THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

DEPARTMENT OF
COMPUTER SCIENCE
Au Reikura

2025

Contributors

J. Turner

J. Kasmara

BUDDY IS ARTIFICIAL AND HAS INTELLIGENCE

Welcome to the eight sessions of the Artificial Intelligence Series. In today's session, we will conclude the series by learning that Buddy is artificial and has intelligence. For all the artificial intelligence enthusiasts out there, this is an exciting time to be joining the Computer Science field.

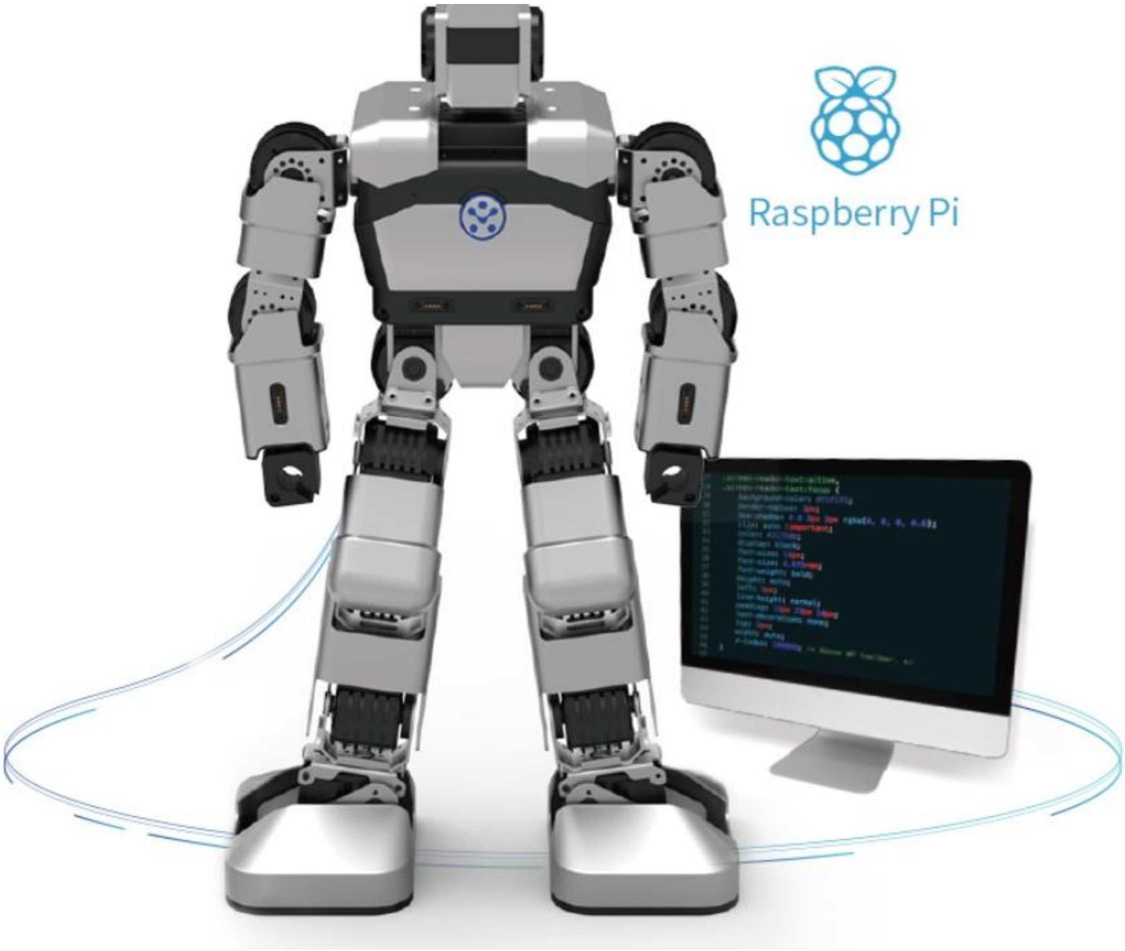


Figure 1: Our humanoid robot, Buddy

In the last session, we introduced you to an open-source software called Blockly. This tool is important because it allows you to see visually what your code is doing before running your program.

