Project brief to evaluate the use of an AI camera.

Paul Mailes

Amber Technical Solutions Limited

Solent university student project – Jan 2025

Project sponsors. Amber Technical Solutions & Telemetry Engineering Solutions.

Version 1.4 – 05-02-2025

Robert Potter

Telemetry Engineering Solutions Limited

07833 677761

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Change Details** | **Version** |
| **17-01-2025** | BP | Introduction | 1.0 |
| **20/01/2025** | BP | Added a link to setting up the camera  Added B.O.M. – added table of contents  Added Hardware and software passages  Added a ChatGPT created appendix on making your own model | 1.1 |
| **21/01/2025** | BP | Added more references for building your own model | 1.2 |
| **30/01/2025** | BP | Increased the reference list to include web pages dedicated to AI reading of analogue gauges. | 1.3 |
| **05/02/2025** | BP | Added some hardware photos. | 1.4 |

Contents

[Contents 2](#_Toc189636315)

[Table of Figures. 3](#_Toc189636316)

[Bill of Materials 5](#_Toc189636317)

[General. 6](#_Toc189636318)

[Hardware. 6](#_Toc189636319)

[Software. 7](#_Toc189636320)

[What next? 8](#_Toc189636321)

[Appendix 1 – Creating your own models 9](#_Toc189636322)

[References 13](#_Toc189636323)

# Table of Figures.

[Figure 1 - Raspberry PI with AI – camera – plus case fan – Standard PI camera mount 6](#_Toc189636090)

[Figure 2- raspberry PI Camera module 3 and AI Hat - PI 5 cooling fat / heatsink – Standard PI camera mount 6](#_Toc189636091)

Introduction.

The Raspberry PI foundation recently introduced to market an ‘AI camera’ which is compatible with their Raspberry pi range of embedded Linux computers. The difference between their existing range of cameras and the AI version is that the Raspberry Pi AI Camera integrates a Sony IMX500 image sensor with on-chip AI capabilities, allowing for efficient on-device processing of neural network models. This design offloads AI processing from the Raspberry Pi's CPU, enabling more efficient performance.

OpenCV and machine learning opens up a world of possibilities where technology meets creativity. The student would potentially develop an intelligent system that can interpret and understand visual information. With packages such as OpenCV, a powerful open-source computer vision library, they can explore projects ranging from real-time object detection to facial recognition. By integrating machine learning algorithms, these systems can learn and improve over time, leading to applications in autonomous vehicles, healthcare diagnostics, and more. Engaging in this field not only hones their programming skills but also allows them to contribute to cutting-edge innovations that are shaping the future. Whether they're interested in building a smart attendance system that recognizes faces or tracks gestures, the fusion of OpenCV and machine learning offers a dynamic and exciting avenue for exploration and discovery.

Applications for consideration

I would suggest the students initially explore the feasibility of using this camera with a Raspberry PI 5 to:

1. Validate the performance of the camera for simple object recognition. This will include some academic content in the field of machine learning and computer vision.
2. Integrate the camera module with the CM5 to produce either:
   1. A back-end’ detection device to create a smart alarm system (such as a Gert board)
   2. A simple people counter system for say a shop’s ‘footfall’ metrics
   3. A door access system using facial recognition
   4. A home automation system that can detect where people are and adapt the background lighting and heating accordingly.
   5. A watchdog system for water discharge into rivers
3. Use a dashboard system to display the information to the user – Node-red / Flow fuse and also to send the data via MQTT to a central server for later retrieval and analysis

ATS understand that the University has the necessary hardware to carry out so research in this area.

# Bill of Materials

All equipment has been sourced from the Pi-Hut

|  |  |  |
| --- | --- | --- |
| CR2032 3V Lithium Coin Cell Battery | 1 | 1.9 |
| Raspberry Pi Compute Module 5 Development Kit - UK | 1 | 124.4 |
| Raspberry Pi Compute Module 4/5 Antenna Kit | 1 | 9.5 |
| Raspberry Pi AI Camera | 1 | 62.8 |
| KKSB Camera Holder | 1 | 8.2 |
| TBC |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# General.

