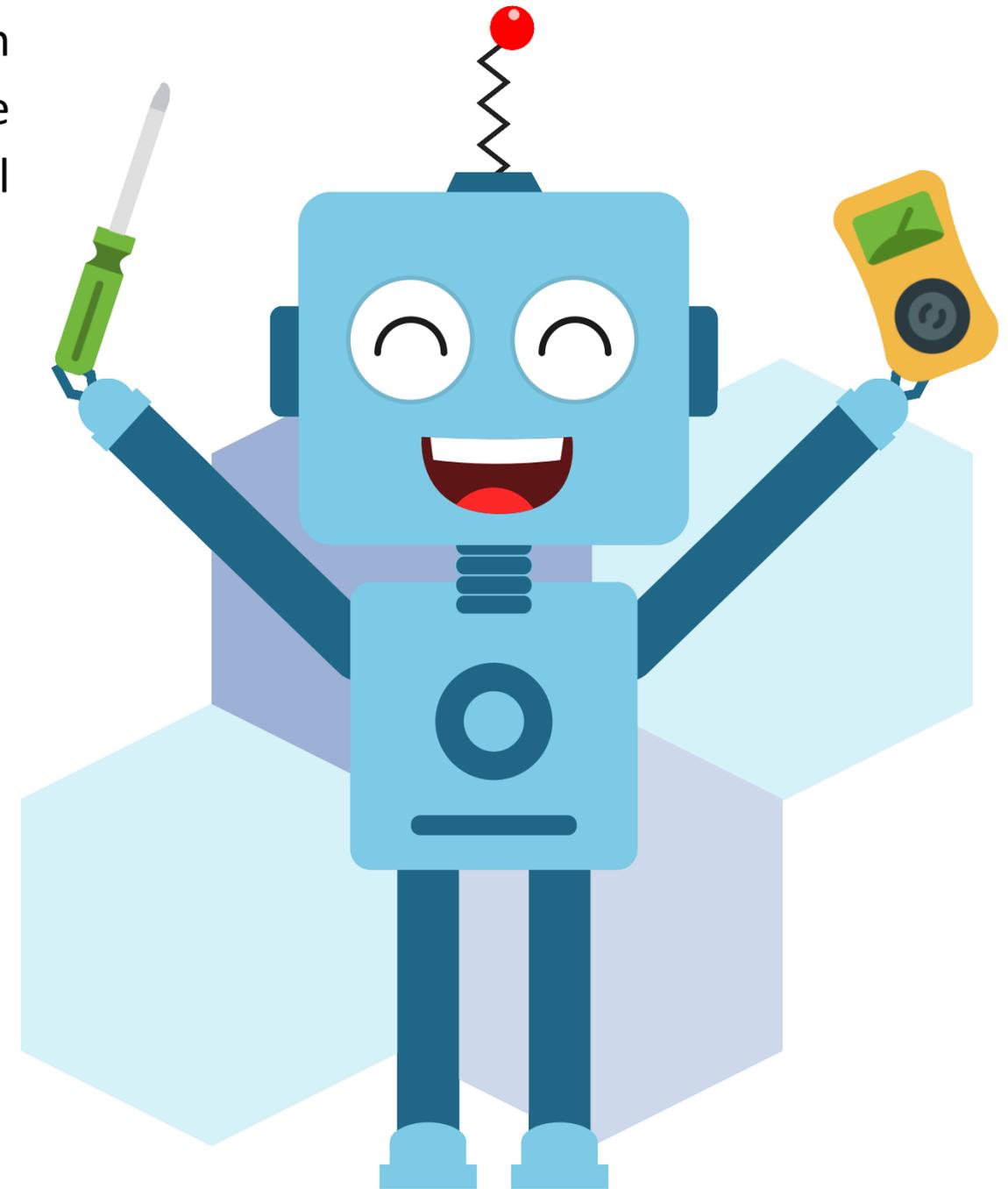
- Face of celebrity, x-position, y-position, and height of face
- Celebrity name
- Confidence (in %) with which the algorithm has detected the celebrity