Remember three important concepts of Computer Science:

1. **Sequential:** Code runs in sequential order.
2. **Selection:** Variables are initiated, selected and assigned values, especially when computing mathematical expressions.
3. **Loops:** Using loops, we strive to be efficient and avoid repetition, as it consumes unnecessary computing space.

With all this in mind, we have been hard-coding Buddy to perform specific tasks, such as recognising objects he has been trained on or solving mathematical problems using the If-Do-Else logic block. But what if Buddy encounters a new mathematical problem or an object he hasn't been trained on, like your neighbour's cat? How would he respond?

Pause the video and take a few minutes to think about what the answer could be...

Welcome back! If you thought Buddy would display an error message because he wouldn't know what to do, you are right!

Buddy's thought process

Buddy, just like any other robotic interface, is not as intelligent as humans, who can adapt to new situations. Let's visualise what might go through Buddy's thought process.

```
01      > is this a cat?
02              Error:
03              'cat' undefined

04      > Do something new
05              Error:
06              'new' undefined
07      > Arrrrrgggghhh!
08              Error:
```

Line 1 shows that Buddy is questioning, "Is this a cat?". Line 2 will confuse Buddy since a cat is not an object that Buddy has been trained to detect. Buddy delves deeper into his dictionary of training data but finds that "cat" is undefined as shown in Line 3. In real life, when Buddy sees a cat, he would say something like, "I cannot compute, what are you?" And then the cat would stare blankly back at Buddy.

Figure 2: Computer artificial mind.

What is the learning here? Pause the video and take a moment as what might be the key takeaway here…

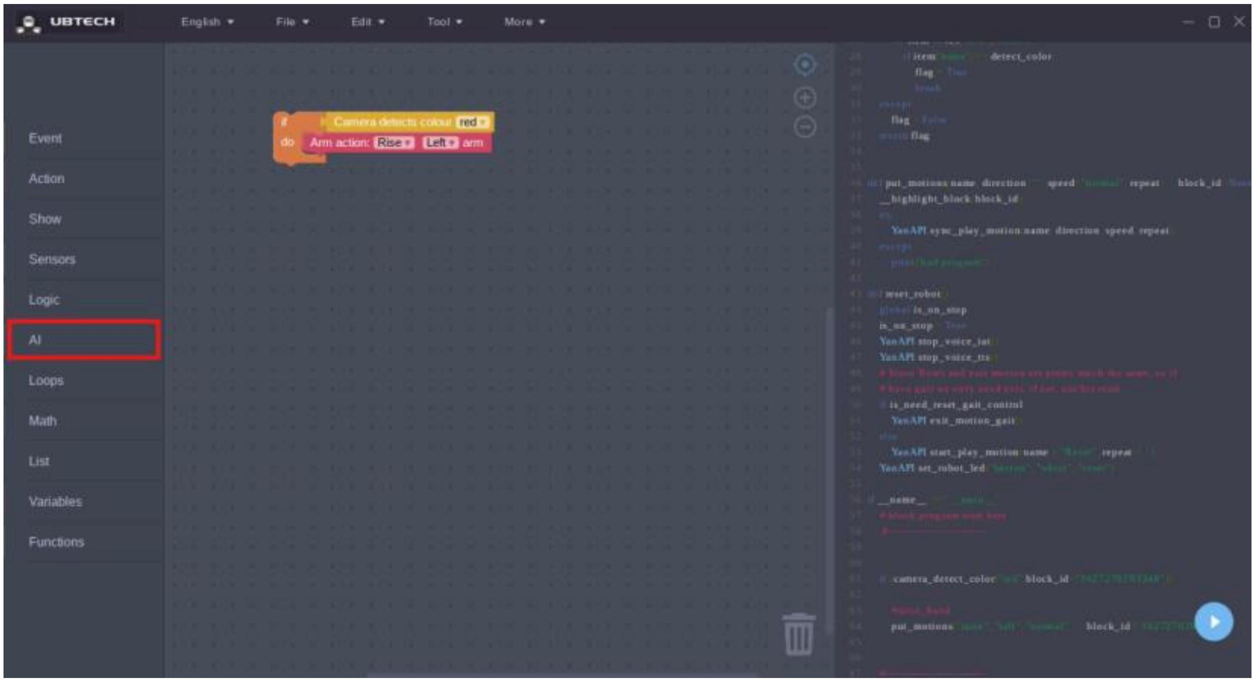
Ans: The learning here, is that a computer only follows the rules it is given

