AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Sucessos Criativos, Lda

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

1. Project Title: 4

2. Project Category: 4

3. Project Description: 4

4. Keywords: 4

5. Target Audience: 4

6. Project Objectives: 5

7. Project Timeline: 5

8. Resources Provided: 5

9. Tools and Technologies Used: 6

10. Step-by-Step Implementation Plan: 7

Step 1: Prepare and Connect the Camera Module 7

Step 2: CONNECT THE CAMERA 7

Step 3: INSTALL THE CAMERA 10

Step 4: Using the code editor - OpenCV 11

11. Evaluation Criteria: 14

12. Languages Available: 15

13. Contributors: 15

14. Feedback and Continuous Improvement: 15

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Project Title:

First Steps with AI-VET Kit: Connecting The Camera

---

2. Project Category:

Microelectronics Fundamentals

---

3. Project Description:

In this project, you will connect a **Raspberry Pi Camera Module V2** to a **Raspberry Pi 5** and use it to capture high-resolution photos and stream live video. You'll learn how to handle the hardware with care, install required software tools, and write Python code to activate and preview the camera feed using **OpenCV** and **Picamera2**. This forms a foundation for future projects in object detection, facial recognition, or remote monitoring.

---

4. Keywords:

Raspberry Pi, Camera Module, Picamera2, OpenCV, Image Capture, Video Streaming, Computer Vision

---

5. Target Audience:

Educators, learners, and makers with or without prior experience who are beginning their journey into microelectronics and physical computing.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6. Project Objectives:

- Safely connect and configure a Raspberry Pi Camera Module V2.
- Capture still images using command-line tools.
- Stream live video in real time using Python with OpenCV and Picamera2.
- Learn best practices for camera alignment and manual focus adjustments.

7. Project Timeline:

The estimated duration is approximately **2.5 to 3 hours**, including environment setup and physical assembly.

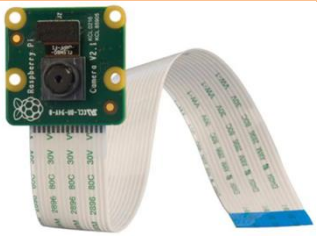
8. Resources Provided:

- AI-VET Kit (Raspberry Pi 5, Raspberry Pi Camera)

MATERIAL	
Raspberry Pi 5	

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

27W USB Type-C Power Supply for Raspberry Pi 5	
Raspberry Pi Camera Board V2 8MP 1080p (IM8)	
Official FPC camera cable for Raspberry Pi 5 200mm	

---

9. Tools and Technologies Used:

- Raspberry Pi OS
- Terminal (bash)
- Geany IDE
- Picamera2 Library
- OpenCV (cv2)
- libcamera

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

10. Step-by-Step Implementation Plan:

Step 1: Prepare and Connect the Camera Module

This project builds directly on previous AI-VET kit experiences, including:

- **"Getting Started with Raspberry Pi 5: Setup, Configuration, and Hello World in Python"**

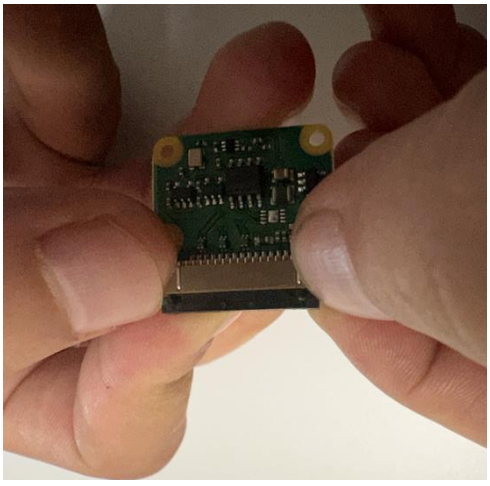
If you've completed this, you've already enabled critical interfaces and got the set up ready. Now we'll extend that by integrating the Raspberry Pi Camera.

Step 2: CONNECT THE CAMERA

If you have the case installed, you need to remove it. Remove the fan and the rest of the case. Start by unscrewing these parts and removing them carefully. Next, unplug the fan from the board. This step should be done very, very carefully!