Input Image

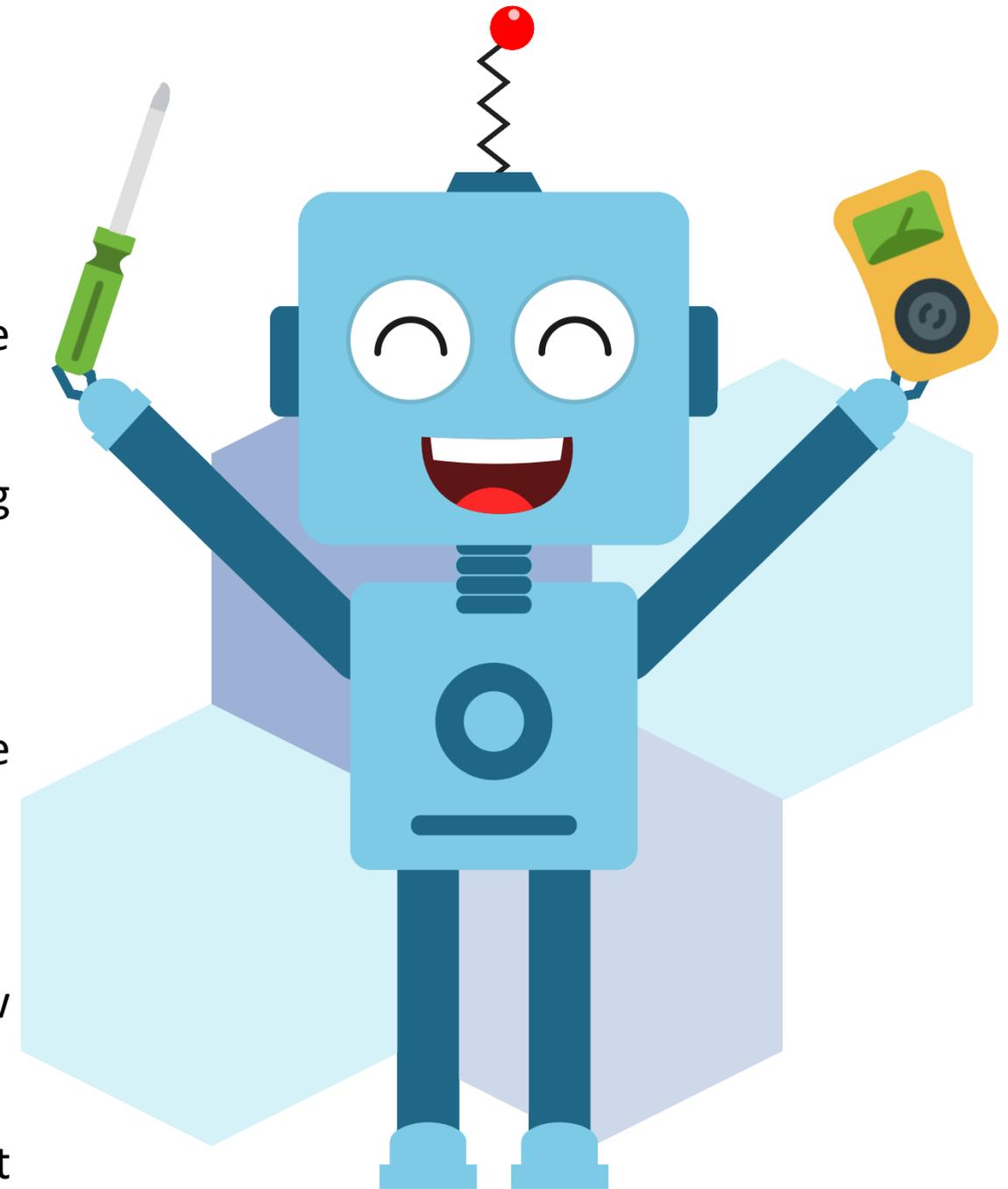


Celebrity Detection

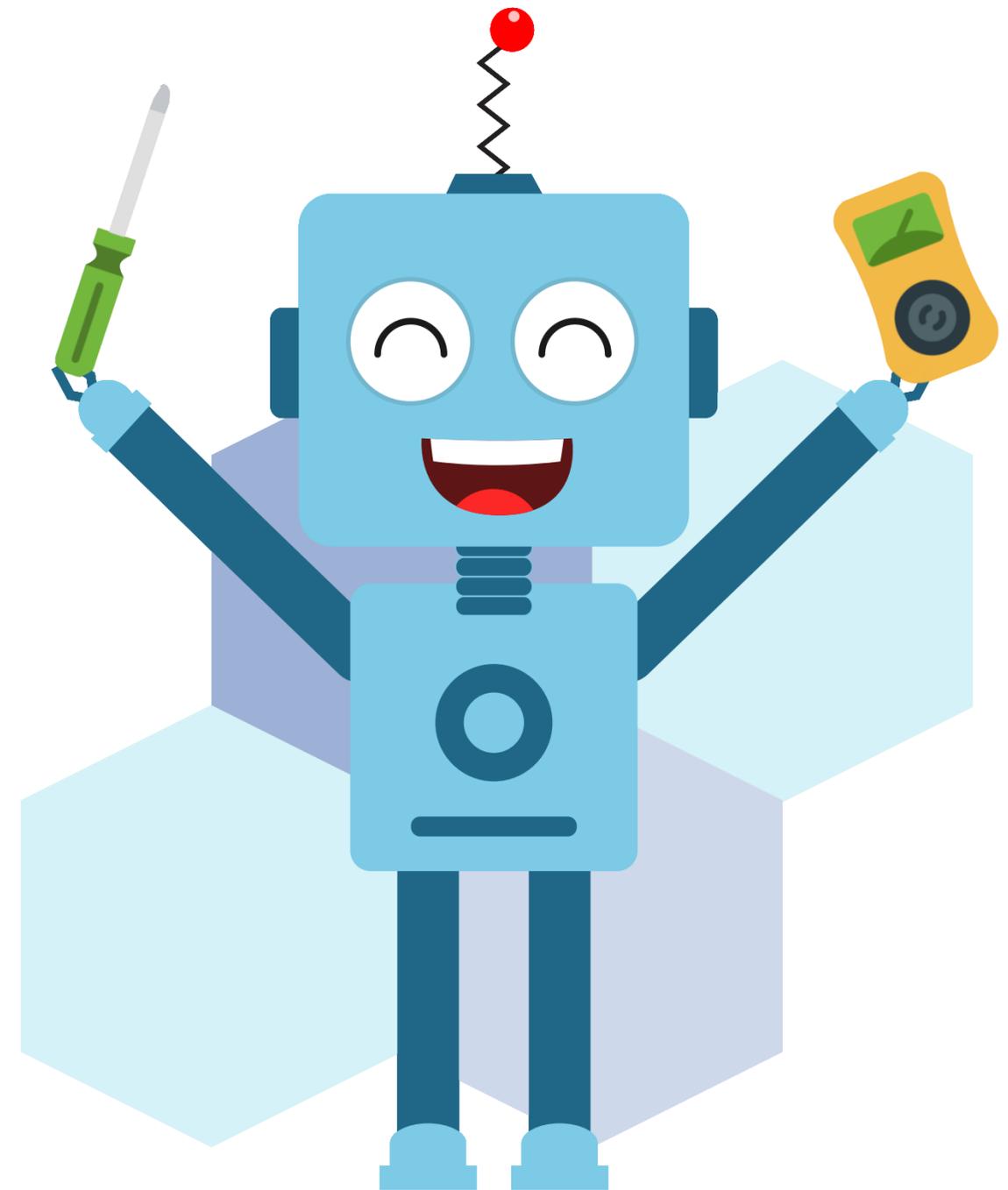


STEP-BY-STEP PROCEDURE

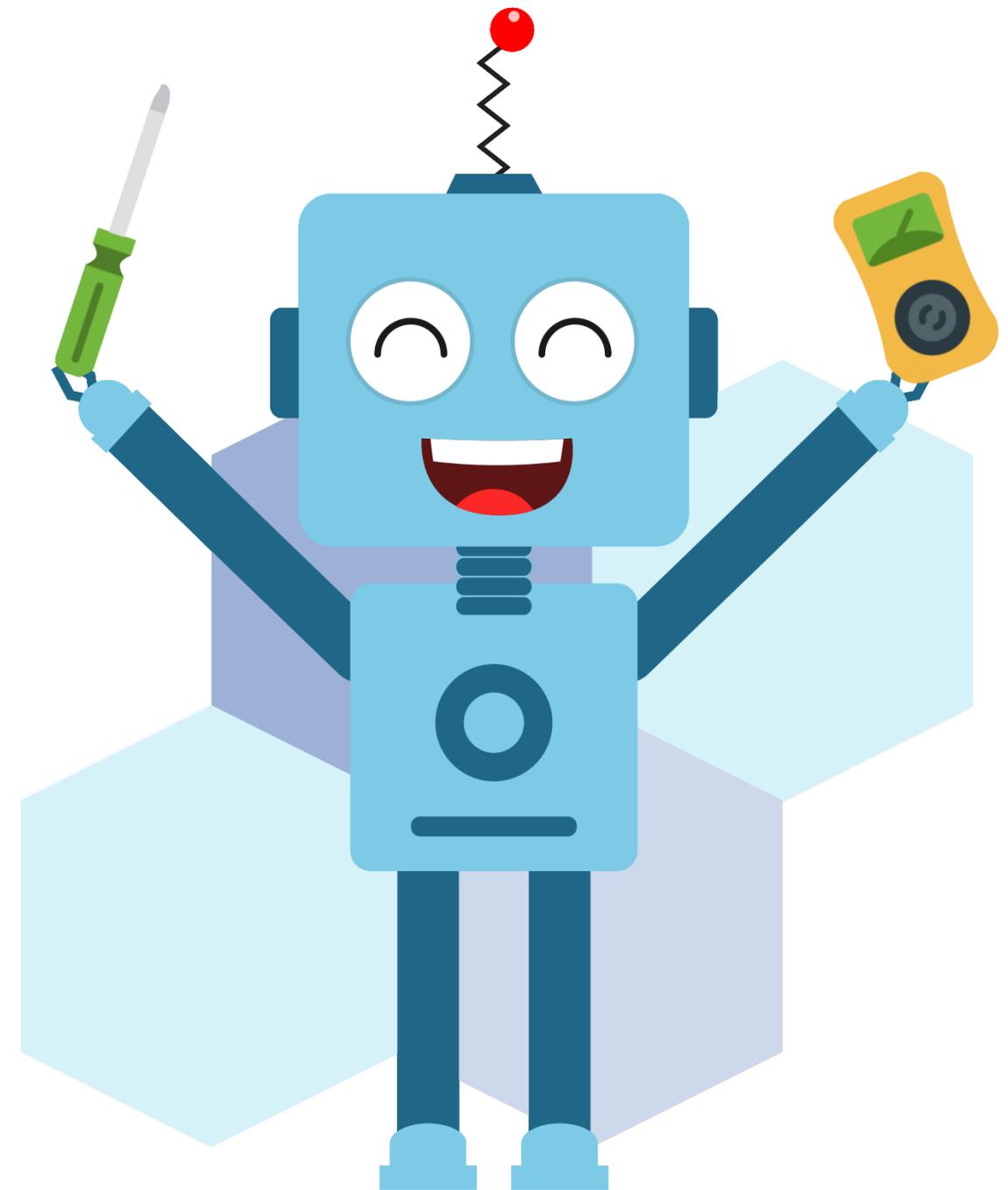
- Follow the steps below:
- Open a **new file** in **Pictoblox** and choose the **Python Coding** environment.
- Click on **Choose a sprite** button (bottom right corner) and select the sprite **Square Box**.
- For this, first upload this image as a backdrop in the PictoBlox by following the below steps:
 - Scan this QR code for input image as shown here. Save the image.
 - Then Choose a **Backdrop** button (bottom right-hand corner) and choose the **Upload Backdrop** option.
 - Select the image you just downloaded and click **Open**.
 - Click on **Square Box.py** from the **Project Files**. The image is now uploaded as backdrop.
- The sprite object 'Square Box' is already initiated by default. Let us import the time library.