Upgrade

In this final session of the CSNeT Artificial Intelligence Series, you will have the opportunity to train Buddy and similar systems so that computers can adapt to new situations. However, before we begin, let's take a closer look at your chance to be a computer scientist for the day and how we might achieve that.

Buddy, representing a computer, can be upgraded in many ways. Some suggestions include:

- Expanding Buddy's memory size, in other words, adding more RAM.
- Adding additional functions to Buddy, such as an AI module.
- Introducing impressive human-like movements.



```
        if item[illegible] == detect_color
            flag = True
            break
    except
        flag = False
    return flag


def put_motions(name, direction=  , speed="normal", repeat= , block_id=[illegible]
    __highlight_block(block_id)
    try
        YanAPI.sync_play_motion(name, direction, speed, repeat)
    except
        print("bad program")

def reset_robot()
    global is_on_stop
    is_on_stop = True
    YanAPI.stop_voice_iat()
    YanAPI.stop_voice_tts()
    # [illegible]
    # [illegible]
    if is_need_reset_gait_control
        YanAPI.exit_motion_gait()
    else
        YanAPI.start_play_motion(name="Reset", repeat= )
    YanAPI.set_robot_led("button", "white", "on")

if __name__ == "__main__"
    # block program start here
    #--------------------


    if camera_detect_color("red", block_id="[illegible]")

        [illegible]
        put_motions("raise", "left", "normal", , block_id="[illegible]

    #--------------------
```

Figure 3: AI module in Blockly

Introduction to Buddy's Capabilities

Buddy, the humanoid robot, represents a leap in AI-powered education. Designed to teach and engage, Buddy uses AI to mimic human-like behaviours, such as object recognition and adaptive learning. However, Buddy’s intelligence is limited to the instructions and data provided via the AI module in Blockly.

Buddy's training demonstrates the principle that AI systems require robust datasets to function effectively. When confronted with untrained inputs, such as recognising an unfamiliar object, Buddy defaults to error responses. This highlights the critical importance of continuous learning and adaptability in AI systems.

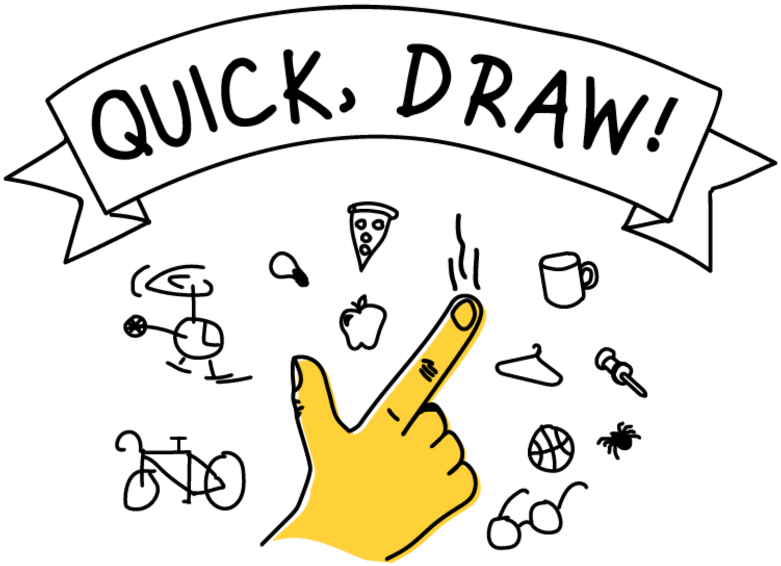


Figure 4: Google Quick, Draw!

