2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51215

Cloud Computing for Industry 4.0 Education

Dr. Sheng-Jen Hsieh, Texas A&M University

Dr. Sheng-Jen ("Tony") Hsieh is a Professor in the Department of Engineering Technology and Industrial Distribution and a member of the Graduate Faculty at Texas A&M University, College Station, TX. His research interests include automation, robotics, cyber-manufacturing and Industry 4.0; optical/infrared imaging and instrumentation; micro/nano manufacturing; and design of technology for engineering education. He is also the Director of the Rockwell Automation Laboratory at Texas A&M University, a state-of-the-art facility for education and research in the areas of automation, robotics, and Industry 4.0 systems. He was named Honorary International Chair Professor for National Taipei University of Technology in Taipei, Taiwan, for 2015-21. Dr. Hsieh received his Ph.D. in Industrial Engineering from Texas Tech University, Lubbock, TX.

Cloud Computing (CC) for Industry 4.0 Education

Cloud computing (CC) is the delivery of computing resources—such as databases, software, and analytics—over the Internet. CC is the backbone of any Industry 4.0 system. It links materials, machines, and people; allows real-time visualization of a problem; and enables engineers to do real-time process improvements and maintenance. This work describes lesson materials, lab exercises, and a low-cost cloud computing platform developed to help Manufacturing Automation & Robotics students learn about cloud computing. The lab exercises involve creating a face recognition application on the cloud computing platform. Learning objectives include: 1) how to set up a cloud server; 2) how to utilize a machine learning application in the cloud; 3) how to set up a communication protocol and hardware devices; 4) how to interface a microcontroller, webcam, and other I/O devices needed for the application; and 5) how to write applications on the client and server sides. Survey results suggest students found the topic to be relevant and the lab exercises to be engaging. They requested additional exercises on advanced Industry 4.0 topics such as deep learning, machine vision, digital twins, and cyber-physical systems.

1 Introduction

Industry 4.0 technologies such as industrial networks, Internet of Things (IoT), and big data have enabled industry to collect, integrate, and analyze real-time data about materials, machines, and operations from manufacturing systems to enable smart manufacturing via continuous improvements in productivity and efficiency [1]. Industry 4.0 was projected to add \$2.2 trillion to domestic GDP by 2025. The estimated maximum value of the operational transformation brought by Industry 4.0 to the global manufacturing industry is \$3.7 trillion dollars per year [2]. Gázquez et.al. [3] surveyed more than 65 companies regarding their priority focus in developing new training programs in the next five years; 90% of respondents noted that areas such as Additive Manufacturing, Robotics, Ecodesign and Augmented Reality were priorities. Other areas considered important for training included Automation and Mechanization, **Cloud Computing,** ICT, ICT/Networking, AR, 3D printing, System Management and Technological Surveillance.

Incorporating these advanced topics into existing courses such as manufacturing automation and control courses can be challenging due to limited lecture, lab time, and pre-requisite knowledge of the new advanced topic. In addition, the cost to acquire 10 sets of lab exercise kits for 20 students can be challenging. Needed are low-cost (less than \$500) demonstration kits that utilize common controllers and sensors that students already know about, or that are easy to learn to use. For example, Siddiqi et al. presents a case study of low-cost gantry candy sorting system developed for Industry 4.0 technology education and outreach. Additive manufacturing was used to fabricate customized parts and components. A “You Only Look Once” (YOLO) machine learning model was built to recognize candy with different colors and shapes, and a low-cost Arduino controller was used to drive the system [4].

2 Role of Cloud Computing in Industry 4.0

As described in the 2013 report published by German National Academy of Science and Engineering, Industry 4.0 is the digital transformation of manufacturing and production [1]. A 2022 McKinsey report suggests three critical technologies: Industrial Internet of Things (IIoT), Cloud & Edge Computing, and Big Data and Analytics [5]. IIoT (e.g. internet ready sensors) can be placed on every machine and tool to collect real-time data. Cloud computing acts as the "central nervous system" that allows massive amounts of data from the factory floor to be stored, processed, real-time visualization and process improvements, and accessed globally. Cloud computing applications in manufacturing include 1) predictive maintenance – for example GM and Frito lay use this technology to reduce unplanned downtime; and 2) agile supply-chain and inventory management – for example, Tesla's Gigafactories use cloud-integrated systems to ensure parts arrive "just-in-time," reducing the need for expensive on-site warehouses [6].

