

# Enhancing AI Education Through Practical IoT Applications and Gesture Recognition

Yoshiyasu Takefuji

Faculty of Data Science, Musashino University, Tokyo 135-8181, Japan

© Higher Education Press 2025

**Abstract** The burgeoning field of artificial intelligence (AI) has led to the development of new educational approaches, particularly in the realm of gesture recognition and Internet of Things (IoT) device control. Despite these rapid advancements, practical applications and hands-on learning opportunities remain scarce. Many learners, including students, educators, and software engineers, have limited knowledge of hardware due to a lack of exposure to IoT, AI libraries, and human-machine interfaces. This gap is exacerbated by the absence of demonstrated examples and academic hardware journals. A significant challenge lies in the cumbersome process of updating IoT firmware, which is essential for incorporating new features. This paper introduces a novel solution that eliminates the need for firmware updates. By leveraging the Python Firmata library, applications on the host computer can be updated without affecting the IoT device's firmware. The Firmata protocol enables seamless communication between the host and microcontroller, facilitating real-time interactions. Additionally, the abstraction capabilities of AI libraries, such as MediaPipe, simplify complex tasks into manageable components. For instance, MediaPipe provides precise hand landmark coordinates, enabling direct control of simple Arduino Nano devices without requiring detailed calculations. The paper's contributions are valuable for a wide range of professionals, including mathematicians, AI engineers, software engineers, hardware engineers, IoT engineers, and network programmers.

**Keywords** artificial intelligence (AI), Internet of Things (IoT), Firmata protocol, MediaPipe library, updating IoT firmware

## 1 Introduction

Despite the rapid advancement of Internet of Things

(IoT) and artificial intelligence (AI) technologies, there remains a significant gap in accessible tutorials for both beginners and professionals who want to implement AI-controlled IoT devices. The lack of straightforward, programming-free guides for AI-IoT integration has become a major obstacle in both industrial applications and academic settings. Many professionals and students are seeking user-friendly tutorials that explain how to harness AI for IoT control without requiring extensive programming knowledge. While these technologies continue to evolve, the absence of comprehensive, beginner-friendly tutorials has created a significant learning barrier for potential adopters. Both industry practitioners and academic institutions are calling for step-by-step guides that bridge the gap between AI technology and IoT device control without heavy reliance on programming skills. The development of user-friendly tutorials could significantly accelerate the adoption of these technologies in both educational and industrial settings.

This paper introduces a novel approach for controlling IoT devices using hand gestures and readily available open-source AI libraries. By leveraging AI libraries like MediaPipe and pyFirmata, the system simplifies development, reduces programming complexity, and enables real-time control. This approach eliminates the need for C++ programming on IoT devices, making it more accessible to a wider range of developers and learners.

Despite the rapid advancements in AI and IoT, there is a noticeable absence of academic journals that combine these technologies with practical examples for classes (Chataut et al., 2023). The gap in the literature highlights the need for more comprehensive educational resources that demonstrate the integration of AI libraries with IoT devices. The paper also demonstrates the ability to update IoT firmware through the host application, providing a cost-effective solution. This work has significant educational value, serving as a practical example for AI engineers, software engineers, hardware engineers, and network

Received February 10, 2025; revised April 10, 2025; accepted May 7, 2025

Yoshiyasu Takefuji (✉)

E-mail: takefuji@keio.jp

programmers. By addressing the lack of combined IoT and AI library examples in academic journals for classes (Chataut et al., 2023; Kitsios et al., 2023), this paper aims to bridge the gap and provide valuable insights for both educators and students. Overall, it presents a valuable contribution to the field of gesture-based IoT control (Sadi et al., 2022; Vijaya Kumari et al., 2024), offering a practical and accessible solution for various applications in beginner-level classes.

With the advent of generative AI (GenAI), novice and non-programmers can now create accurate Python code. However, due to the limited examples and insufficient training in IoT interface programming with AI libraries such as MediaPipe, GenAI does not yet enable users to create advanced applications, but it does facilitate useful human-computer interactions. Users must possess fundamental skills in handling the pyFirmata library for interfacing between humans and AI with IoT devices. This paper showcases an example.

With the progress of open-source software, AI with IoT has advanced science and technology. However, due to the dispersion of information on AI and IoT, we are lagging behind in AI and IoT technologies in the classroom (Alahi et al., 2023; Chataut et al., 2023).

Japan clearly stated that they are falling behind in AI research (Ministry of Internal Affairs and Communications of Japan, 2016; Prime Minister of Japan and His Cabinet, 2019). Companies deploying AI systems will need human workers, for example, to train the technology, create algorithms, and explain the complex inner workings of AI to non-technical colleagues, and AI will need so-called sustainers to ensure that the AI system works properly (Daschle & Beier, 2018). The growing influence of AI in nearly every aspect of our lives has made it essential for everyone to grasp the fundamentals of how this technology works. Unfortunately, the current education system, including policymakers and educators, is lagging behind the rapid advancements in AI (Almasri, 2024; Sanusi et al., 2024).