Adapting Buddy's Abilities

Expanding Buddy's capabilities involves programming enhancements, such as increasing RAM for better performance, incorporating advanced AI modules, or improving its mechanical movements for realistic engagement. These upgrades allow Buddy to better simulate human interactions and provide a deeper understanding of AI principles.

```
> Load new program



> If
        New animal found:

> Then
        Update existing dictionary to include new animal:
```

Now, we can say that Buddy can adapt to new situations. He has gained intelligence.

PS: In your own time, when Buddy hasn’t come around to your table yet, please download *Google Quick, Draw!* And this is one of the ways to train Buddy in the background with your help to recognise more everyday objects and enhance his object recognition ability.

API

Let's jump into the world of APIs! Imagine an API as a waiter in a restaurant. You place your order with the waiter, who then tells the chef what you want. The chef prepares the food, and the waiter brings it back to you – voilà, your yummy lasagne is ready to enjoy.

In the tech world, an API works in a similar way. Picture yourself looking for a university course on the University of Waikato website. When you search for a course like COMPX101, you use an API to ask the database if that course is available. The API takes your request, checks the database, and then tells you if COMPX101 is open for Trimester A 2026.

See? Just like a helpful waiter, an API makes sure your requests are handled smoothly!

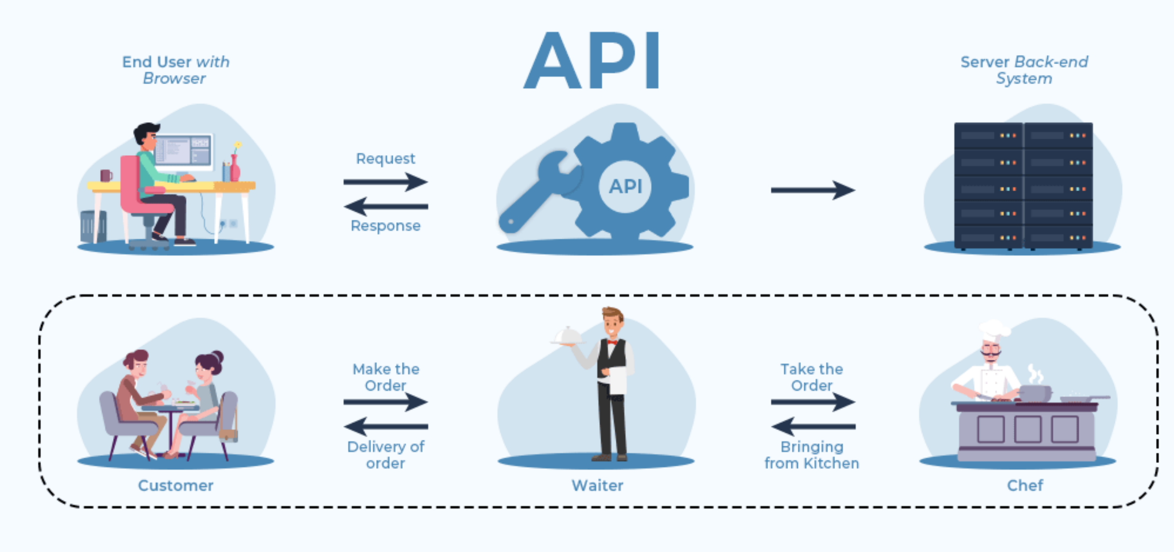
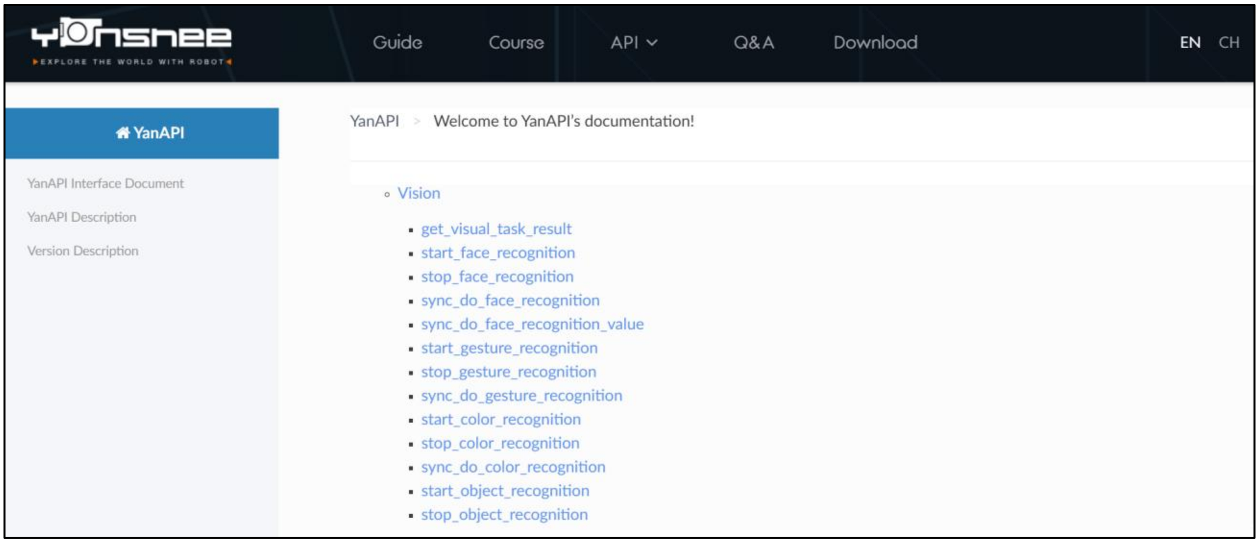


