

# **Beyond a Single-Board Computer: A Systematic Study of Raspberry Pi Architecture, Comparative Analysis of Raspberry Pi Models, and Insights with NVIDIA Jetson for Embedded Computing**

**Smit Rupera**

Department of Computer Science  
and Engineering  
Gyanmanjari Innovative University  
Sidsar road, Bhavnagar, India

**Parthrajsinh Mori**

Department of Computer Science  
and Engineering  
Gyanmanjari Innovative University  
Sidsar road, Bhavnagar, India

**Mahi Patel**

Department of Information  
Technology  
Gyanmanjari Innovative University  
Sidsar road, Bhavnagar, India

**Dev Sojitra**

Department of Computer Science and Engineering  
Gyanmanjari Innovative University  
Sidsar road, Bhavnagar, India

**Mansi Shanishwara**

(Formerly Published as Mansi Bosamia)  
Department of Computer Science and Engineering  
Gyanmanjari Innovative University  
Sidsar road, Bhavnagar, India

## **ABSTRACT**

The Raspberry Pi represents a low-cost, small sized single board computer designed to encourage computer science education and provide applications for various purposes. Over the years, the Raspberry Pi has become a flexible computing platform for education, embedded applications, the Internet of Things (IoT) community, robotics, home automation, and industrial automation. This study gives a complete description of the Raspberry Pi system. It includes the computer system architecture, the history of the Raspberry Pi system on chip, the available memory and storage solutions, connectivity modules, and support for various external devices. A comparison among various Raspberry Pi boards explains the performance variations, power requirements, and applications of the Raspberry Pi 4 Model B and the Raspberry Pi Zero 2 W. The use of Raspberry Pi operating system software and various computer science education tools, such as Python, Scratch, Greenfoot, and BlueJ, has been discussed to demonstrate the viability of Raspberry Pi for computer science education. The Raspberry Pi computer represents a scalable, efficient, and accessible platform. It fills the gap for computer science education and engineering applications across various fields of study.

## **Keywords**

Raspberry Pi, Single Board Computer, Embedded Systems, STEM Education, Internet of Things (IoT), Raspberry Pi OS, GPIO, Low-Cost Computing, Educational Technology, Industrial Automation.

## **1. INTRODUCTION**

The Raspberry Pi is a small, affordable single board computer that started out as a tool for teaching to computer science students. The Raspberry Pi Foundation wanted something that felt a bit like the old BBC Micro simple, cheap, and easy for beginners to pick up and start tinkering with. They wanted students to have something they could actually mess around with, not just read about in a textbook. Since the first Raspberry Pi Model B launched in 2012, people have found all sorts of

ways to use it. It's not just for classrooms anymore. The Raspberry Pi's flexibility, tiny size, and huge, active community mean you'll find it in hobby projects, industrial automation, IoT devices, and plenty of other places you wouldn't expect [1][2][3]. Over time, Raspberry Pi turned into a full-blown computing platform. People now use it for all sorts of things embedded systems, IoT gadgets, robots, smart homes, industrial controls, and edge computing. These days, a Raspberry Pi board packs everything you need onto one little circuit board: processor, memory, network connections, HDMI output, USB ports, and GPIO pins. You can write software on it, but you can also hook it straight up to hardware and start building. [4][5][6]. Raspberry Pi started out as a simple tool for teaching, but things have changed fast. With better processing power, improved connectivity, and more options for peripherals, it's now a flexible platform you'll find in research labs, industrial automation setups, and real world projects. That's why Raspberry Pi matters so much in modern computing today [7][8][9].

## **2. RASPBERRY PI DETAILS AND PARAMETERS**

Raspberry Pi is a line of small, affordable single board computers. They're great for learning, research, and building all sorts of embedded systems. Each board packs a processor, memory, networking, storage support, display output, and GPIO pins onto one compact circuit board. Because everything's built in, a Raspberry Pi runs like a full computer but stays energy efficient and tiny.

