# **LAB # 7: Edge Computing**

**Lab Objective:**

The objective of this lab is to get acquainted with Raspberry Pi 5 and set it up for running different Python scripts that have already been utilized in the previous labs.

**Lab Description:**

Raspberry Pi 5 single-board computer has a 2.4GHz quad-core 64-bit Arm Cortex-A76 processor and boasts enhanced graphics with dual 4Kp60 display output, improved I/O capabilities, including faster USB and microSD card performance, and introduces a PCI Express interface for high-speed peripherals. The Raspberry Pi 5 is available in various RAM configurations ranging from 2GB to 16GB, making it a powerful and versatile option for various applications, from desktop computing to industrial automation and DIY projects. The small size of the board allows it to become an integrated part of larger projects as well.

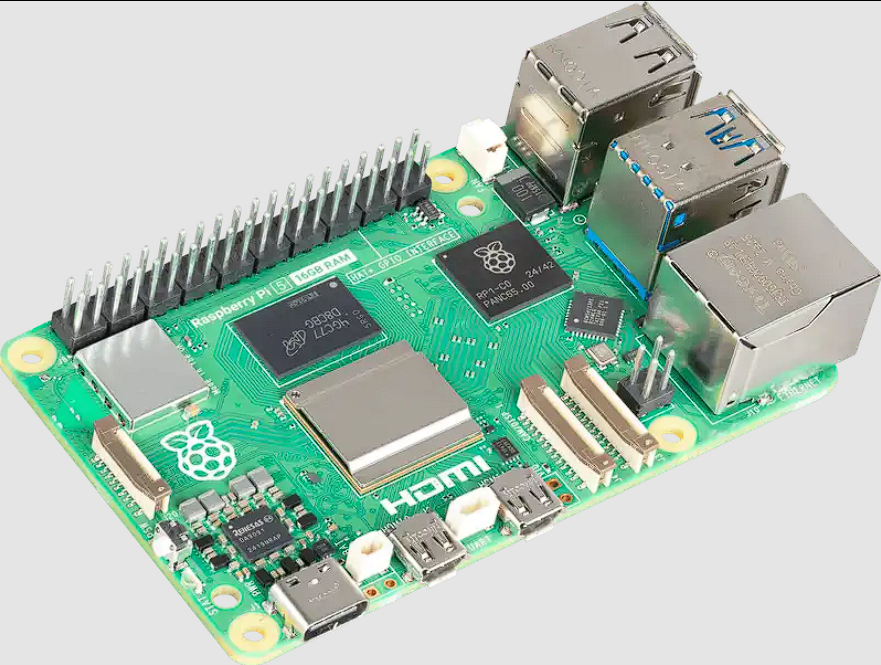


Fig: 01 Raspberry Pi 5

## Setting up the Raspberry Pi:

* Selecting an Operating System:

Much like the broader Linux ecosystem, the Raspberry Pi supports a variety of distributions, each tailored for specific applications or use cases. Among these, Raspberry Pi OS is the most popular, offering robust support for accessing the board’s various I/O pins.

For this lab, we will be using the Raspberry Pi OS as it provides a GUI and an easy installation process. The OS can be downloaded from the following link (please make sure that you select the correct version): <https://www.raspberrypi.com/software/operating-systems/>

There is also another very useful distro for Raspberry Pi called Diet Pi which just keeps the essentials elements to run the OS and throws everything else away. It also allows for a headless boot (without any GUI) over SSH which can be very useful if you do not have a monitor to spare. It can be downloaded from: <https://dietpi.com/>

* Burning/Etching the OS:

Once you have the OS image (.iso or similar file), you need to write it to a microSD card as the Raspberry Pi will use the microSD to run the OS (as it does not have any built-in storage). A useful tool to use for this is called Rufus which can be downloaded from: <https://rufus.ie/en/>

* Booting it up:

Once you have written the OS to the card, you should be able to boot up your Raspberry Pi. Connect all the peripherals (monitor, mouse, keyboard and power) and the Pi should boot up without any problem.

* Installing Open-CV:

Almost all Raspberry Pi distros are pre-installed with Python, so you do not have to install it again. However, you have to install OpenCV by running the pip install command that we used in the 1st lab.

Once all of the steps have been completed successfully, your Raspberry Pi is fully configured and ready to execute Python scripts.

## Lab Tasks:

Run the following tasks on your Raspberry Pi.

**Lab Task 1:**

Image enhancement can be performed using different types of filters. Run the code for convolution and blur (any) image using a 5x5 averaging filter. Also apply Sobel filters and display the results.

**Lab Task 2:**

Take any image and perform histogram equalization on the image. Display the original image and the equalized image side by side.

**Lab Task 3:**

Write a function that takes hematological image as input and segments the image using K-mean clustering with and k=2. Write the function such that the k value can be passed as a parameter. Display the resulting image.

Do It Yourself:

As mentioned earlier, Dietpi is another distro that has the advantage of headless setup. Your task is to follow the tutorial given here (<https://dietpi.com/docs/install/>) and setup Dietpi in a *headless* manner. Such installations are crucial for using Pi as part of a larger system.