3 Cloud Computing Curriculum for College Education

Adams et al. [7] summarize findings from a working group that surveyed, classified, and categorized existing cloud computing learning materials into exemplar modules for general course inclusion in response to industry needs for practical and hands on learning experience in Cloud Computing area. In addition, several major universities and industry—including CMU, MIT, Standford, AWS Academy, Google Cloud for Education, and Microsoft Azure—offer courses on Cloud Computing. However, these all involve semester-long and/or project-based courses. We have not found a low-cost hands-on DIY module that utilizes microcontrollers, Wi-Fi communications, web services, and machine learning for practical application such as Facial Recognition.

4 Educational Approach

To fit into the scope of a current course on Manufacturing Automation and Robotics, we have designed and developed a lab exercise to 1) introduce cloud computing concepts; and 2) provide hands-on practice with cloud computing through the experience of building a small scale cyber-physical system for facial recognition using readily available off-the-shelf technologies such as Arducam, Arduino, ESP32 microcontrollers, and HiveMQ web server.

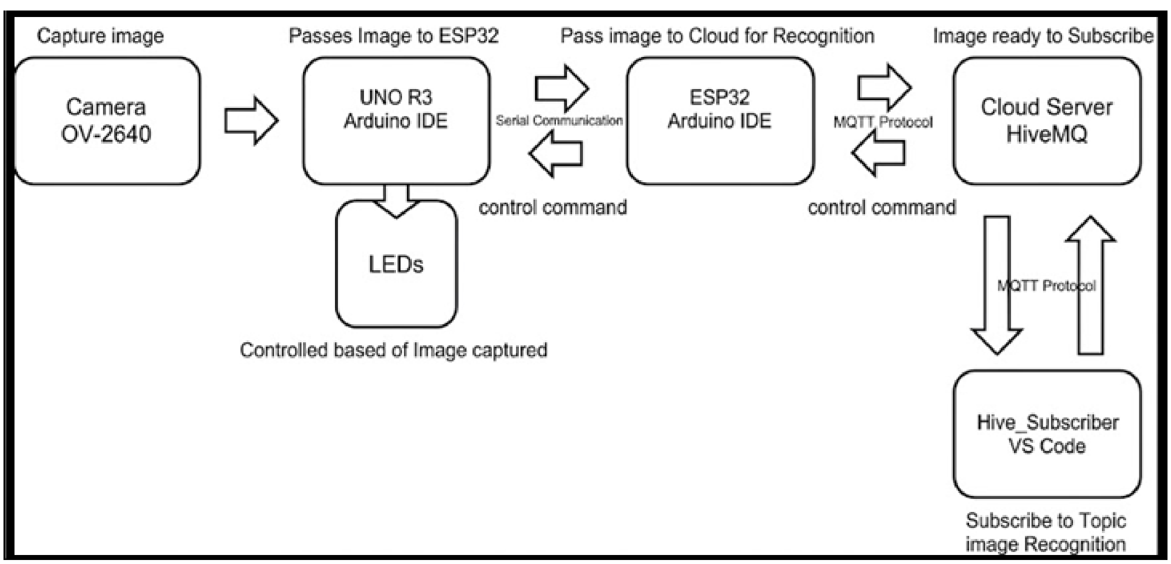

Figure 1. Cloud Computing Pipeline System.

Learners practice 1) using serial and wireless communication protocols, and 2) cloud-based data handling via the practical application of edge-to-cloud data flow in IoT systems. The lab exercise allows users to see an image captured by Arducam being sent out from Arduino UNO R3 via

ESP32 to HiveMQ web server for analysis. Later a detection signal will be sent back to turn LEDs on or off. To provide everyone with hands on experience with limited resources from university, we decided to adopt a low-cost cloud computing (CC) platform integrating existing available technologies. Figure 1 shows the CC platform.

4.1 Low-Cost Cloud Computing Platform

The low-cost cloud computing platform follows a Cyber-Physical System (CPS) architecture. This setup represents a "Miniature Industry 4.0" model, demonstrating how edge devices connect to high-level cloud analytics. This system utilizes a tiered architecture where data is captured at the Physical Layer, processed at the Edge Layer, and stored or visualized at the Cloud Layer. Figure 2 shows that low-cost cloud computing prototype that integrates all the hardware components needed for cloud-based facial recognition application development. The Arduino is connected to a Arducam and interfaces with the I/O devices (LEDs) via a second breadboard.