Like any hardware add-on, this camera module should only be connected or disconnected when the Raspberry Pi is powered off. So, shutdown the Raspberry Pi and disconnect it from the power supply.

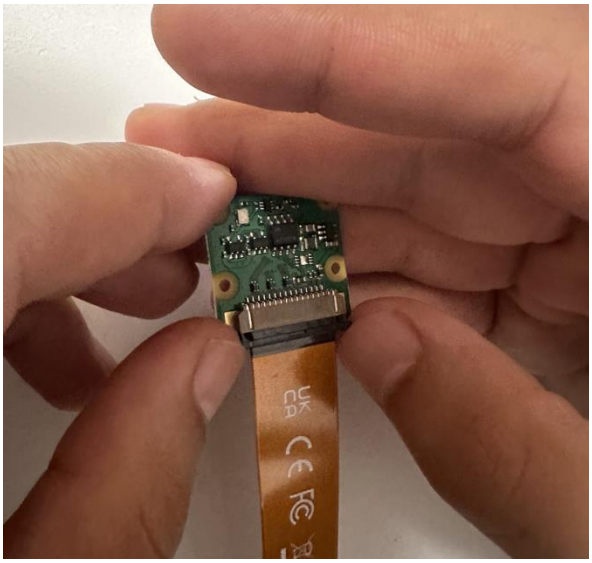
Now, connect the FPC camera cable to the Camera Module. First you will need to hook your fingernails around the edges of the connector and pull outwards until the flap lifts.



Co-funded by the European Union

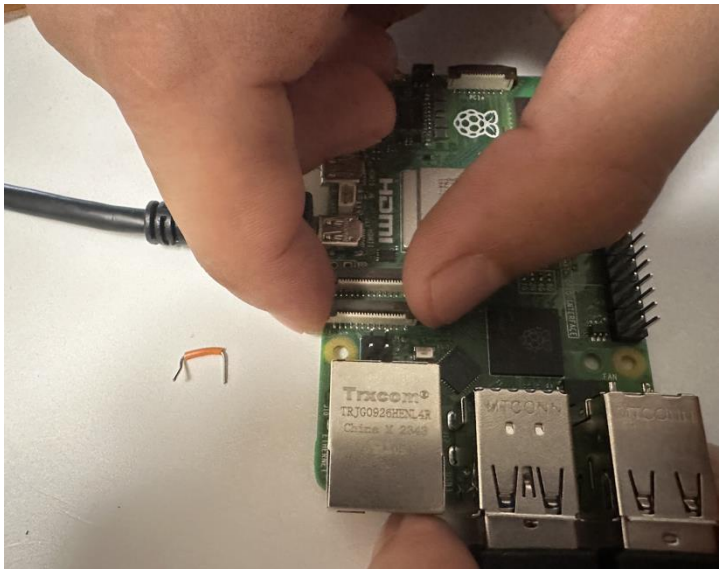
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Insert the FPC camera cable under the flap you just lifted, with the plastic side facing upwards.



Once the cable is inserted, gently push the flap back into place with a click. If it's not securely connected, pull it out and try again.