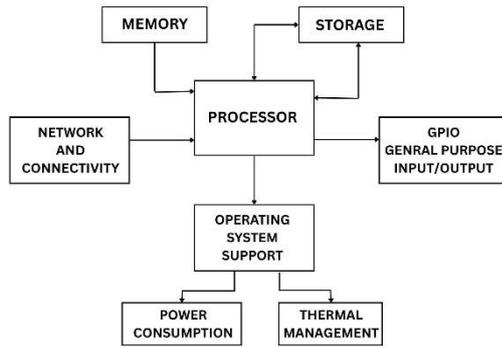


Fig 1: Raspberry Pi Parameters

## 2.1 Processor and System-on-Chip (SoC)

Raspberry Pi devices use System on Chip (SoC) designs produced largely by Broadcom. The older generation used single core ARM11 processors, while the newer use quad core ARM Cortex-A series processor cores and are 64-bit. The recently released Raspberry Pi 4 and Raspberry Pi 5 have the faster Cortex-A72 and Cortex-A76 processors respectively. These allow much higher computing power, multitasking, and support of heavy applications with the newer models [1][10][11].

## 2.2 Memory (RAM)

Memory in Raspberry Pi boards is system memory that is dynamically shared between the CPU and the GPU. The amount of memory has increased from the 256 MB and 512 MB of the original Raspberry Pi 1 Model A & B to support a maximum of 8 GB in Raspberry Pi 4 and up to 16 GB in Raspberry Pi 5. This scaling has made Raspberry Pi devices suitable for a wide range of workloads, from small, embedded tasks to desktop class applications [10][12].

## 2.3 Storage

The main storage device for a Raspberry Pi device is a microSD card, which contains the operating system and any additional data. Some Compute Module models include an eMMC flash memory chip soldered on board, offering more reliable storage in industrial or embedded use cases. The latest models can boot from USB storage devices, and the Raspberry Pi 5 adds support for PCIe based NVMe SSDs, allowing for higher performance storage [10][4][12].

## 2.4 Networking and Connectivity:

Networking capabilities vary depending on the Raspberry Pi model. The older models were equipped with wired Ethernet, while most of the modern models feature on board Wi-Fi and Bluetooth for wireless networking and Internet of Things applications. Additionally, the higher end models, namely Raspberry Pi 4 and 5, feature Gigabit Ethernet to allow high data transfer rates, making them feasible for applications like servers, multimedia processing, and edge computing [10][13].

## 2.5 Input/Output Interfaces and GPIO

The GPIO header is one of the major hardware identifying features of the Raspberry Pi. It usually includes 40 pins that are arranged in a standard pattern. These pins are considered the main digital input and output sources for the board, and communication between devices can be carried out through them in different ways, such as UART, SPI, IC, and IS. The general purpose input/output interface provides a direct link with the electronic world through sensors, actuators, motors, or any other electronic element, which in turn turns the Raspberry

Pi into a perfect tool for embedded systems, robotics, and hardware prototyping [4] [5] [14].

## 2.6 Power Consumption and Thermal Management

Raspberry Pi devices are designed to be energy efficient and thus consume significantly less power than traditional desktop computers. However, power consumption increases with greater processing load and higher model complexity. To deal with heating and keep the system stable, the latest boards have dynamic frequency scaling and thermal throttling. If you want to keep the Raspberry Pi running at full speed for a long time, it is better to use some passive or active cooling methods like heat sinks, fans, etc. [15][16][17].

## 2.7 Operating System Support

Raspberry Pi OS is a Debian, based Linux distribution tailored with low memory usage and hardware compatibility in mind. It serves as the best operating system for Raspberry Pi in three flavors of the OS are available: Lite, Desktop, and Full. Raspberry Pi is also capable of running a variety of other operating systems, including Ubuntu, Kali Linux, Libre ELEC, and Android, based distributions, thus making it a flexible tool for the sectors of education, research, and industry.

## 3. PROBLEM DEFINITION AND OBJECTIVE

### 3.1 Problem Definition

Problem Definition Affordable, flexible, and practical computing platforms have become a major hurdle in computer science education, the development of embedded systems, and low, cost digital innovation. In many cases, the typical desktop and laptop computers are expensive, consume a lot of power, and are not ideal for direct hardware interfacing. Therefore, their suitability for hands on learning and prototyping is limited. Students, especially those in resource, poor regions, do not have access to hardware platforms that can be programmed and that also offer a seamless integration of software development with the real world of electronics [1][3][9]. On the other hand, developers and researchers who are involved in the design of embedded systems, IoT, and automation need small, energy, efficient, and powerful computing solutions that can still run modern operating systems, provide networking capabilities, and allow for peripheral interfacing. Some of the existing embedded platforms are underpowered, proprietary, or have complicated development environments. As a consequence, there is a divide between classroom learning and practical implementation, as well as between low, level microcontroller systems and personal computers in general. Hence, a device that would cost little, be scalable, and work as a versatile platform to bridge the gap by allowing the programming of software, interaction with hardware, and development of applications for the real world in a single, user, friendly system is required [4][11][18].