## Hardware.

The raspberry PI AI camera is connected to the Compute module via a delicate ribbon cable. This needs to be protected so I suggest that we put all this equipment into a case. We will cut a viewing hole in the case so that the camera can be used. All ingress and egress connecting cables will be via special glanding. This will be prepared by the sponsor.

When setting up the Raspberry 5 we will need some form of casing.

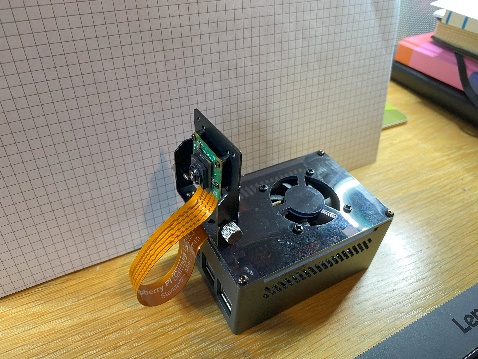


Figure - Raspberry PI with AI – camera – plus case fan – Standard PI camera mount

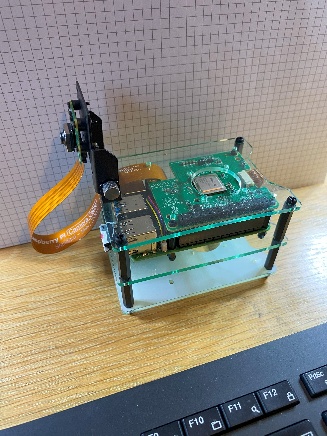


Figure - Raspberry PI Camera module 3 and AI Hat - PI 5 cooling fat / heatsink – Standard PI camera mount

Figure 1 - Raspberry PI with AI – camera – plus case fan – Standard PI camera mount shows a PI5 in a case purchased from PI HUT. This case comes with a fan built into the case. Stick on copper heat sinks have been fitted to the hot spots on the PI 5. This unit doesn’t have a PI 5 AI HAT but does have an AI camera fitted. A cursory investigation suggests that this camera may not work with YOLO but does work with a model that Raspberry PI recommend.

Figure 2- raspberry PI Camera module 3 and AI Hat - PI 5 cooling fat / heatsink – Standard PI camera mount – shows a Raspberry PI mounted in an open frame case available from PI HUT. This unit has a close coupled fan/heatsink fitted and the AI HAT. A standard module 3 camera is mounted on the camera mount again all components are available from the PI HUT.

I do not recommend running the raspberry pis without any form of cooling as they run very hot and quickly throttle back their performance if the temperature goes above certain built in protection setpoints (Which will happen when doing any model processing even with AI cameras and AI HATS.)

## Software.

The software required to get started is given here in the raspberry pi document: 3

There are many pre-built ‘out the box’ models covering a range of objects available that can be used on the hardware provided.

Many new open-source models are being created almost weekly but as a learning exercise it’s better to decide on one and stick with it and leave newer models (that may appear during the project duration) for a mention in the recommendations section at the end of the project write up.

A ‘stretch’ activity would be if the students were to create their own model using their own photo sets. An application that we have been considering, albeit unglamourous, is to train a camera on a sewer ‘flap valve’ or open channel so that changes can be remotely monitored.

Water supply companies are responsible for providing clean water and safely disposing of waste water. In an ideal world waste water and sewage would be cleaned in a treatment plant and safely discharged into the river and/or directly to the sea. In recent years all Water companies in England have been made to monitor the **times and durations** that they discharge any untreated sewage into the river network. This does not always monitor the volume of discharge.

To monitor this data using traditional methods is expensive (Google EDM/water companies for more details) so using a simple camera *may be* a cheaper option.

For this project ATS would like to establish whether using AI based camera detection methods would be worth further investigation.

This project for the students is constrained in time and resources so all ATS are expecting is a thumbs up/thumbs down recommendation. It may not be possible to get photographs of real plant used by the water companies but with a little ingenuity, I’m sure a multi-position ‘Lego’ model could be built for the purpose of demonstrating the ability of the system to detect the position and display the position on a suitable dashboard.