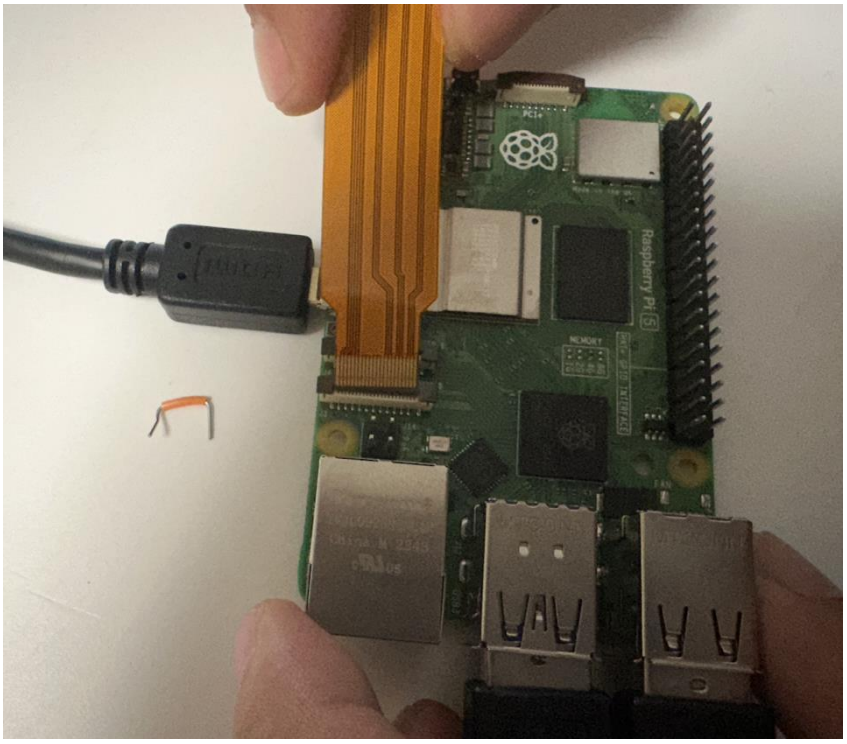
On the Raspberry Pi 5 you will see camera/display ports, marked "CAM/DISP 0". With the Raspberry Pi positioned so that the USB ports are to the left, carefully pull this port connector edges outwards:



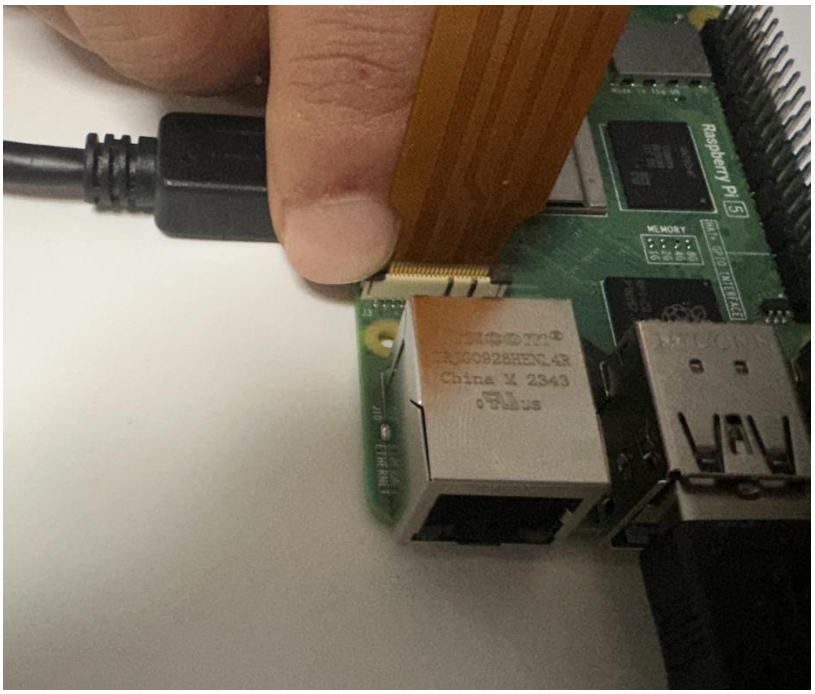
Insert the FPC camera cable with the gold edges facing to the left:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Then gently push the connector back into place. If the cable is installed properly, it will sit flat and won't come out when given a gentle tug:



Finally, reconnect the Raspberry Pi to the power supply and let it boot into Raspberry Pi OS.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 3: INSTALL THE CAMERA

To get started and check if the camera is working, you need to install the libcamera-apps package on bash:

```
sudo apt install libcamera-apps
```

```
(env) AI-VET@AI-VET01:~ $ sudo apt install libcamera-apps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra
  rpi.gpio-common
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  libcamera-apps
```