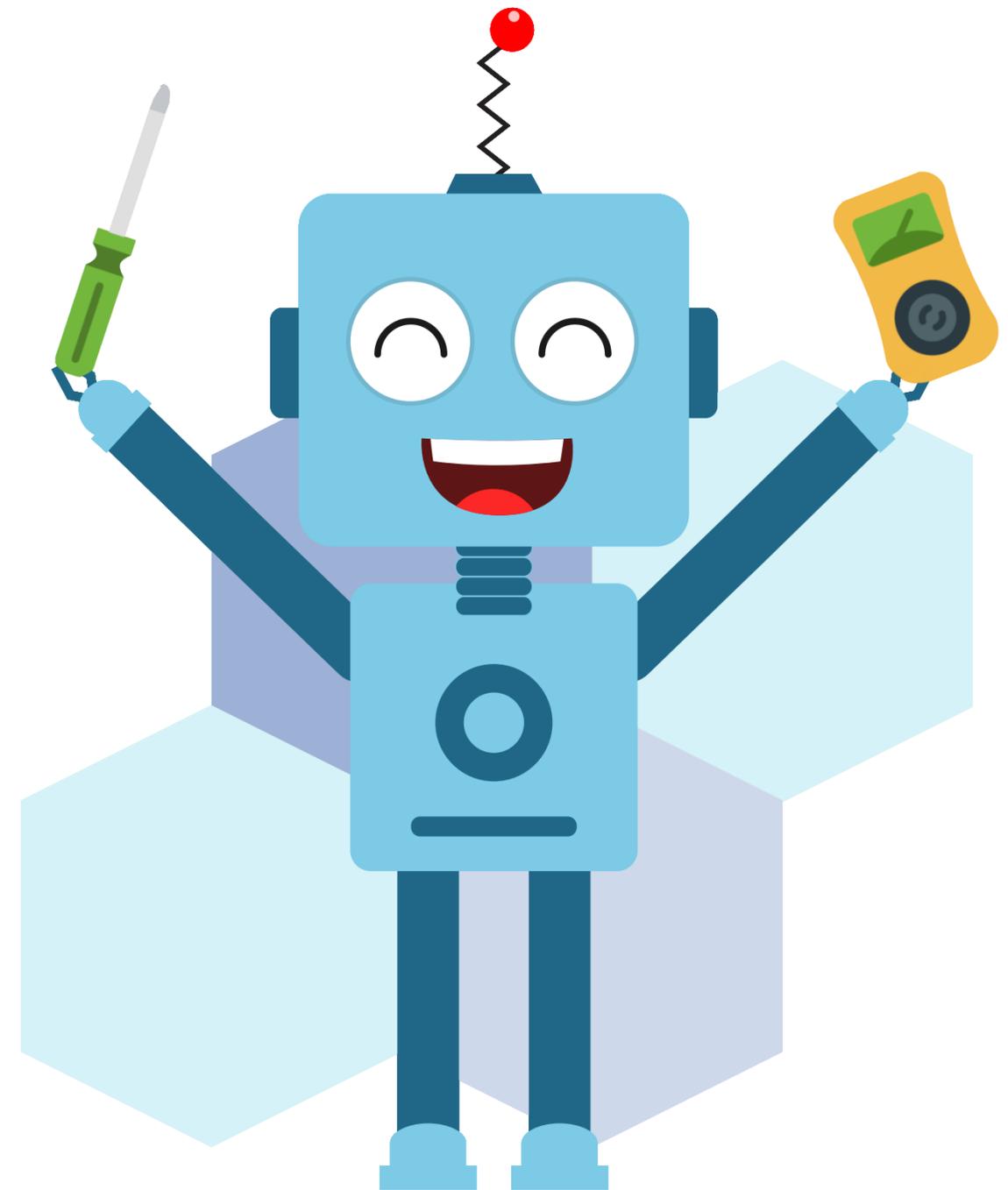
- Let us now also initiate the Computer Vision class, so that we can use the functions from this class.
- In order to analyze the backdrop image that we have added for celebrity detection we use the function **analysebackdrop([1])** from **Computer Vision** class and choose the parameter “celebrity”.
- Now, we will calculate the number of celebrities present in the backdrop image using `imagefeaturecount()` function. This function returns a string, so we will convert it to integer using **int()** function of python and store it in a variable.
- We want to detect all the celebrities in the image, one-by-one. Hence, we will use a for loop to go through all of them by keeping the range from 1 to `num_celebrities + 1`. We add ‘+1’ since the end parameter of **range()** function is excluded.
- In order to extract the x-position of the celebrity face, we use the `imagefeatureinfo()` function from the **Computer Vision** class and add the appropriate parameters as follows:
 - **[1]** = “celebrity” (for what we want to detect)
 - **[2]** = i (for the number of celebrity)
 - **[3]** = “xPos” (for x-position)