Figure 5: API analogy visual chart

YanAPI

YanAPI is an API for the Python programming language. YanAPI provides the ability to use Python to obtain robot status information as well as to control Yanshee robot, Buddy. The API has pre-programmed the complicated AI functions and allows you as the programmer to call those functions by a single line of code.


yanshee
EXPLORE THE WORLD WITH ROBOT
Guide
Course
API
Q&A
Download
EN
CH
YanAPI
YanAPI Interface Document
YanAPI Description
Version Description
YanAPI > Welcome to YanAPI's documentation!

- Vision
  - get_visual_task_result
  - start_face_recognition
  - stop_face_recognition
  - sync_do_face_recognition
  - sync_do_face_recognition_value
  - start_gesture_recognition
  - stop_gesture_recognition
  - sync_do_gesture_recognition
  - start_color_recognition
  - stop_color_recognition
  - sync_do_color_recognition
  - start_object_recognition
  - stop_object_recognition

Figure 6: Yanshee YanAPI documentation

Here are some of the functions you can try Buddy with:

- **Object Detection:** Identifying objects in images. start_object_recognition
- **Facial Recognition:** Identifying people by their faces. start_face_recognition

Now you can understand all the different tools we need to program Buddy to adapt to new situations.

Programming Buddy to learn ADAPTABILITY and addressing ALGORITHM BIASES

Time to play! Are you ready computer scientist? The sessions are split into two categories: adaptability and addressing algorithm biases.

To recap:

On the Yanshee operating system, you should see the **Jupyter Lab** Desktop icon (Figure 7).