And now you can run the command

```
libcamera-still -o image.jpg
```

```
(env) AI-VET@AI-VET01:~ $ libcamera-still -o image.jpg
[0:05:50.812574279] [2730]  INFO Camera camera_manager.cpp:316 libcamera v0.3.1+
50-69a894c4
[0:05:50.820065820] [2733]  INFO RPI pisp.cpp:695 libpisp version v1.0.7 28196ed
6edcf 29-08-2024 (16:33:32)
[0:05:50.838521453] [2733]  INFO RPI pisp.cpp:1154 Registered camera /base/axi/p
cie@120000/rp1/i2c@88000/imx219@10 to CFE device /dev/media2 and ISP device /dev
/media0 using PiSP variant BCM2712_C0
Made X/EGL preview window
Mode selection for 1640:1232:12:P
    SRGGB10_CSI2P,640x480/0 - Score: 4504.81
    SRGGB10_CSI2P,1640x1232/0 - Score: 1000
    SRGGB10_CSI2P,1920x1080/0 - Score: 1541.48
    SRGGB10_CSI2P,3280x2464/0 - Score: 1718
    SRGGB8,640x480/0 - Score: 5504.81
    SRGGB8,1640x1232/0 - Score: 2000
    SRGGB8,1920x1080/0 - Score: 2541.48
    SRGGB8,3280x2464/0 - Score: 2718
```

This captures an image and saves it as image.jpg in the current directory.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 4: Using the code editor - OpenCV

In this step we will use cv2 (OpenCV) and Picamera2 which are libraries used for image and video processing.

cv2 is the Python interface for OpenCV (Open Source Computer Vision Library), an open-source library focused on real-time computer vision. It's used for tasks involving image and video analysis, manipulation and recognition. In this example, you will use cv2 to capture video from the camera and process frames in real-time

Picamera2 is a Python library designed to interface with the Raspberry Pi Camera Module.

After making the circuit you are almost ready to start coding!

With the terminal open in the environment we are working, you should now install cv2 by typing:

```
pip install opencv-python
```

Now, let's go to the code editor Geany! You should write the python code for the activation of the camera. Here's a detailed commented example of the code:

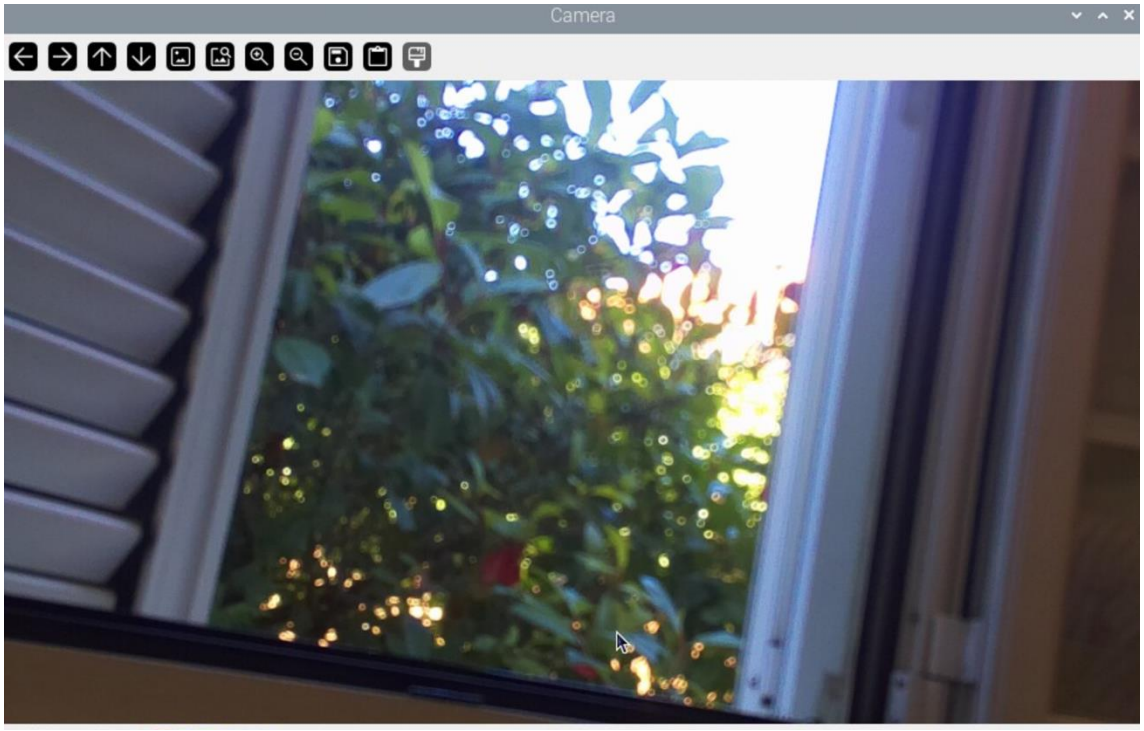
```
import cv2
from picamera2 import Picamera2
# Initialize the Picamera2 instance
picam2 = Picamera2()
# Set the preview configuration with a resolution of 1280x720 pixels
picam2.preview_configuration.main.size = (1280, 720)
# Set the image format to RGB888 (standard 24-bit RGB format)
picam2.preview_configuration.main.format = "RGB888"
# Align the configuration (if needed) for proper camera settings
picam2.preview_configuration.align()
# Configure the camera for preview mode
picam2.configure("preview")
# Start the camera capture
picam2.start()
# Start an infinite loop to continuously capture and display frames
while True:
    # Capture the current frame as a numpy array
    frame = picam2.capture_array()
    # Display the captured frame in a window titled "Camera"
    cv2.imshow("Camera", frame)
    # Break the loop and stop the capture if the 'q' key is pressed
    if cv2.waitKey(1) == ord("q"):
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
        break
# Release resources and close the OpenCV display window
cv2.destroyAllWindows()
```