- We enclose the above function inside the **setx()** function from **Sprite** class to set the x-position of the sprite.
- Using similar steps, we used in the above point; we can set the y-position of the sprite equal to the celebrity face.
- Here, again we will use the `imagefeatureinfo()` function with the difference in parameter [3] = “yPos” (for y-position).
- We will enclose this function inside the **sety()** function to set the y-position of the sprite.
- Using similar steps, we used in the above point; we can set the size of the sprite equal to the height of the celebrity face.
- Now that we have displayed the name of celebrity, we want to pause the display for 1 second before we display the confidence. We use **sleep()** function of time module for this.
- In AI, “Confidence” is **the probability with which the machine learning algorithm is sure of its prediction**. We will calculate the confidence of our Computer Vision algorithm on its detection of a particular celebrity, and store it in a variable. For this, we will use the following functions:

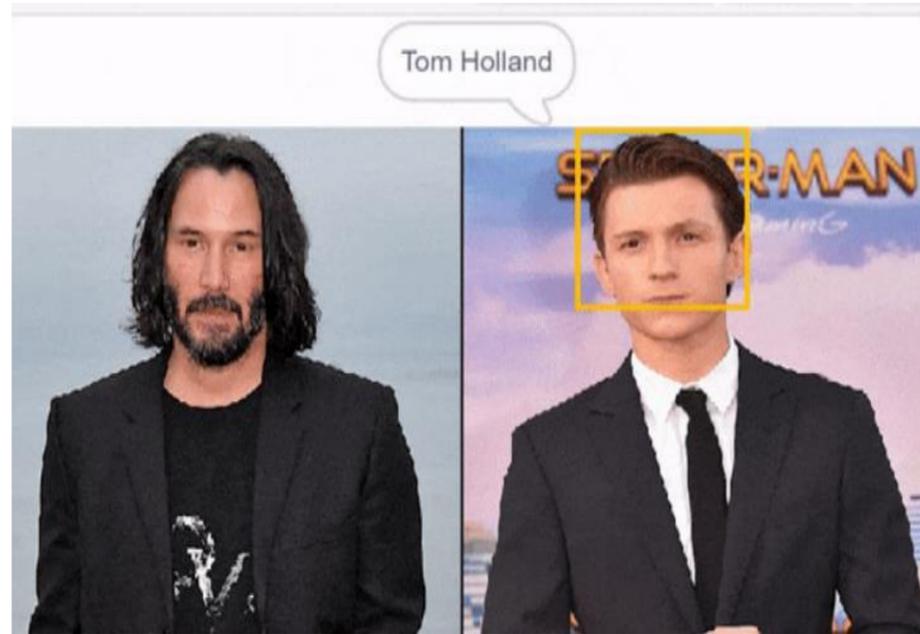


- We use the function `imagefeatureinfo()` with the same parameters as above except for parameter [3] where we add [3] = “confidence”.