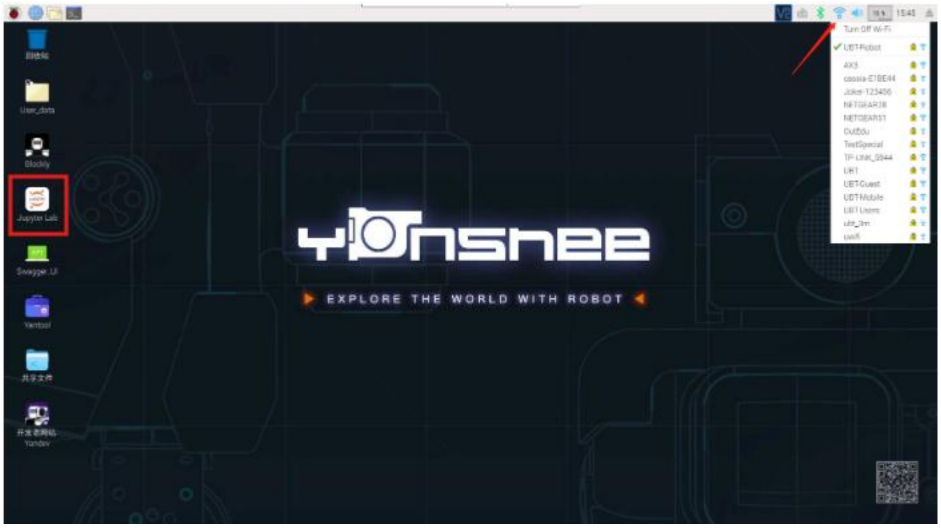
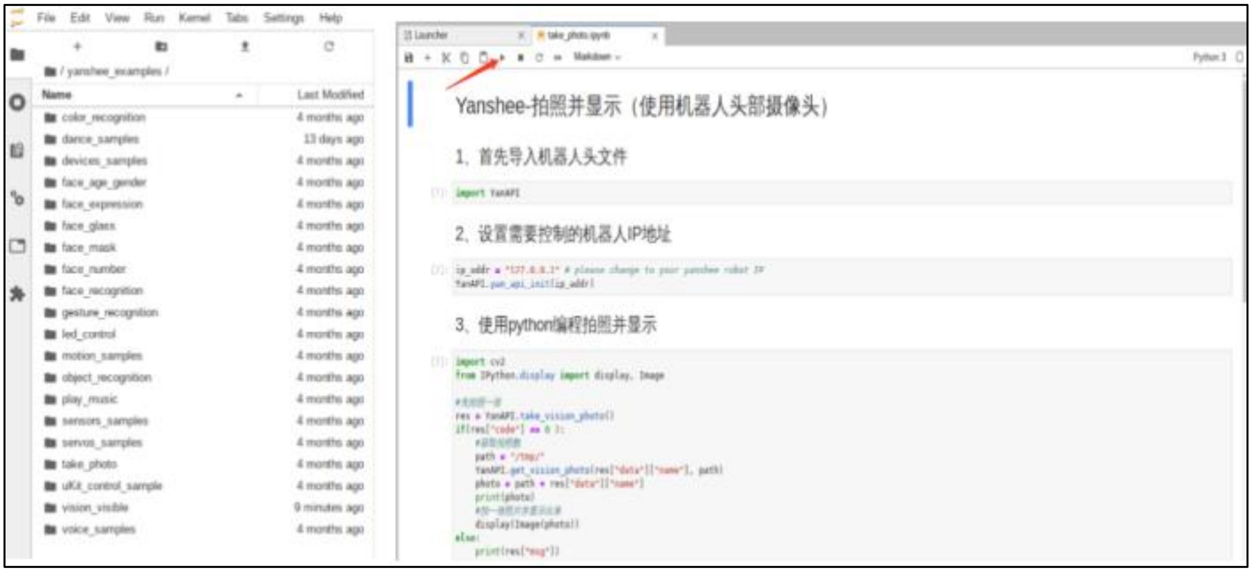