### 3.2 Objectives

The first set of the several objectives of this experiment is to explore the architecture and essential components of the Raspberry Pi platform, which runs from a processor, memory, storage, networking, to GPIO interfaces, and the Raspberry Pi model's evolution by analyzing the performance, connectivity, and power efficiency improvements to understand how they meet different application requirements. This study also intends to understand the impact of Raspberry Pi on education, with a particular emphasis on the support it provides to programming environments such as Python, Scratch, Greenfoot, and BlueJ for STEM learning. Additionally, the positioning of Raspberry

Pi as an embedded system platform and a future in the Internet of Things (IoT) scenario is justified by features such as low power consumption, hardware interfacing capability, and operating system support. Moreover, the study explores the extent to which Raspberry Pi as a low, cost computing solution is effective for research, prototyping, home automation, and industrial automation use cases. Finally, the research demonstrates how the Raspberry Pi platform serves as an intermediate stage between traditional personal computers and microcontroller, based systems by providing a unified hardware and software ecosystem [3][5][6].

#### 4. TOOLS USED

Tools Used This research involved hardware and software tools that are part of the Raspberry Pi ecosystem. Their capabilities, performance, and applicability in education and embedded system development were analyzed.

##### 4.1 Hardware Tools

The main hardware tool is the Raspberry Pi single, board computer, which combines a processor, RAM, networking interfaces, storage support, display output, and GPIO pins in a small platform. Various Raspberry Pi models are referred to in the present work: the Raspberry Pi Zero series, the Raspberry Pi 3 series, and the Raspberry Pi 4, as the representative platforms for assessing changes in processing power, memory, connectivity, and power consumption. Peripheral hardware tools are microSD cards for storage, power adapters, HDMI display interfaces, USB input devices (keyboard and mouse), and external electronic components such as sensors and actuators connected through the GPIO header [10][12][13].

##### 4.2 Software Tools

Software Tools The main software tool is Raspberry Pi OS, a Debian, based Linux operating system designed for Raspberry Pi hardware. Raspberry Pi OS delivers a stable and low, resource computing environment that is suitable for both desktop and embedded applications. It includes support for essential system utilities, networking tools, and development environments that are needed for programming and hardware interaction [19][20][21]. Various development tools and environments are also utilized for educational and programming purposes. These consist of Python (IDLE) as a general purpose and hardware oriented programming language, Scratch as a visual and beginner and friendly programming language, and Java, based educational environments like Greenfoot and BlueJ which help in understanding object oriented programming concepts. Besides these, some other educational tools like Sonic Pi are employed to explain creative programming techniques through music and sound synthesis [7] [8] [22].

##### 4.3 Development and Interfacing Tools

Development and Interfacing Tools: Interfacing with hardware as well as developing an embedded application are supported by the Raspberry Pi's GPIO interface and the communication protocols associated with it, like UART, SPI, and IC, which are used for the interaction with the external devices. Configuration and system management are carried out by means of standard Linux command line utilities and system configuration files. To install and manage operating systems, Raspberry Pi Imager is utilized to flash Raspberry Pi OS and other supported operating systems to the storage media.

### 5. METHODOLOGY

This research assessed the Raspberry Pi through a structured analysis of its hardware architecture, software environment, and applications in education and embedded systems. The research began with an exhaustive investigation into the hardware ecosystem of the Raspberry Pi. The review covered the examination of the System on Chip (SoC), processor architecture, memory configurations, storage mechanisms, networking interfaces, and GPIO capabilities for different Raspberry Pi models. The study also traced back the evolution of the platform, figuring out how computational performance, connectivity, and power efficiency had improved over the different models like the Raspberry Pi Zero series, Raspberry Pi 3 series, and Raspberry pi 4 [1][4][12].