However, the shadow-side effects of AI should be aware of (Akgun & Greenhow, 2022; Al-Zahrani, 2024). While AI can revolutionize education, it poses challenges like privacy concerns, ethics, data security issues, and potential bias in algorithms. Automation may reduce or change teachers' roles and limit personalized interactions (Ghamrawi et al., 2024). Overreliance on AI could affect students' critical thinking and social skills (Zhai et al., 2024). The opaque nature of AI algorithms complicates understanding decisions. These challenges necessitate careful consideration and regulation to maximize benefits and minimize harms in educational systems (Zhai et al., 2024).

The Japanese government has confessed that

Japan lags far behind in AI and IoT technologies (Ministry of Internal Affairs and Communications of Japan, 2016; Prime Minister of Japan and His Cabinet, 2019). In other words, Japanese universities have not been able to produce AI practitioners to utilize AI and IoT technologies.

AI is not only for computer scientists and engineers but also for all professionals and novice practitioners to thrive in the age of AI (Sanusi et al., 2024). Open-source AI is a great example which is publicly available for commercial and non-commercial use under various open-source licenses. With the rapid progress of open-source AI, there are many useful AI libraries which contain various technologies that are helpful for product teams, independent App developers, and enterprises. However, it is difficult for novice programmers to find hands-on examples using the complex libraries.

In this paper, the innovation in open-source libraries allows users to create a simple state-of-the-art application within a few days or hours which is called rapid open-source prototyping. In traditional software development, programmers must create programs from scratch. Therefore, the traditional software development is costly and time-consuming.

With the rapid open-source prototyping, a variety of applications have been reported (Baillargeon et al., 2022; Ching et al., 2022; Günther et al., 2021; Liegmann et al., 2021). In rapid open-source prototyping, selecting right libraries and gluing them together with minimum effort are essential for programmers and practitioners.

However, the existing studies on rapid open-source prototyping rarely discuss using AI libraries to eliminate programming languages to save cost and time. This paper explores the apply of a pre-learned AI library on gesture recognition and eliminate C++ programming in the IoT product and complex tasks in programming in rapid open-source prototyping.

Existing IoT devices have major problems updating IoT firmware, and there is a growing industry demand for a solution (Al Kabir et al., 2023; Lu et al., 2023). From the rapid open-source prototyping, MediaPipe and pyFirmata libraries enable users to build prototype applications in a short period of time. In order to update the IoT firmware using the proposed method, there is no need to modify the firmware on the IoT device itself; instead, updating the host program can alter and manage the functionality of the IoT device. In the proposed method, the use of the pyFirmata library in Python on the host and Firmata library on the IoT device enables users not only to build mission-critical applications in real time without C++ language but also to eliminate the IoT latency in AI control, which can significantly contribute to industry for time and cost saving.

With the advent of GenAI, novice and non-programmers can now create accurate Python code through multiple interactions with AI, despite its imperfections. However, GenAI is not applicable for interfacing between MediaPipe and IoT device control due to an insufficient number of training samples.

This paper challenges and introduces two hands-on AI applications using MediaPipe library (Azhand et al., 2021; Lugaresi et al., 2019) and Open-Source Computer Vision Library (OpenCV) (Guilbeault et al., 2021; OpenCV, 2021). OpenCV is a rich library of AI algorithms intends to address real-time computer vision functionalities. It is used for displaying the result of hand gesture recognition on the screen in real time. MediaPipe offers cross-platform and customizable machine learning solutions for live and streaming media. In the proposed AI applications, MediaPipe is the first library used for real-time hand gesture recognition for controlling devices including light-emitting diodes (LEDs) and servomotors on Arduino Nano. If AI controls cause noticeable control delays in IoT devices, they cannot be used in industrial products. Therefore, it would be very useful for practitioners and software developers to propose two applications that solve the problem of IoT device latency in AI control.

Despite the growing significance of integrating AI and IoT, there remains a notable gap in educational resources for beginners and novices. The scarcity of courses specifically focuses on this integration results in limited awareness of how to effectively interface state-of-the-art AI libraries with IoT devices. Furthermore, the global trend of diminishing hardware courses exacerbates this challenge, restricting opportunities for learners to engage with practical applications of AI and IoT technologies. This study aims to fill this educational void by thoroughly exploring the interface between AI and IoT, addressing current challenges, and proposing solutions. By doing so, we hope to contribute valuable insights to the literature and promote greater understanding and accessibility in this pivotal area of technology.