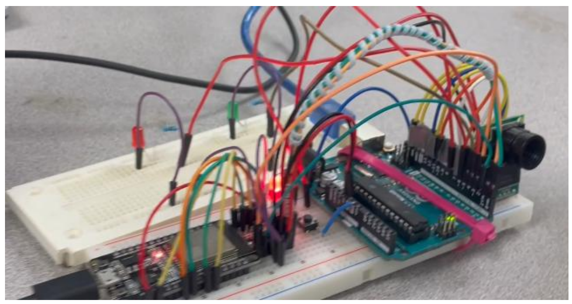

Figure 2. Low-Cost Arduino Platform.

The development is managed via a Unified Software Layer.

Physical & Edge Sensing Layer (Arducam camera and ESP32 module). The process begins with the Arducam camera (OV2640 model or similar). This sensor acts as the "eyes" of the system. Unlike a standard webcam, this camera interfaces directly with the Arduino UNO R3, a high-performance microcontroller with integrated Wi-Fi and Bluetooth. Figure 3 shows the ESP32 module, ArduCAM , and Arduino UNO R3 microcontroller.

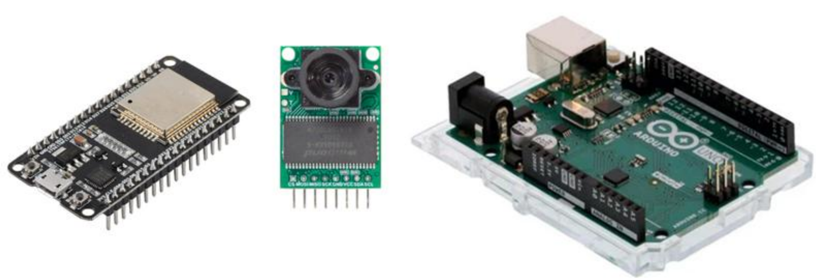

Figure 3. ESP32 module, Arducam, and Arduino UNO Microcontroller.

- Data capture: The Arducam captures raw light data and converts it into digital frames (JPEG or BMP).
- Edge processing: The ESP32 does not just pass data along; it performs "Edge Computing." It can execute light AI models for face detection or color recognition. By

processing data locally, the system reduces latency (the time it takes for a system to react).

- Wireless gateway: The ESP32 acts as a web server, hosting a local IP address where a live video stream can be viewed within a local area network (LAN).

Hardware Control Layer (Arduino Uno R3). While the ESP32 handles high-speed data and Wi-Fi, the Arduino Uno R3 serves as the "muscle" of the operation. In many Industry 4.0 applications, one chip handles the "thinking" (ESP32) while another handles the "doing" (Arduino).

- Serial Communication (UART): The ESP32 and Arduino are linked via a serial bridge. When the camera "sees" a specific object, the ESP32 sends a tiny string of data (a trigger) to the Arduino.
- Actuation: The Arduino Uno R3 receives this trigger and controls physical components. This separation of duties ensures that a heavy Wi-Fi upload doesn't cause a delay in a critical mechanical movement.

Cloud & Monitoring Layer (Cloud Server). The Cloud Server represents the "brain" and "memory" of the system. While the ESP32 has very limited storage, the Cloud Server offers infinite space and higher processing power.

- Information Flow: Using the MQTT or HTTP POST protocol, the ESP32 sends summarized data (such as "Object Detected at 10:00 AM") or compressed images to the cloud.
- Data Persistence: The Cloud Server hosts a database (such as Firebase or AWS) that keeps a historical log of every event. This allows managers to look back at data from a month ago to identify patterns—a core concept of Big Data Analytics.
- Global Accessibility: Because the data is in the cloud, it is no longer restricted to the local factory Wi-Fi; it can be monitored from a smartphone anywhere in the world.

The Development & Logic Layer (VS Code). VS Code is the command center for engineers. It is the environment where the "Digital" meets the "Physical."

- Firmware Development: C++ code is used to tell the ESP32 how to connect to Wi-Fi and tell the Arduino what action to take.
- Debugging: Through the Serial Monitor in VS Code, the developer receives real-time feedback from the hardware, such as signal strength (RSSI) or error codes, allowing for rapid troubleshooting.