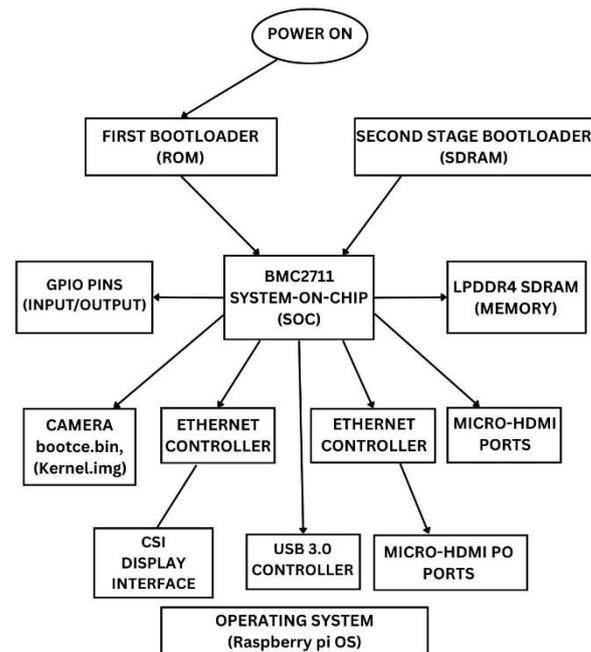


Fig 2: Raspberry Pi Boot Process & Components

The next step involves an in depth examination of the software environment of Raspberry Pi. To get the OS ready for evaluation, Raspberry Pi OS, which is a Debian, based Linux distribution designed for efficient use of low resources, is installed by means of Raspberry Pi Imager and then set up. The study of the operating system's compatibility with educational and development tools is carried out by the deployment of programming environments like Python (IDLE), Scratch, Greenfoot, and BlueJ. These tools undergo scrutiny in terms of how well they can serve as a medium for programming education, the learning of object-oriented principles, and the development of interactive applications [19], [20]. The Raspberry Pi's GPIO interface serves as the medium through which the evaluation of hardware interfacing and embedded system functionalities is carried out. To that end, the standard communication protocols such as UART, SPI, and IC are put to use for establishing connections with external peripherals, e.g., sensors and actuators. The described method paves the way for gauging the Raspberry Pi's potential of not only executing the tasks of real time data acquisition and device control but also of achieving seamless hardware and software integration in the local embedded and IoT oriented scenarios [5][14][23]. At the very end of the study, a comparative analysis is conducted to weigh the different Raspberry Pi models against each other in regard to factors like processing

capability, memory support, connectivity options, and power consumption. The main purpose of this analysis is to formulate the criteria for decision making about which of the models can be utilized in which application domains, e.g., education, prototyping, home automation, and industrial automation. Employing such a comprehensive systematic approach is what makes it possible to have an objective evaluation of Raspberry Pi as a platform that is not only low cost but also versatile.

## 6. RASPBERRY PI BASED TOOL WORKING

The Raspberry Pi is essentially a single, board computer platform that incorporates CPU, RAM, storage, networking, and I/O interfaces all in a tiny circuit board. When the Raspberry Pi is powered on, firmware is loaded, which sets up the hardware and initiates the bootloader. Subsequently, the bootloader finds the OS image on the microSD card or any other supported storage device; it loads the Linux kernel into the RAM, and thus the system is started and is available for use [10][20][24]. After the operating system boots, the Raspberry Pi OS takes over core system functions like process scheduling, memory allocation, peripheral control, and network communication. The System on Chip (SoC) delivers CPU and GPU resources, with memory being shared dynamically between them to optimize performance. The latest models are using dynamic frequency scaling to be able to adjust the processor speed according to the workload and thus still keep the operation energy efficient even though the performance is stable. The Raspberry Pi is an interactive computing platform. It supports standard input and output interfaces such as HDMI, USB ports for keyboard and mouse, and wired or wireless network connectivity. Users may also run desktop applications, develop software, and execute scripts using programming environments such as Python, Scratch, Greenfoot and BlueJ. All these environments being available enables users for both beginner level and advanced programming tasks, thus facilitating education and research.

One of the main features of the Raspberry Pi based tool is how it can directly communicate with the external hardware via the GPIO header. The GPIO pins are able to be set up as digital inputs or outputs and even allow UART, SPI, and IC communication protocols. Using these interfaces, the Raspberry Pi gets the data from the sensors, uses software algorithms to handle the input, and then turns on actuators like motors, relays, or displays. The Raspberry Pi can be the controller in embedded systems, IoT devices, or automation applications because of this close hardware and software integration. The Raspberry Pi based tool combines general purpose computing with real time hardware interaction. It operates on a low cost and energy efficient platform. Such a working mechanism allows the use of the Raspberry Pi as an educational computer, an embedded controller, a data acquisition unit, or an edge computing device in different application domains [4][6] [14].

## 7. SYSTEM ARCHITECTURE DIAGRAM

### 7.1 Raspberry Pi–Based System Architecture

The Raspberry Pi based system architecture of the platform is designed in a modular and layered manner, incorporating hardware, firmware, operating system, and application layers. All these layers are integrated into a compact, single board computing system. The architecture ensures smooth interaction

between software applications and external hardware components using standard interfaces.