## What next?

It’s hardly a literature review but I have recorded the trawl I made of the web for information in the references section below, so if the students were to look at one or two of these they may get a feel for the state of play in this area. I am particularly impressed by YOLO and have run a model (version 11) which has a Class base of some 80 everyday objects. If nothing else it’s impressive and didn’t need any accelerators fitting.

# Appendix 1 – Creating your own models

**Source ChatGPT. – Beware – only to give you a flavour not to be religiously followed – look at the YOLO11 references below too!!!!**

Creating a custom machine learning (ML) model using the Raspberry Pi Camera Module involves several key steps: setting up your Raspberry Pi and camera, collecting and annotating data, training your ML model, and deploying it on the Raspberry Pi. Here's a rough guide to help you through the process:

**1. Set Up Your Raspberry Pi and Camera Module**

* **Hardware Connection**: Connect the Raspberry Pi Camera Module to the CSI port on your Raspberry Pi. Ensure the metallic contacts of the ribbon cable face the correct direction as per your Raspberry Pi model.
* **Enable the Camera Interface**:
  + Boot up your Raspberry Pi.
  + Open a terminal and run: ```bash sudo raspi-config
  + Navigate to 'Interfacing Options' > 'Camera' and select 'Enable'.
  + Reboot your Raspberry Pi to apply the changes.

**2. Install Necessary Software**

* **Update the System**: ```bash sudo apt update && sudo apt full-upgrade -y
* **Install Python and pip**: ```bash sudo apt install python3 python3-pip -y
* **Install OpenCV**: ```bash pip3 install opencv-python
* **Install TensorFlow Lite**: TensorFlow Lite is optimized for edge devices like the Raspberry Pi. You can install it using pip: ```bash pip3 install tflite-runtime
* \*Note: Pre-built TensorFlow Lite packages are available for Raspberry Pi. Ensure compatibility with your Raspberry Pi model.\*

**3. Collect and Annotate Data**

* **Capture Images**: Use the Raspberry Pi Camera Module to capture images of the objects you want your model to recognize. Ensure you capture images from various angles and lighting conditions to create a robust dataset.
* **Annotate Images**: Use annotation tools like LabelImg to label your images. This involves drawing bounding boxes around the objects of interest and assigning them appropriate labels. ```bash pip3 install labelImg
* \*Note: LabelImg provides a graphical interface to facilitate the annotation process.\*

**4. Train Your Machine Learning Model**

* **Set Up a Training Environment**: It's recommended to use a more powerful machine, such as a PC or cloud service, for training due to the computational demands.
* **Prepare the Dataset**: Organize your annotated images and corresponding annotation files into training and validation sets.
* **Choose a Model Architecture**: Models like MobileNetV2 or SSD are suitable for edge devices due to their efficiency.
* **Train the Model**: Use frameworks like TensorFlow to train your model on the prepared dataset.
* **Convert the Model to TensorFlow Lite Format**: After training, convert your model to TensorFlow Lite format for deployment on the Raspberry Pi. ```python import tensorflow as tf

converter = tf.lite.TFLiteConverter.from\_saved\_model('path\_to\_saved\_model') tflite\_model = converter.convert()

with open('model.tflite', 'wb') as f: f.write(tflite\_model)

**5. Deploy and Run the Model on Raspberry Pi**

* **Transfer the Model**: Copy the model.tflite file to your Raspberry Pi.
* **Run Inference**: Use a Python script to load the TensorFlow Lite model and perform inference on images captured by the Raspberry Pi Camera Module. ```python import cv2 import numpy as np import tflite\_runtime.interpreter as tflite

**Load the TFLite model**

interpreter = tflite.Interpreter(model\_path='model.tflite') interpreter.allocate\_tensors()

**Get input and output tensors**

input\_details = interpreter.get\_input\_details() output\_details = interpreter.get\_output\_details()