The asynchronous I/O protocol via USB interface between the host personal computer (PC) and Arduino Nano is cumbersome for novices and is solved using Firmata library on Arduino and pyFirmata library on the host. Since Firmata is a protocol for communicating with microcontrollers from software on a host computer, Firmata library is useful and indispensable for directly controlling the devices on Arduino by Python programs on the host PC. In other words, novice programmers only need to install StandardFirmata on their Arduino Nano and then concentrate on Python programming on their own host PC. Without Firmata (pyFirmata) library, all programmers must suffer from synchronization and

communication between the host PC and Arduino Nano.

Many researchers remain unfamiliar with the design for the combination of gesture recognition and IoT devices. This division of labor is prevalent in industry settings, where software engineers typically focus on software development, while hardware engineers and network specialists concentrate on IoT device implementation. Similarly, AI engineers might lack familiarity with IoT infrastructure and networking protocols. The proposed applications are intentionally simple yet innovative, addressing specific niche use cases that combine real-time gesture recognition with IoT device control. Specifically, this paper advances interdisciplinary collaboration by bridging gaps between AI, software, hardware, and networking domains. Its contributions are particularly significant for AI engineers, software engineers, hardware engineers, and network programmers.

The author was in charge of turning 50,000 audio/visual engineers into AI or IoT engineers for the automotive industry such as autonomous driving or connected vehicles in Japan for four years. In the proposed applications, learners can learn the recipe for state-of-the-art AI using IoT devices. Engineering companies have expressed a strong demand for AI and IoT courses to help reskill their engineers with the latest technologies. Designed for beginners, these courses have encouraged students to actively engage in writing their own Python code to demonstrate their understanding and create innovative applications. The AI techniques discussed in this paper utilize pretrained machine learning models for gesture recognition, which are lightweight enough for students' devices to manage effectively. To gain deeper insights into the long-term benefits of this technology, the author recognizes the critical need for a longitudinal study. Such an analysis would track the educational outcomes of learners, focusing on improvements in knowledge retention and application skills over time.

For firmware updates, this paper proposes a new solution for 8-bit Arduino microcontrollers. Without updating the firmware of 8-bit Arduino microcontrollers, device functionality updates can be managed by a Python application on the host, saving time and cost at the same time.

This paper also introduces two useful Python libraries that enable novice and non-programmers to control IoT devices effectively. By combining MediaPipe with Firmata, the real-time control of remote IoT devices become possible. Novice hardware learners can achieve their desired outcomes and manipulate IoT devices with ease.

Although the integration of AI and IoT has garnered significant attention and a plethora of articles have been published on the subject, many of these

studies delve into complex interactions that necessitate robust machine learning models (Alahi et al., 2023; Chen & Dai, 2024; Sehito et al., 2025; Wu, 2025). While these approaches are valuable, they often require substantial machine learning resources and time commitments, which can be prohibitive for certain applications. In contrast, this paper advocates for the use of pretrained models, which can effectively mitigate the burdens associated with resource-intensive machine learning processes while still delivering reliable performance. Our approach aims to streamline implementation, making it more accessible to practitioners and researchers alike. This paper intends to fill the gap by providing a practical solution that emphasizes efficiency without compromising on effectiveness.

The introduction of MediaPipe represents a significant step forward in the efficiency and accessibility of developing robust AI applications, particularly for gesture recognition and real-time IoT control. One of MediaPipe's most notable contributions is the substantial reduction in time and resources traditionally required for training machine learning models. By employing pre-trained models, developers can bypass the labor-intensive stages of data collection and model training, which is especially crucial in real-time applications where responsiveness and accuracy are paramount. This allows developers to concentrate on application design and integration, effectively enhancing both productivity and speed of innovation. Consequently, this capability democratizes access to AI development, making it more approachable for individuals who may lack extensive expertise in machine learning.

MediaPipe's representation of hand gestures through two-dimensional coordinates exemplifies the effectiveness of abstraction in complex problem domains. Gesture recognition typically relies on three-dimensional data, however, by reducing this to two dimensions, the author does not compromise the quality of recognition for a wide range of practical applications. This strategic abstraction simplifies the development process while optimizing computational efficiency, as demonstrated in our virtual switch application, which accurately detects finger placement in a defined area. The ability to operate effectively with reduced input while maintaining task performance illustrates a key advancement in AI design for real-world challenges. The applications built using MediaPipe, including the virtual switch and servo motor controls, showcase innovative use cases that integrate gesture recognition with IoT technologies. For instance, the virtual switch can be activated with simple finger gestures, while the precision of servo motor controls based on finger position highlights the seamless interaction between AI and hardware. This capability for real-time control opens new avenues for

developing user-friendly interfaces for complex systems, thus making technology more intuitive and accessible to end-users.