Figure 8: Yanshee operating system

Double-click the icon to open the **Jupyter Lab** interface and once opened, your screen should see look like what is being shown in Figure 8**.**



/ yanshee_examples /

Name	Last Modified
color_recognition	4 months ago
dance_samples	13 days ago
devices_samples	4 months ago
face_age_gender	4 months ago
face_expression	4 months ago
face_glass	4 months ago
face_mask	4 months ago
face_number	4 months ago
face_recognition	4 months ago
gesture_recognition	4 months ago
led_control	4 months ago
motion_samples	4 months ago
object_recognition	4 months ago
play_music	4 months ago
sensors_samples	4 months ago
servos_samples	4 months ago
take_photo	4 months ago
uKit_control_sample	4 months ago
vision_visible	9 minutes ago
voice_samples	4 months ago

Yanshee-拍照并显示（使用机器人头部摄像头）

1、首先导入机器人头文件

```
import YanAPI
```

2、设置需要控制的机器人IP地址

```
ip_addr = "127.0.0.1" # please change to your yanshee robot IP
YanAPI.yan_api_init(ip_addr)
```

3、使用python编程拍照并显示

```
import cv2
from IPython.display import display, Image

#拍照一张
res = YanAPI.take_vision_photo()
if(res["code"] == 0 ):
    #获取照片
    path = "/tmp/"
    YanAPI.get_vision_photo(res["data"]["name"], path)
    photo = path + res["data"]["name"]
    print(photo)
    #[illegible]
    display(Image(photo))
else:
    print(res["msg"])
```

Figure 8: Jupyter Lab Interface

Create a new file called `voice_interaction.ipynb` and save it in your folder. This is where we will write our code in Python for today's session. Feel free to ask the staff for assistance if you get stuck.

Adaptability

Let’s Revisit the Advanced Exercises from Level 5 Session 5: Buddy

In this section, we’ll build on our knowledge of programming Buddy by incorporating voice interaction capabilities. Buddy’s adaptability lies in his ability to recognise objects and respond to voice commands, bringing together advanced concepts in AI and robotics. This challenge will test your programming skills as you teach Buddy to combine visual and auditory inputs to perform intelligent actions.

Objective

Using the start_voice_asr method or the listen_res = YanAPI.sync_do_voice_asr() function from the YanAPI library, you will program Buddy to recognise spoken commands, visually identify objects, and provide a meaningful response.

What You’ll Do

1. **Enable Voice Interaction**

   Program Buddy to listen for the phrase **"object recognition"** and respond with:

   "Object detecting, please wait."

   Additionally, Buddy’s eye indicator should turn red to indicate that the object detection process has started.
2. **Object Detection**

   Provide a photo of an object for Buddy to analyse. Once Buddy identifies the object, he will announce the result. For example:

   "I see a banana."
   "This object is a rose."

Programming Steps

1. **Set Up Voice Recognition**

   Use the start_voice_asr or listen_res = YanAPI.sync_do_voice_asr() method to enable Buddy’s voice recognition capabilities.

   ```
   # Import YanAPI for Buddy's programming
   import YanAPI

   # Initialize API connection
   ```

```
api = YanAPI("127.0.0.1")  # Buddy's server address

# Start voice recognition
command = api.sync_do_voice_asr()

# Check if the command matches "object recognition"
if command == "object recognition":
    print("Object detecting, please wait.")
        api.set_eye_indicator("green")  # Set Buddy's eye indicator to green
```

2. **Add Object Detection Functionality**

Program Buddy to process visual data and announce the object recognition result.

```
# Perform object recognition
result = api.do_object_recognition(sync=True)
object_name = result["data"]["recognition"]["name"]

# Respond based on recognition results
if object_name != "none":
    print(f"I see a {object_name}.")
else:
        print("No object recognised.")
```