**Initialize the camera**

cap = cv2.VideoCapture(0)

while cap.isOpened(): ret, frame = cap.read() if not ret: break

# Preprocess the frame

input\_data = cv2.resize(frame, (input\_details[0]['shape'][2], input\_details[0]['shape'][1]))

input\_data = np.expand\_dims(input\_data, axis=0)

input\_data = input\_data.astype(np.float32) / 255.0

# Set the tensor to point to the input data

interpreter.set\_tensor(input\_details[0]['index'], input\_data)

# Run inference

interpreter.invoke()

# Get the output

output\_data = interpreter.get\_tensor(output\_details[0]['index'])

# Process the output as needed

# ...

# Display the frame

cv2.imshow('Frame', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release() cv2.destroyAllWindows()

\*Note: This script captures video frames, preprocesses them, runs inference using the TensorFlow Lite model, and displays the frames. Modify the processing of `output\_data` as per your model's requirements.\*

**6. Optimize Performance**

* **Use Coral Edge TPU**: For faster inference, consider using the Coral Edge TPU USB Accelerator, which enhances performance on the Raspberry Pi.
* **Optimize the Model**: Quantize your model during conversion to reduce its size and improve inference speed. ```python converter.optimizations = [tf.lite.Optimize.DEFAULT]

By following these steps, you can create a custom ML model using the Raspberry Pi Camera Module, enabling your Raspberry Pi to perform tasks like object detection, image classification, or other computer vision applications.

# References

1. [Raspberry Pi AI Camera | The Pi Hut](https://thepihut.com/products/raspberry-pi-ai-camera?variant=43973693046979&country=GB&currency=GBP&utm_medium=product_sync&utm_source=bing&utm_content=sag_organic&utm_campaign=sag_organic&msclkid=82da61c372b115a61c13064919745f96&utm_term=4585169655019870)
2. [Compute Module 5 on sale now from $45 - Raspberry Pi](https://www.raspberrypi.com/news/compute-module-5-on-sale-now/#:~:text=Compute%20Module%205%20gives%20you%20everything%20you%20love,2%20%C3%97%204-lane%20MIPI%20camera%2Fdisplay%20transceivers%20More%20items)