By effectively combining MediaPipe with pyFirmata library, the workflow for IoT integration is significantly streamlined. This integration allows developers to manage Arduino-controlled devices without the complexities of C++ programming, further democratizing access to IoT development. Consequently, creators can focus more on design and functionality rather than intricate programming protocols. This paper aims to lower the barriers to entry for developers by providing clear examples of implementing MediaPipe in practical applications. The author documents the setup process for Arduino, including software installation and hardware configuration, thereby facilitating a smoother integration of gesture recognition with IoT systems. As a crucial consideration, this paper selected cost-effective hardware components and open-source libraries to minimize project expenses. Importantly, the proposed methods accommodate diverse computing environments. However, the author recognized that users may encounter challenges related to library compatibility across different operating systems, which could affect the system's overall robustness in varying conditions.

## 2 Methods

Over the past 4 years, a total of 120 undergraduate students and more than 400 industry engineers have enrolled in the AI and IoT applications course to evaluate the proposed methods. Experimental comparisons reveal significant challenges with traditional approaches, particularly regarding latency and response times, which lead to considerable delays in interactions between AI and IoT devices. This paper's simplicity in AI-IoT integration offers unique advantages, such as reduced implementation complexity and faster deployment.

The tested environment for instructors, run on Windows 10 and 11, includes Arduino Nano, Arduino IDE 2.3.6, Python 3.9.18, and pyFirmata 1.1.0. Additionally, many students utilize various versions in the class structure, which spans 7 days with a total of 200 minutes each day, 100 minutes dedicated to instruction and 100 minutes for self-guided training and exercises for ongoing assessment. This course is an introductory class focused on understanding AI and IoT devices for undergraduate students. This paper has conducted numerous experiments and successfully implemented real-time communication between pretrained AI models and IoT devices using the proposed methods. However, it is important to note that this

paper did not perform a quantitative analysis, such as recognition accuracy or latency statistics, as the focus was primarily on implementation and understanding the concepts rather than on statistical evaluation.

After the initial 2 days of foundational learning, students begin to develop their own projects, allowing them to apply what they have learned in a practical context. On the final day, students showcase their projects in a demonstration that resembles an exciting competition, which not only highlights their achievements but also allows for peer and instructor feedback. To truly measure the long-term impact of this educational approach, this paper plan to implement a follow-up assessment after a period of time, allowing the author to track each student's progress and the skills they have retained. This will enable the author to gather valuable data on the effectiveness of the program and provide deeper insights into student learning trajectories of macOS.

Unlike C programming, Python users are not required to explicitly define all variables. Additionally, Python's syntax resembles natural English, which diminishes the need for flowcharts and diagrams, making the code more intuitive and easier to read. All information is publicly accessible on GitHub, offering educational resources that encompass the development of teaching materials, practical projects, and online courses. GitHub is publicly available for students, engineers and other users.

## 2.1 | MediaPipe: Knowledge Representation

In general, knowledge representation is at the core of AI systems. Various machine learning algorithms have been proposed, but teaching machine learning models with datasets is time-consuming and requires significant effort. This paper shows how to use pre-trained AI libraries such as MediaPipe for controlling real-time remote IoT devices without learning machine learning.

Abstraction in AI is a powerful concept that simplifies complex systems into more manageable

models. By focusing on essential features and ignoring unnecessary details, AI developers create simplified models that are easier to understand and implement. This process of abstraction allows AI systems to handle vast amounts of data and make decisions more efficiently. For example, in natural language processing, AI models use abstraction to distill the essence of human language, enabling them to understand and generate text without processing every grammatical rule. Similarly, in computer vision, abstraction helps AI systems recognize objects by focusing on key features rather than analyzing every pixel. Overall, AI abstraction enhances the ability of AI to tackle complex real-world problems while maintaining user-friendly interfaces (Conroy, 2024; Nahas, 2024).

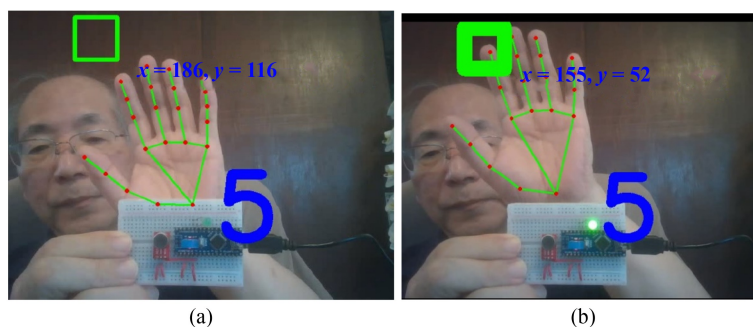
In the MediaPipe library, hand-gesture knowledge is represented by two-dimensional coordinates on the  $x$  and  $y$  axes. In general, gesture recognition is a three-dimensional problem, but for many gesture recognition problems, two-dimensional information is sufficient. Therefore, in the proposed hand-gesture recognition, time-series two-dimensional coordinates are used.