3. **Combine Voice and Object Recognition**

Integrate both functionalities to create a seamless interaction experience.

```
#integration
command = api.sync_do_voice_asr()
if command == "object recognition":
    print("Object detecting, please wait.")
    api.set_eye_indicator("green")
    result = api.do_object_recognition(sync=True)
    object_name = result["data"]["recognition"]["name"]
    if object_name != "none":
        print(f"I see a {object_name}.")
    else:
        print("No object recognised.")
```

Tips

- **Lighting and Background:** Ensure a plain, simple background and avoid overexposure to natural light when testing Buddy's object recognition capabilities.
- **Objects for Testing:** Test Buddy's recognition skills using a variety of objects from the following categories:
  - **Common Flowers:** Sunflowers, carnations, roses, daisies, etc.
  - **Green Plants:** Bamboo, Chinese orchids, tiger lilies, etc.
  - **Fruits:** Apples, oranges, bananas, grapes, peaches, etc.
  - **Specific Objects:** AlphaMini robot, Yanshee robot, Rubik's cube, QR codes, etc.

Today's Exercise:

Put your code to the test! Once you've completed your programming, ask a staff member to watch your code in action. Show how Buddy responds to voice commands, recognises objects, and provides feedback.

1. Begin with getting into group of 3.
2. Select the 3 objects from the pile to test Buddy's newly learnt skill.
3. Take turn raising the images to Buddy's eyes and make note how many out of the 3 objects raised, Buddy was able to recognise.
4. Show your findings in terms of probability to the student ambassadors.

Advanced Exercises:

Can you improve Buddy's ability to respond dynamically to a wider range of voice commands? Experiment with expanding his vocabulary and responses!

Remember to refer to the YanAPI documentation
https://yandev.ubtrobot.com/#/en/api?api=YanAPI

Once you think you are ready, ask a staff member to see how adaptable Buddy can be under your guidance!

Understanding Algorithmic Biases

Algorithmic bias is a critical topic in AI education. Students explore how biases arise from imbalanced training datasets and learn strategies for mitigating these biases. For instance, Buddy's ability to recognise objects depends on the diversity of the dataset used in its training. If the dataset lacks representation of certain objects or scenarios, Buddy may misclassify or fail to recognise them altogether.

Current state of Buddy

- Introducing Buddy
- Face detection
- Object detection

Today’s Exercise:

Look at how the limits of the dataset affect Buddy's ability to recognise things accurately. Show how Buddy responds to different varieties of objects from the ‘Tips section’ and determine which objects lacks balanced training datasets.

1. Begin with getting into group of 3.
2. Select the 3 objects from the pile to test Buddy’s newly learnt skill.
3. Take turn raising the images to Buddy’s eyes and make note how many out of the 3 objects raised, Buddy was able to recognise.
4. Show your findings in terms of probability to the student ambassadors.

Advanced Exercises:

Can you improve Buddy’s ability to respond dynamically to a wider range of test data? Experiment with expanding his vocabulary and responses!

Download and have fun with it

∉ Quick, Draw: https://quickdraw.withgoogle.com/

Once you think you are ready, ask a staff member to see how adaptable Buddy can become under your guidance!

Summary

This session introduced students to Buddy’s capabilities, emphasising key concepts such as APIs, adaptability, algorithmic biases, and programming Buddy for speech interaction, demonstrating how Buddy is artificially intelligent. Through hands-on exercises and discussions, students gained a deeper understanding of AI principles and their applications in robotics.

Useful Resources

∉ What is an API: https://www.geeksforgeeks.org/what-is-an-api/

∉ Yanshee API documentation: https://yandev.ubtrobot.com/#/en/api?api=YanAPI

∉ Quick, Draw: https://quickdraw.withgoogle.com/

∉ Yanshee mobile app on App Store: https://apps.apple.com/us/app/yanshee/id1290088340

∉ Yanshee mobile app on Google Play:
https://play.google.com/store/apps/details?id=com.ubtedu.alpha1x&hl=en_US