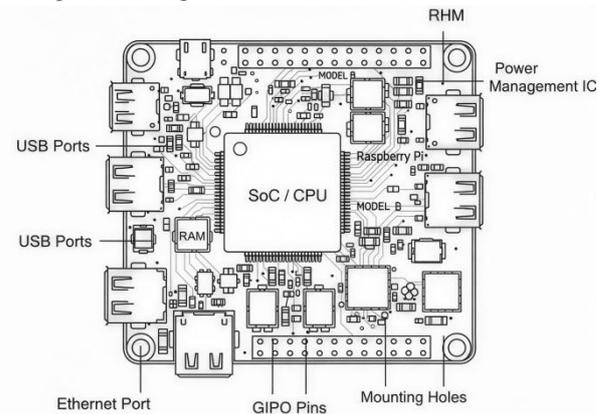


Fig 3: Raspberry Pi Based System Architecture

### 7.2 Architecture Description

Architecture Description The central component of the architecture is the System on Chip (SoC), which combines the ARM, based CPU, GPU, and memory controller. The SoC delivers both the CPU and GPU computational capabilities and graphics processing while allowing system memory to be shared dynamically between CPU and GPU resources. The firmware and bootloader layer is responsible for hardware initialization and operating system kernel loading from the microSD card or any supported storage device during system startup. On top of this layer, the operating system layer, generally Raspberry Pi OS (a Debian, based Linux distribution), is responsible for process scheduling, memory allocation, networking, and peripheral access. Device drivers and system libraries serve as intermediaries between the operating system and hardware interfaces, thus enabling communication with USB devices, display units, networking modules, storage media, and GPIO pins [11][18][24]. The hardware interface layer reveals the physical connection possibilities to the user via the standard interfaces like HDMI, USB, Ethernet, Wi-Fi, Bluetooth, and the 40-pin GPIO header. The Raspberry Pi connects to the outside world through sensors and actuators via the GPIO interface and also uses the communication protocols (UART, SPI, IC) for this purpose. The application layer is made up of user level programs and development environments that can perform simple programming educational tasks, as well as embedded control and IoT applications [4][5].

## 8. ADVANTAGES AND DISADVANTAGES

### 8.1 Advantages

The Raspberry Pi is a low, cost and affordable computing platform, which makes it very accessible to students, educators, researchers, and developers. Thus, it allows large scale deployment in educational institutions and resource constrained environments without a heavy financial burden. Moreover, its small size and energy efficient design enable it to use much less power than a traditional desktop computer, and therefore it is very suitable for embedded systems, Internet of Things (IoT) devices, and battery powered scenarios. The presence of versatile hardware features, in particular, the General purpose Input/Output (GPIO) pins, facilitates direct interfacing with sensors, actuators, and electronic components through standard communication protocols such as UART, SPI, and IC thus, the Raspberry Pi can be a computing and hardware control platform simultaneously. Besides that, proper

software and operating system support through Raspberry Pi OS and other Linux, based as well as third party operating systems gives a stable and flexible development environment, while the support of multiple programming languages and educational tools makes it an easy to use device for learning, research, and development. An extensive global community continuously strengthens the platform by providing tutorials, libraries, documentation, and open source projects, thus greatly facilitating learning, troubleshooting, and innovation. Thanks to these features, the Raspberry Pi serves as the backbone for numerous applications in areas such as education, robotics, IoT, home automation, industrial automation, multimedia systems, and edge computing, to name a few, thereby proving its adaptability and scalability [1] [3] [10].

## 8.2 Disadvantages

Despite its versatility, the Raspberry Pi has some drawbacks. The performance of the Raspberry Pi has been improved quite a lot from one generation to another; however, a Raspberry Pi device still has less computational power than a normal desktop or laptop computer, which limits its use for highly intensive workloads. The shared memory architecture between the CPU and GPU can also slow down performance in the case of graphics intensive or memory demanding applications, especially on lower end models. MicroSD cards are the common storage medium for most Raspberry Pi boards. These cards are, in general, slower and less reliable than dedicated solid state storage solutions. Over time, card corruption can impact system stability. Moreover, since Raspberry Pi runs a general purpose Linux operating system, it does not provide strict real time performance guarantees; thus, it cannot be used for time critical or deterministic applications. To prevent thermal throttling during long periods of heavy use, extra cooling may be needed. Also, the smaller models have few built in ports, so external hubs or expansion boards may be needed [1][15][17].