The proposed virtual switch application can recognize switch-on and switch-off by identifying whether the tip of index finger is within a predefined virtual zone (area). The servo motor control application determines the angle of the servo motor by identifying the horizontal axis of the index-finger.

MediaPipe runs lightly on inexpensive CPU computers without GPU modules. The MediaPipe library automatically generates 21 hand landmark points for five fingers represented by  $(x, y)$  coordinates, while the main program detects these points to control IoT devices in real-time. The specific figure of the hand landmarks will be shown in electronic supplementary material.

## 2.2 | Virtual Switch in the Air

A virtual switch is displayed on the left-top corner as shown in Figure 1. Red points in Figure 1 indicate the coordinates of 21 MediaPipe hand landmarks. The virtual switch square with 2 points  $(110,10)$  and



**Figure 1** Virtual switch in the air. (a) Green square is a virtual switch; (b) virtual switch is turned on.

(180,80) is defined by OpenCV function:

```
cv2.rectangle(image, (110, 10), (180, 80), (0, 255, 0), 5),
```

where (0,255,0) defines the color of the green line and the last number indicates the line width of 5.

Placing the tip of the index finger in the area of the virtual switch turns on the LED light as shown in Figure 1(b).

In `fvsw.py` program, as long as the following condition is satisfied, the LED will be turned on. The values of  $x$ -axis and  $y$ -axis can be obtained by the following program commands:  $x = \text{handLandmarks}[8][1]$  and  $y = \text{handLandmarks}[8][2]$ , respectively, where the number [8] indicates the tip of the index finger in the hand landmarks. The virtual switch zone can be defined by the following condition as an example:

```
if (x > 110 and x < 180) and (y > 10 and y < 80):.
```

In order to display the turned-on condition, the following command on the host will be executed in `fvsw.py` where the last number, 25 indicates the width of the square:

```
cv2.rectangle(image, (110, 10), (180, 80), (0, 255, 0), 25).
```

In Firmata library on Arduino, the value of nonzero will be written at pin number 2 on Arduino Nano which can cause the LED turning-on. The host program can turn on the LED on Arduino by the following command in Python: `b.digital[2].write(1)`. The command of `b.digital[2].write(0)` can turn off the LED.

Demonstration video is uploaded in Electronic Supplementary Material.

### 2.3 | Firmata Servo

In `fservo.py` application that controls the servo motor with MediaPipe and pyFirmata, the angle of the servo motor is controlled by the position of the index finger on the horizontal axis. One of the key features of the Firmata library is its ability to control servo motors in real time, making it possible to use it for mission-critical applications.

The angle of the servo motor is determined by the horizontal coordinates of the tip of the index finger. It is essential that programmers and developers must understand the data flow. Traditional data flow and data flow programming requires programmers to understand complex data flow protocols to control remote IoT devices.

In the proposed environment with MediaPipe and pyFirmata in Python and Firmata on Arduino, data flow can be easily established from MediaPipe to pyFirmata and pyFirmata to Firmata on Arduino.

In other words, the horizontal axis of the tip of

the index finger can change the angle of a servomotor. Figure 2 shows a screenshot of demonstration of `fservo.py`.



Figure 2 21 red points indicating hand landmarks.

In `fservo.py`, the  $x$ -axis value of the tip of the index finger can be obtained by the following command:

```
x = int(handLandmarks[8][1]/3),
```

where [8] indicates the tip of index finger in 21 hand landmarks.

The value of  $x$  will be written through Firmate library at digital pin#2 on Arduino Nano for controlling a servo motor: `b.digital[2].write(x)`.

Demonstration video of `fservo.py` is uploaded in Electronic Supplementary Material.

### 2.4 | Standard Firmata

There are 2 libraries such as pyFirmata in Python and Firmata on Arduino. Combining pyFirmata with Firmata eliminates C++ programming on the Arduino. This is because the pyFirmata library allows all control on the Arduino to be done via Firmata.

Programmers and designers of AI systems that control remote IoT devices always need to simplify their systems in terms of programming development time and cost. The 2 proposed applications are good examples of time and cost savings. Detailed guidance to use Firmate library will be showed in Electronic Supplementary Meterial.