- The return value of the function `imagefeatureinfo()` is a string. So we convert it to a float using the **`float()`** in-built function of python.
- We enclose the above code in the first parameter [1] of the in-built function `round([1], 4)` of python, to keep the number of decimal places limited to ‘4’.
- Now, we will make the sprite say the ‘confidence percentage’ by multiplying the **confidence** value by **100**. We will also convert the ‘confidence percentage’, from number to a string using the inbuilt function **`str()`** of python.
- Finally, we add a **`time.sleep()`** function to add a pause between the detection of two celebrities.
- Press Run to run the code.

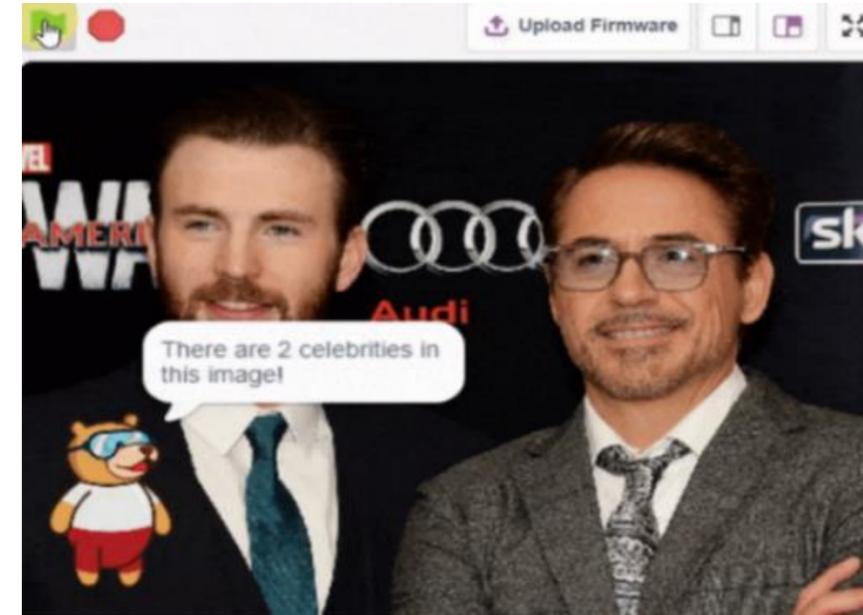


```
sprite = Sprite('Square Box')
import time
obj = ObjectDetection()
pen = Pen()
cv = ComputerVision()
cv.analysebackdrop("celebrity")
num_celebrities = int(cv.imagefeaturecount("celebrity"))

for i in range(1, num_celebrities + 1):
    sprite.setx(cv.imagefeatureinfo("celebrity", i, "xPos"))
    sprite.sety(cv.imagefeatureinfo("celebrity", i, "yPos"))
    sprite.setsize(cv.imagefeatureinfo("celebrity", i, "height"))
    sprite.say(cv.imagefeatureinfo("celebrity", i))
    time.sleep(1)
    confidence = round(float(cv.imagefeatureinfo("celebrity", i, 'confidence')), 2)
    sprite.say("Confidence = " + str(confidence * 100) + "%")
    time.sleep(2)
```