## 9. COMPARATIVE ANALYSIS

### 9.1 Comparative Analysis of Raspberry Pi Models

The Raspberry Pi family includes multiple hardware models designed for different levels of computational performance and energy efficiency. Although all Raspberry Pi boards use ARM-based processors and support Linux-based operating systems, their hardware configurations vary significantly in terms of CPU architecture, memory capacity, connectivity options, and power consumption. These differences determine their suitability for different embedded and IoT applications. The Raspberry Pi Zero and Zero W are compact and low-cost boards designed for lightweight embedded systems and IoT sensor nodes. These models use a single-core ARM11 processor with 512 MB RAM, providing limited computational capability but extremely low power consumption. Because of these characteristics, they are commonly used in small-scale IoT deployments, sensor monitoring systems, and battery-powered devices [10][15]. The Raspberry Pi Zero 2 W improves processing performance by integrating a quad-core Cortex-A53 processor while maintaining a small form factor. This architectural improvement allows the board to support moderate workloads such as multi-sensor data collection and lightweight edge processing tasks while still maintaining relatively low energy consumption [10][12]. The Raspberry Pi 3B and 3B+ models provide a balance between performance and power efficiency. These boards include quad-core Cortex-A53 processors, 1 GB RAM, and integrated Wi-Fi, Bluetooth, and Ethernet connectivity, making them suitable for

educational environments, rapid prototyping, and small-scale automation systems [3] [10][13]. The Raspberry Pi 4B offers the highest performance among the Raspberry Pi models. It features a quad-core Cortex-A72 processor with memory options up to 8 GB, along with USB 3.0 and Gigabit Ethernet support. These improvements allow the Raspberry Pi 4 to handle more demanding workloads such as multimedia processing, edge computing applications, and local machine learning inference tasks. However, higher performance also results in increased power consumption and thermal output [10] [12][13].

**Table 1. Comparative Analysis of Raspberry Pi Models [1][2][10][12]**

Parameter	Pi Zero / Zero W	Pi Zero 2 W	Pi 3B / 3B+	Pi 4B
Processor	Single core ARM11	Quad core Cortex A53	Quad core Cortex A53	Quad-core Cortex-A72
RAM	512 MB	512 MB	1 GB	Up to 8 GB
Connectivity	Wi-Fi (Zero W)	Wi-Fi, Bluetooth	Wi-Fi, Bluetooth, Ethernet	Wi-Fi, Bluetooth, Gigabit Ethernet
Performance	Low	Medium	Medium	High
Power Consumption	Very Low	Low	Moderate	Higher
Typical Applications	Basic IoT, embedded	Compact embedded, IoT	Education, prototyping	Multimedia, edge computing

### 9.2 Comparative Analysis of Raspberry Pi and NVIDIA Jetson

Single-board computers such as Raspberry Pi and NVIDIA Jetson are widely used in embedded and edge computing systems. Although both platforms support Linux-based operating systems and ARM architectures, they are designed for different computational purposes. The Raspberry Pi platform focuses on affordability, accessibility, and general-purpose embedded computing. It is widely used in educational environments, IoT systems, and low-cost prototyping platforms due to its low power consumption and strong community support [1][3][5]. In contrast, the NVIDIA Jetson platform is designed for high-performance artificial intelligence and machine learning applications. Jetson boards combine ARM CPUs with powerful NVIDIA GPUs that support CUDA and TensorRT frameworks for accelerated parallel computing. These capabilities make Jetson platforms suitable for computer vision, robotics, and edge AI applications [25][26][27]. While Jetson devices provide significantly higher computational performance for AI workloads, they generally require higher power consumption and greater system cost compared with Raspberry Pi boards. Therefore, Raspberry Pi remains more appropriate for cost-sensitive IoT systems, whereas NVIDIA Jetson is preferred for AI-driven edge computing environments [13][27].