## 3 Discussion

When using Firmata on Arduino in conjunction with pyFirmata on a PC, excellent response times and latencies consistently under 100 milliseconds were observed, with no missed acknowledgments. In contrast, when the PC was operated without pyFirmata and the Arduino without Firmata, significant delays were observed, frequently exceeding 1.5 seconds and often surpassing 2 seconds. This stark difference highlights the effectiveness of the Firmata and

pyFirmata combination in minimizing latency and ensuring reliable communication. However, to bolster the claims, a more rigorous quantitative evaluation will be conducted, incorporating detailed measurements of latency, response times, and accuracy across different setups.

The introduction of MediaPipe into the realm of knowledge representation for AI systems marks a significant advancement in the efficiency and accessibility of developing robust AI applications, particularly in gesture recognition and real-time IoT control. The contributions of this library to the field are multifaceted, addressing both the complexities of conventional machine learning processes and the practical demands of real-time applications.

One of the most notable contributions of MediaPipe is the reduction of time and resources traditionally required for training machine learning models. As identified, the deployment of pre-trained models allows developers to bypass the labor-intensive stage of data collection and model training. This is particularly crucial in real-time applications, where responsiveness and accuracy are paramount. By leveraging MediaPipe's pre-trained models, developers can immediately focus their efforts on application design and integration, enhancing productivity and speed of innovation. This capability transforms the landscape of AI development, making it accessible to a broader audience, including those who may not possess deep expertise in machine learning.

The representation of hand gestures through two-dimensional coordinates in MediaPipe demonstrates the effectiveness of abstraction in complex problem domains. While gesture recognition typically requires three-dimensional data, the reduction to two dimensions does not compromise the quality of recognition for many practical applications. This strategic abstraction simplifies the development process and optimizes computational efficiency, as seen in the virtual switch application that detects finger placement in a defined area. The ability to operate effectively with reduced input without losing essential task performance illustrates a key advancement in how AI can be designed to address real-world challenges.

The applications built using MediaPipe, such as the virtual switch and servo motor controls, exemplify innovative use cases that combine gesture recognition with IoT technologies. The virtual switch can be activated with simple finger gestures, and the precision with which the servo motor can be controlled based on finger position showcases the seamless interaction between AI and hardware. This real-time control capability opens avenues for developing user-friendly interfaces for complex systems, making technology more intuitive and accessible to end-users.

By effectively combining MediaPipe with the

pyFirmata library, the workflow for IoT integration has been notably streamlined. Developers can now manage Arduino-controlled devices without delving into the complexities of C++ programming. This further democratizes access to IoT development, allowing creators to focus on design and functionality rather than intricate programming protocols.

In providing clear examples of how to implement MediaPipe in practical applications, this work serves to guide developers through the initial hurdles of integrating gesture recognition with IoT systems. By documenting the setup process for Arduino, including software installation and hardware configuration, a significant barrier to entry is lowered. Developers can build upon the outlined methods to innovate further in both consumer and industrial applications where human-computer interaction is essential.

To minimize project costs, this study selected the most affordable hardware components along with open-source libraries. The proposed methods are designed to accommodate diverse computing environments, including Windows, macOS, and Linux. However, users may encounter challenges related to library compatibility across different operating systems, which could impact the overall robustness of the system in varying environmental conditions. Additional considerations, such as hardware compatibility and security concerns, should also be acknowledged to present a more balanced view of the technology's applicability.

It is essential to clearly articulate the limitations of the proposed method to set realistic expectations for future applications. Notably, the performance of the hardware selected for this project may impose certain constraints on the overall functionality and efficiency of the system. Additionally, while the gesture recognition capabilities are designed to be robust, accuracy may vary depending on environmental factors, lighting conditions, and the specific gestures being recognized. These limitations should be considered when applying this technology in real-world scenarios, as they may impact user experience and operational effectiveness. By acknowledging these challenges, this paper aims to provide a clearer understanding of the parameters within which the method can be effectively utilized.

Several key constraints and limitations still exist. First, the experiments in the paper were primarily conducted on Windows 10 and 11 operating systems, where we encountered various challenges related to compatibility and performance. Additionally, the diverse versions of macOS introduces further complications, as different versions may not support all functionalities uniformly. Second, accuracy in gesture recognition is critical for the effectiveness of the proposed approach. However, factors such as varying

lighting conditions, the speed of gestures, and sensor calibration can impact the reliability of the recognition algorithms, and the author is actively working to refine these aspects to enhance performance. Finally, the robustness of our system is another important consideration. While the author managed to address issues related to software dependencies by upgrading or downgrading Python and adjusting the versions of MediaPipe, pyFirmata, and Firmata on Arduino, the overall stability of the system may still be influenced by the hardware used, as compatibility between different components can lead to unforeseen challenges. By clearly discussing these limitations, this paper aims to provide a more comprehensive understanding of the challenges faced and the areas for improvement as the author continues to refine the educational approach and technology integration.