Below is an example of the window that should appears after you run this code:



Although the Pi Camera v2 has a fixed-focus lens, you can manually adjust the focus by physically rotating the lens. Here's how:

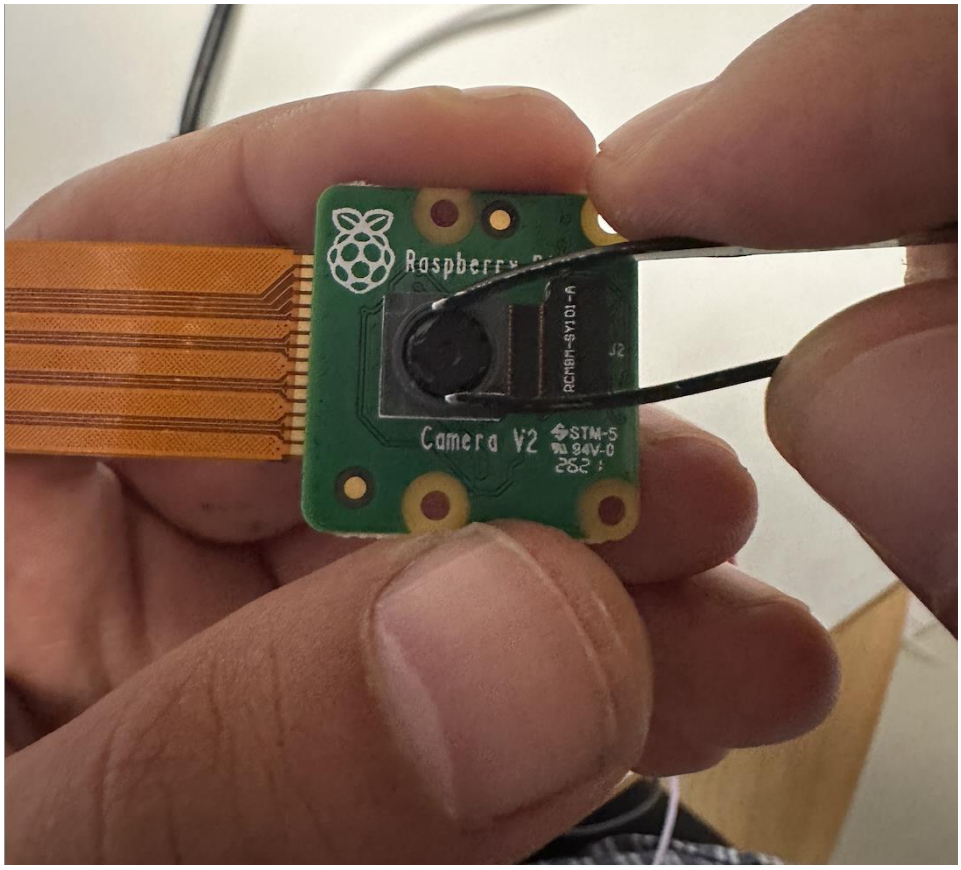
Step 5: Steps to Manually Adjust the Lens:

Remove the lens cover: If your camera has a protective cover over the lens, gently remove it.
Use a small tool (like tweezers in this caseor a small screwdriver) to carefully rotate the lens clockwise or counterclockwise. Be gentle, as applying too much force can damage the camera module.