Final Output

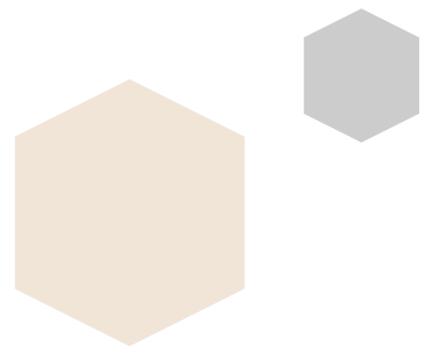
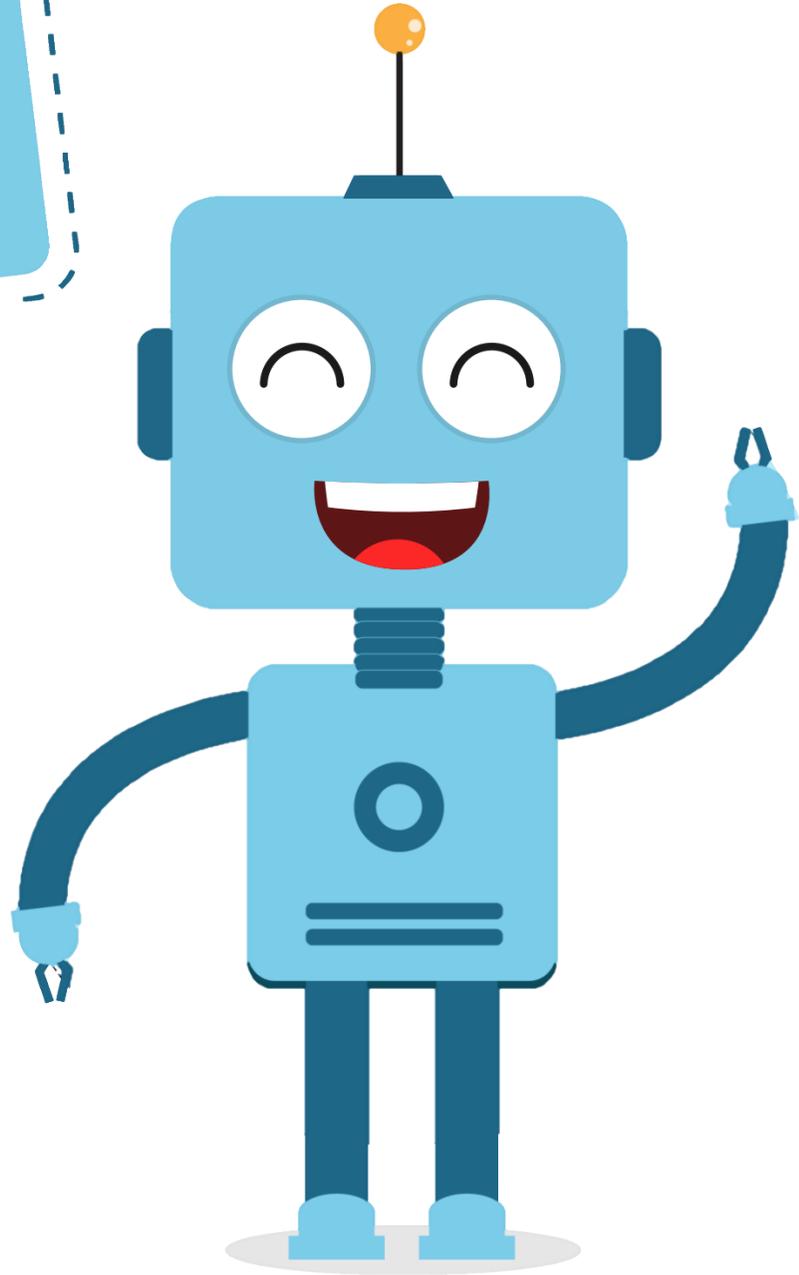


Output

SAVE THE PROGRAM

Save the project file as **Object Detection**.

**THANK
YOU**



POWERED BY
STEMpedia