## 4 Conclusions

This paper presents a robust approach to integrating AI and IoT within educational contexts through the innovative use of MediaPipe and pyFirmata. Effective knowledge representation and rapid prototyping within an open-source environment are facilitated through the utilization of MediaPipe, a pre-trained library. The implementation of pyFirmata in Python, alongside Firmata on Arduino, eliminates the need for C++ programming, addressing latency issues while significantly reducing the time and costs associated with IoT system development.

The result indicate that the combination of these 2 libraries enables the creation of real-time IoT control systems tailored for mission-critical applications. Notably, pyFirmata successfully resolves the challenge of IoT firmware updates by allowing a unified update for the host application, rather than requiring individual updates for multiple devices. This enhancement simplifies the maintenance of IoT systems and increases their operational efficiency.

This paper also demonstrates the effectiveness of real-time hand gesture recognition using MediaPipe. Specific implementations, such as `fsw.py` for LED control and `fservo.py` for servo motor manipulation, showcase practical applications of this technology. These implementations highlight how intuitive interactions can be achieved with IoT devices through simple gestures, thereby enhancing user experience and accessibility.

Importantly, the use of Firmata library on Arduino Nano simplifies the development process for novice programmers. By installing StandardFirmata directly onto their Arduino boards, users can focus on Python programming without dealing with complex communication protocols, broadening accessibility for

those without extensive programming backgrounds.

Overall, this paper substantiates the effectiveness of real-time gesture recognition with MediaPipe in a Python environment, coupled with the utilization of StandardFirmata library for Arduino integration. The outcomes represent a unique contribution at the intersection of gesture recognition and IoT device control, advancing the fields of AI, software and hardware engineering, and network programming.

Looking ahead, there are exciting opportunities for future applications, particularly in the field of education. By incorporating our approach into smart classrooms, educators can leverage gesture-based controls for interactive learning experiences, facilitating a more engaging and hands-on environment for students. Future research could also explore enhancing gesture recognition accuracy and robustness under varying environmental conditions. Additionally, investigating the integration of more complex gestures and interoperability with other platforms and devices may enhance the versatility of our approach, establishing a solid foundation for developing sophisticated AI-driven IoT solutions applicable in educational and industrial contexts. This potential for application in diverse scenarios underscores the transformative impact of our methods on both teaching and learning experience.

Lastly, it is essential to articulate the limitations of the proposed method to set realistic expectations for future applications. The performance of the chosen hardware may impose constraints on the system's overall functionality and efficiency. Moreover, while the gesture recognition capabilities are designed to be robust, variability in accuracy may arise due to environmental factors, lighting conditions, and specific gestures being recognized. Acknowledging these limitations allows for a more nuanced understanding of the parameters within which our method can be effectively utilized, ultimately enhancing user experience and operational effectiveness.

**Acknowledgment** No funds, grants, or other support was received.

**Author Contributions** Yoshiyasu Takefuji completed this research and wrote the program and this article.

**Conflict of Interest** The author declares that he has no conflict of interest related to the content of this paper.

**Data Availability Statements** The author confirms that all data generated or analysed during this study are included in this published article.

**Electronic Supplementary Material** Supplementary material is available in the online version of this article at

<https://doi.org/10.1007/s44366-025-0066-7> and is accessible for authorized users.