**Table 2. Comparative Analysis of Raspberry Pi and NVIDIA Jetson [1] [10][13][25][28][26][27].**

Parameter	Raspberry Pi	NVIDIA Jetson
Primary Focus	Education, IoT, embedded systems	AI, machine learning, robotics

Processor	ARM Cortex-A series CPU	ARM CPU + NVIDIA GPU
GPU Capability	Basic graphics processing	CUDA enabled GPU with AI acceleration
AI/ML Support	Software based, limited	Hardware accelerated (CUDA, TensorRT)
Power Consumption	Low (battery friendly)	Moderate to high
Cost	Low and affordable	Relatively expensive
Ease of Use	Beginner friendly	Requires advanced expertise
Community Support	Very large educational community	Strong developer/AI community
Typical Applications	Education, home automation, IoT	Autonomous robots, vision systems, edge AI

## 10. APPLICATIONS

Application Usage Distribution

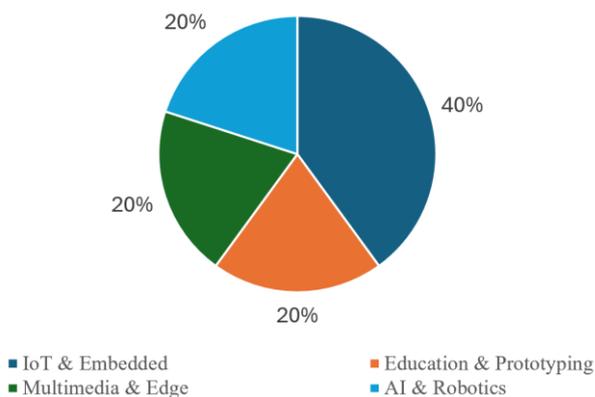


Fig 4: Application Usage Distribution

The Raspberry Pi is a small, affordable tool widely used in classrooms to teach programming, computer science, and electronics. With tools like Python, Scratch, Greenfoot, and BlueJ, students can learn by building projects, making it a key platform for hands on STEM education [3][7][8].

Raspberry Pi is widely used in embedded systems and IoT projects as a controller and data processor. Its GPIO pins and support for protocols like UART, SPI, and I<sup>2</sup>C make it easy to connect sensors and actuators, ideal for applications like environmental monitoring, smart farming, and remote sensing [5][6][23].

Raspberry Pi is used at home to control lights, monitor energy, run security systems, and manage appliances. Its affordability, reliability, and multimedia features also let it serve as a media center, digital sign, or lightweight desktop.

Raspberry Pi and its Compute Modules are increasingly used in factories and labs for industrial monitoring, control systems, and edge computing. Their flexible hardware and Linux

support make them reliable for Industry 4.0 projects, showing their versatility from education to automation [29][30][31][32].

## 11. CONCLUSION

This research examines the Raspberry Pi, exploring its design, functionality, strengths, and applications. Originally a simple teaching tool, it has evolved into a versatile single board computer used in classrooms, smart homes, IoT projects, and industrial setups. Its low cost, small size, minimal power use, GPIO pins, and strong community support make it ideal for learning, prototyping, and simple computing tasks. Compared to NVIDIA Jetson boards, Raspberry Pi is best for affordable, energy efficient, and easy to use projects, while Jetson excels in AI, robotics, and computer vision but is costlier and more complex. The choice depends on project needs, including computing power, energy, budget, and complexity, but for most educational and general embedded projects, Raspberry Pi offers the most practical and accessible solution [1][3][4][6]. Choosing the right computing platform depends on your budget, power availability, performance needs, and ease of use or scalability. After digging into the details, it's clear that the Raspberry Pi stands out for education, embedded systems, IoT, or any project that doesn't demand tons of processing power. It's cheap, small, doesn't eat up a lot of energy, and you get handy GPIO pins for hardware tinkering. Plus, the community support is massive, which is a lifesaver when you're learning or prototyping in places where resources are tight. NVIDIA Jetson boards offer significantly higher GPU powered performance. They shine in heavy duty AI work like computer vision, deep learning, and robotics. But they cost more, draw more power, and honestly, they're a bit trickier to set up. So, while Jetsons are perfect when you need that extra horsepower, they're not the best fit for everyday projects or classrooms. [13][25][27]. Considering the study's goals, the Raspberry Pi emerges as the best option. It's affordable, easy to scale, and just works for most practical computing needs. NVIDIA Jetson only really makes sense if you need serious AI power otherwise, it's overkill. All in all, Raspberry Pi does a great job connecting what you learn in the classroom to actually building real embedded systems.

## 12. FUTURE SCOPE

The Raspberry Pi continues to evolve, with faster processors, more memory, and better connectivity, making it increasingly useful for edge computing, real time data processing, and industrial automation. Its low power use, expanding hardware add-ons, and growing AI support make it ideal for hands on STEM education, AI learning, and research projects. Future developments aim to improve reliability, performance, and storage, further bridging the gap between traditional embedded systems and advanced AI powered devices. Raspberry Pi is poised to keep driving affordable innovation, education, and technology research for years to come. [6] [17] [29][27][33].