1. <https://www.raspberrypi.com/documentation/accessories/ai-camera.html#getting-started>
2. [run yolo on raspberry pi ai camera - Raspberry Pi Forums](https://forums.raspberrypi.com/viewtopic.php?t=378056)
3. [Prepare and Deploy AI Models Tutorial | Sony Semiconductor Solutions Group](https://aitrios-portal.wpp.developer.sony.com/en/raspberrypi-ai-camera/develop/ai-tutorials/prepare-and-deploy-ai-models-tutorial?version=2024-11-21&progLang=)
4. [5 Easy AI Projects on Raspberry Pi for Beginners – RaspberryTips](https://raspberrytips.com/easy-raspberry-pi-ai-projects/#training-your-own-ai-model-tensorflow)
5. [Raspberry Pi AI Camera review: Even more approachable AI](https://magazinmehatronika.com/en/raspberry-pi-ai-camera-review-even-more-approachable-ai/)
6. [Raspberry Pi AI Camera Quick-Start Guide - Tutorial Australia](https://core-electronics.com.au/guides/raspberry-pi-ai-camera-quickstart-guide/)
7. [Object and Animal Recognition With Raspberry Pi and OpenCV - Tutorial Australia](https://core-electronics.com.au/guides/object-identify-raspberry-pi/)
8. [GitHub - SharpAI/DeepCamera: Open-Source AI Camera. Empower any camera/CCTV with state-of-the-art AI, including facial recognition, person recognition(RE-ID) car detection, fall detection and more](https://github.com/SharpAI/DeepCamera)
9. [Raspberry Pi OCR Edge-AI camera - Hackster.io](https://www.hackster.io/johannasss/raspberry-pi-ocr-edge-ai-camera-2fdaa5)
10. [GitHub - automaticdai/rpi-object-detection: Real-time object detection and tracking with Raspberry Pi and OpenCV!](https://github.com/automaticdai/rpi-object-detection)
11. [New AI Camera Halloween Project. Object detection with Sony IMX500 - Raspberry Pi Forums](https://forums.raspberrypi.com/viewtopic.php?t=378415)
12. [Raspberry Pi AI Camera for object detection: New module released](https://emteria.com/blog/raspberry-pi-ai-camera)
13. [Create a real-time object tracking camera with TensorFlow and Raspberry Pi | Opensource.com](https://opensource.com/article/20/1/object-tracking-camera-raspberry-pi)
14. [run yolo on raspberry pi ai camera - Raspberry Pi Forums](https://forums.raspberrypi.com/viewtopic.php?t=378056)
15. [Face & Eyes Detection with OpenCV Raspberry Pi Camera](https://how2electronics.com/face-eyes-detection-with-opencv-raspberry-pi-camera/)
16. [GitHub - Sahilofficial/Object-Detection-on-Raspberry-Pi-with-Obstacle-Avoiding-Rover: In this project, we performed a real-time-based object detection with a 5k image pre-learned dataset using Raspberry Pi and Pi/USB camera with an Obstacle avoiding Rover.](https://github.com/Sahilofficial/Object-Detection-on-Raspberry-Pi-with-Obstacle-Avoiding-Rover)
17. [Tutorial of AI Kit with Raspberry Pi 5 about YOLOv8n object detection | Seeed Studio Wiki](https://wiki.seeedstudio.com/tutorial_of_ai_kit_with_raspberrypi5_about_yolov8n_object_detection/)
18. [GitHub - kevinmcaleer/object\_detection: Build your own Object Detection Model using Raspberry Pi AI Camera and Pytorch](https://github.com/kevinmcaleer/object_detection)
19. [documentation/documentation/asciidoc/accessories/ai-camera/getting-started.adoc at develop · raspberrypi/documentation · GitHub](https://github.com/raspberrypi/documentation/blob/develop/documentation/asciidoc/accessories/ai-camera/getting-started.adoc)
20. [YOLOv8 - Ultralytics YOLO Docs](https://docs.ultralytics.com/models/yolov8/)
21. [YOLO11 🚀 NEW - Ultralytics YOLO Docs](https://docs.ultralytics.com/models/yolo11/#overview)
22. [Ultralytics Solutions: Harness YOLO11 to Solve Real-World Problems - Ultralytics YOLO Docs](https://docs.ultralytics.com/solutions/)
23. [Security Alarm System - Ultralytics YOLO Docs](https://docs.ultralytics.com/guides/security-alarm-system/)
24. [How to train Ultralytics YOLO11 Model on Medical Pills Detection Dataset in Google Colab | Pharma AI](https://www.youtube.com/watch?v=8gePl_Zcs5c)
25. [Raspberry Pi - Ultralytics YOLO Docs](https://docs.ultralytics.com/guides/raspberry-pi/#best-practices-when-using-raspberry-pi)
26. [Camera software - Raspberry Pi Documentation](https://www.raspberrypi.com/documentation/computers/camera_software.html#introducing-the-raspberry-pi-cameras)
27. [GitHub - KorniiVlasenko/Meters-Reader-YOLOv8-OpenCV: Reads meter readings from a photo](https://github.com/KorniiVlasenko/Meters-Reader-YOLOv8-OpenCV)
28. [Turn Analog Dials into Digital Data with Computer Vision](https://blog.roboflow.com/read-analog-dials-computer-vision/) (could be of interest when working in intrinsically safe/hazardous environments with old legacy equipment that cannot be digitised)
29. [Stop Wasting Time Transcribing Gauges: Automate Gauge Readings with AI in Days, Not Months | Synanthropic AI](https://synanthropic.ai/reading-analog-gauge)
30. [Machine Learning in Practice: Using Artificial Intelligence to Read Analog Gauges | Object Computing, Inc.](https://objectcomputing.com/resources/publications/sett/june-2019-using-machine-learning-to-read-analog-gauges) (Dated now)