## References

- Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI and Ethics*, 2(3), 431–440.
- Al Kabir, M. A., Elmedany, W., & Sharif, M. S. (2023). Securing IoT devices against emerging security threats: Challenges and mitigation techniques. *Journal of Cyber Security Technology*, 7(4), 199–223.
- Alahi, M. E. E., Sukkuea, A., Tina, F. W., Nag, A., Kurdthongmee, W., Suwannarat, K., & Mukhopadhyay, S. C. (2023). Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: Recent advancements and future trends. *Sensors*, 23(11), 5206.
- Almasri, F. (2024). Exploring the impact of artificial intelligence in teaching and learning of science: A systematic review of empirical research. *Research in Science Education*, 54(5), 977–997.
- Al-Zahrani, A. M. (2024). Unveiling the shadows: Beyond the hype of AI in education. *Heliyon*, 10(9), e30696.
- Azhand, A., Rabe, S., Müller, S., Sattler, I., & Heimann-Steinert, A. (2021). Algorithm based on one monocular video delivers highly valid and reliable gait parameters. *Scientific Reports*, 11(1), 14065.
- Baillargeon, P., Fernandez-Vega, V., Ortiz, L., Shumate, J., Marques, N., Deng, L., Spicer, T. P., & Scampavia, L. (2022). Rapid deployment of inexpensive open-source orbital shakers in support of high-throughput screening. *SLAS Technology*, 27(3), 180–186.
- Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and Industry 4.0. *Sensors*, 23(16), 7194.
- Chen, Z. L., & Dai, X. H. (2024). Utilizing AI and IoT technologies for identifying risk factors in sports. *Heliyon*, 10(11), e32477.
- Ching, E. J., Bornhoft, B., Lasemi, A., & Ihme, M. (2022). Quail: A lightweight open-source discontinuous Galerkin code in Python for teaching and prototyping. *SoftwareX*, 17, 100982.
- Conroy, G. (2024, September 20). *Do AI models produce more original ideas than researchers?* Available from Nature website.
- Daschle, T., & Beier, D. (2018, January 24). *The US is falling behind in artificial intelligence research.* Available from the Hill website.
- Ghamrawi, N., Shal, T., & Ghamrawi, N. A. R. (2024). Exploring the impact of AI on teacher leadership: Regressing or expanding? *Education and Information Technologies*, 29(7), 8415–8433.
- Guilbeault, N. C., Guerguiev, J., Martin, M., Tate, I., & Thiele, T. R. (2021). BonZeb: Open-source, modular software tools for high-resolution zebrafish tracking and analysis. *Scientific Reports*, 11(1), 8148.
- Günther, S., Müller, F., Hübner, F., Mühlhäuser, M., and Matviienko, A. (2021). ActuBoard: An open rapid prototyping platform to integrate hardware actuators in remote applications. In: *Proceedings of the 2021 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. New York: ACM, 70–76.
- Kitsios, F., Kamariotou, M., Syngelakis, A. I., & Talias, M. A. (2023). Recent advances of artificial intelligence in healthcare: A systematic literature review. *Applied Sciences*, 13(13), 7479.
- Liegmann, E., Schindler, T., Karamanakos, P., Dietz, A. & Kennel, R. (2021). UltraZohm—An open-source rapid control prototyping platform for power electronic systems. In: *Proceedings of 2021 International Aegean Conference on Electrical Machines and Power Electronics & 2021 International Conference on Optimization of Electrical and Electronic Equipment*. Piscataway: IEEE, 445–450.
- Lu, C. H., Liu, C. H., & Chen, Z. H. (2023). Secure and efficient firmware update for increasing IoT-enabled smart devices. *Journal of Ambient Intelligence and Humanized Computing*, 14(5), 4987–5000.
- Lugaresi, C., Tang, J. Q., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C. L., Yong, M. G., Lee, J., & et al. (2019). MediaPipe: A framework for building perception pipelines. *arXiv Preprint*, arXiv:1906.08172.
- Ministry of Internal Affairs and Communications of Japan. (2016). *2016 white paper on information and communications in Japan*. Available from Ministry of Internal Affairs and Communications of Japan website.
- Nahas, K. (2024, March 20). *Is AI ready to mass-produce lay summaries of research articles?* Available from Nature Index website.
- OpenCV. (2021, June 11). *OpenCV*. Available from OpenCV website.
- Prime Minister of Japan and His Cabinet. (2019, June 11). *AI strategy 2019*. Available from Prime Minister's Office of Japan website.
- Sadi, M. S., Alotaibi, M., Islam, M. R., Islam, M. S., Alhmiedat, T., & Bassfar, Z. (2022). Finger-gesture controlled wheelchair with enabling IoT. *Sensors*, 22(22), 8716.
- Sanusi, I. T., Agbo, F. J., Dada, O. A., Yunusa, A. A., Aruleba, K. D., Obaido, G., Olawumi, O., Oyelere, S. S., & Centre for Multidisciplinary Research and Innovation (CEMRI). (2024). Stakeholders' insights on artificial intelligence education: Perspectives of teachers, students, and policymakers. *Computers and Education Open*, 7, 100212.
- Sehito, N., Yang, S. Y., Larik, R. S. A., Kamal, M. M., Alwabli, A., & Ullah, I. (2025). Artificial intelligence (AI) and Internet of Things (IoT) applications in sustainable technology. In: El Hajjami, S., Kaushik, K., & Khan, I. U., eds. *Artificial general intelligence (AGI) security*. Singapore: Springer.
- Vijaya Kumari, V., Swarup, D., Bhuvaneshwari, V., Chandan, B. V., & Navya Madhuri, K. (2024). Gesture-driven smart homes: A survey of control systems and flexible sensors. *TIJER-International Research Journal*, 11(5), a231–a233.
- Wu, Y. Y. (2025). Integrating AI-IoT-OAHPs with existing elderly care systems. *Digital Health*, 11, 20552076251317378.

Zhai, C. P., Wibowo, S., & Li, L. D. (2024). The effects of over-reliance on AI dialogue systems on students' cognitive abilities: A systematic review. *Smart Learning Environments*, 11(1), 28.