Clockwise will focus the lens for closer objects.
Counterclockwise will focus the lens for objects farther away.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



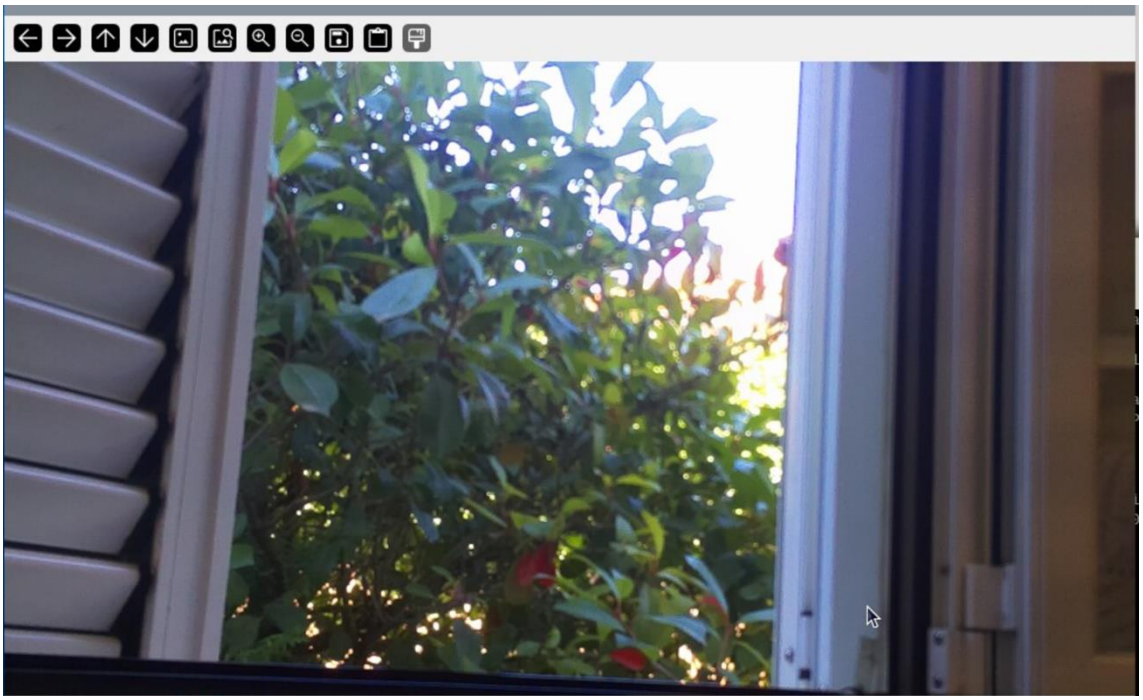
Test the focus: Use the preview window to check how the image looks as you adjust the focus.

Caution: Be very careful when adjusting the lens as it is delicate. Too much force can break the internal components.

Now, you can see the difference!

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Save the program as Camera.py

---

11. Evaluation Criteria:

Camera Hardware Setup	The camera is securely installed and recognized by the system
libcamera Integration	Captures still images without error
OpenCV Video Feed	Camera preview runs smoothly and displays frames in real time
Software Installation	Libraries (libcamera, OpenCV, Picamera2) are installed and functioning

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Focus Control (Optional)	User is able to manually adjust and test focus based on image clarity

---

12. Languages Available:

English, **Portuguese** (translations in progress)

---

13. Contributors:

Fátima Pais, Luís Soares – Project Designers and Facilitators

---

14. Feedback and Continuous Improvement:

- **Surveys**: At the end of this milestone, students and educators will complete feedback surveys to assess the effectiveness of the project.
- **Project Reviews**: After project completion, a final review meeting will be held to discuss potential improvements for future AI-based projects.
- **Data Analysis**: Feedback data will be analyzed to identify strengths and areas for improvement, informing future projects.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.