## 13. REFERENCES

- [1] E. Upton and G. Halfacree, Raspberry Pi User Guide, 4th ed. Hoboken, NJ, USA: Wiley, 2016.
- [2] Wikipedia Contributors, "Raspberry Pi," Wikipedia, The Free Encyclopedia, 2025. [Online]. Available: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [3] S. Balon and M. Simic, "Using Raspberry Pi in computer science education," in Proc. MIPRO, 2019, pp. 1230–1235, doi:10.23919/MIPRO.2019.8756967.
- [4] D. Molloy, Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux, 2nd ed. Wiley, 2016.

- [5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [6] L. Da Xu, W. He, and S. Li, "Internet of Things in industries," *IEEE Trans. Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [7] M. Kölling, "Educational programming environments on the Raspberry Pi," *Electronics*, vol. 5, no. 3, 2016, doi:10.3390/electronics5030033.
- [8] M. Resnick et al., "Scratch: Programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.
- [9] J. Voogt et al., "Computational thinking in compulsory education," *Educational Technology & Society*, vol. 18, no. 2, pp. 1–16, 2015.
- [10] Raspberry Pi Foundation, "Raspberry Pi Documentation," 2024. [Online]. Available: <https://www.raspberrypi.com/documentation>
- [11] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. Morgan Kaufmann, 2019.
- [12] S. Gamess and J. Hernandez, "Performance evaluation of Raspberry Pi models for CPU-intensive workloads," *IEEE Conf. on Emerging Technologies*, 2022.
- [13] P. Raj and A. Raman, "Performance comparison of single-board computers for edge computing," *IEEE Access*, vol. 8, pp. 211245–211256, 2020.
- [14] R. Want, B. Schilit, and S. Jenson, "Enabling the Internet of Things," *IEEE Computer*, vol. 48, no. 1, pp. 28–35, 2015.
- [15] P. Bekaroo and A. Santokhee, "Power consumption analysis of Raspberry Pi devices," *International Journal of Computer Applications*, vol. 150, no. 3, pp. 1–6, 2016.
- [16] A. Carroll and G. Heiser, "An analysis of power consumption in embedded systems," *USENIX ATC*, 2010.
- [17] S. Mittal, "A survey of energy-efficient techniques in embedded systems," *IEEE Trters*, vol. 63, no. 9, pp. 2209–2230, 2014.
- [18] A. S. Tanenbaum and T. Austin, *Structured Computer Organization*, 6th ed. Pearson, 2013.
- [19] Raspberry Pi Foundation, "Raspberry Pi OS Overview," 2024. [Online]. Available: <https://www.raspberrypi.com/software>
- [20] R. Love, *Linux Kernel Development*, 3rd ed. Addison-Wesley, 2010.
- [21] Debian Project, "Debian GNU/Linux," [Online]. Available: <https://www.debian.org>
- [22] S. Aaron, A. Blackwell, and S. Burnard, "The development of Sonic Pi," *Journal of Music, Technology & Education*, vol. 9, no. 1, pp. 75–90, 2016.
- [23] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey," *Information Systems Frontiers*, vol. 17, pp. 261–274, 2015.
- [24] W. Stallings, *Operating Systems: Internals and Design Principles*, 9th ed. Pearson, 2018.
- [25] NVIDIA Corporation, "NVIDIA Jetson platform overview," 2024. [Online]. Available: <https://developer.nvidia.com/embedded-computing>
- [26] NVIDIA Corporation, "CUDA Toolkit Documentation," 2024. [Online]. Available: <https://docs.nvidia.com/cuda>
- [27] W. Shi et al., "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [28] NVIDIA Corporation, *Jetson Nano Developer Kit User Guide*, 2023.
- [29] S. Hankel and B. Rexroth, "The reference architectural model Industrie 4.0," *ZVEI*, Germany, 2015.
- [30] TECHBASE, "ModBerry industrial computer based on Raspberry Pi," 2018.
- [31] SUSE, "SUSE Linux Enterprise support for Raspberry Pi," 2018.
- [32] J. Redmon et al., "You Only Look Once: Unified real-time object detection," *IEEE CVPR*, 2016.
- [33] Owens, John D., Mike Houston, David Luebke, Simon Green, John E. Stone, and James C. Phillips. "GPU computing." *Proceedings of the IEEE* 96, no. 5 (2008): 879-899.