Facial Recognition Application Development. YOLO-Face is a specialized version of the "You Only Look Once" (YOLO) object detection framework, specifically optimized for detecting human faces in images and video. While standard YOLO models are designed to detect a wide variety of objects (such as cars, birds, or chairs), YOLO-Face models are tuned to handle the unique challenges of facial detection, such as small scales, varying orientations, and heavy occlusions (such as masks, sunglasses, or hands) [8-9].

YOLOv8n-Face (the Nano version) was applied for this application because it is anchor-free and handles different head tilts well, resulting in a low false Acceptance Rate (FAR). In addition, the nano version provides a smoother inference speed on CPUs, making it ideal for low-power processors like Raspberry Pi. The application development process included the following steps.

- Step 1: Data collection and annotation. A high-quality dataset is needed to teach the model.
- Step 2. Training the network. Instead of training from scratch, Transfer Learning is used to take the "intelligence" of a pre-trained YOLO8n model and fine-tune its final layers to recognize specific features.
- Step 3: Validating and testing the model. Validation tells you how "trustworthy" a model is before deploying it to a real-world camera. Figure 4 shows a red LED comes on when when the camera does not recognize a human face.

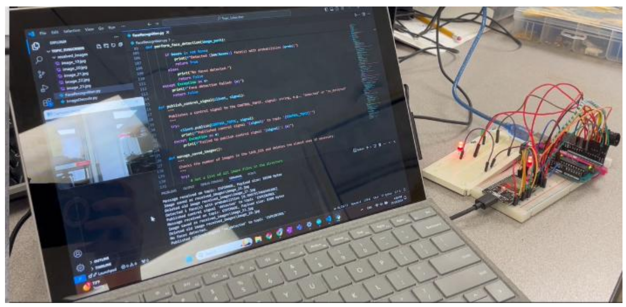

Figure 4. Red LED comes on to indicate No Recognized Face Detected

Overall, the flow of information is as follows: 1) Optical to Digital: Light hits the Arducam and is converted to pixels; (2) Local Intelligence: ESP32 analyzes the pixels and decides if an action is needed; (3) Physical Trigger: ESP32 sends a command to Arduino; Arduino turns on an LED; and (4) Remote Sync: ESP32 uploads a status report to the Cloud Server that is saved to a database.

4.2 Lab exercise

The learning objectives of the lab exercise include 1) interface microcontrollers, webcams, and I/O devices, (2) set up communication protocols and hardware devices, 3) set up a cloud server, 4) utilize machine learning in the cloud, and 5) write applications for client and server sides. The lab exercise consists of the following tasks and deliverables.

<u>Task #0 Preparation:</u>

a) Download Python face recognition code.
b) Install Python (latest, VS Code recommended).
c) Configure MQTT credentials/topics identical to ESP32 setup.
d) Install required libraries: paho-mqtt, Pillow, torch, torchvision, opencv-python, numpy, facenet-pytorch.

<u>Task #1 Circuit Setup. Primary steps include</u>

a) Use level shifter for signal compatibility (Uno = 5v, ESP32 = 3.3v).
b) TX/RX/power/GND connections via level shifter.
c) Connect LEDs and switch to specified Arduino pins.
d) Reserve camera pins.

Since the Arduino UNO R3 outputs a 5v signal while ESP32 GPIO Pins works at 3.3v, it is critical to include a level shifter in our circuit. We will connect all TX/RX pins of UNO R3 to the HV side, including a 5V power supplies and GND. On the other hand, we will connect all TX/RX, 3.3v, and GND pins to the LV side. Figure 5 shows the Shifter board layout.

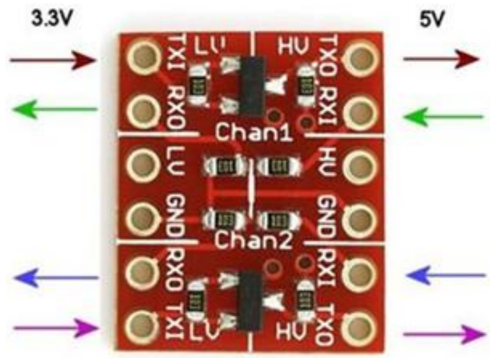

Figure 5. Shifter Board Layout.

All 4 channels need to be set up. The wire connections are as follows:

Arduino Pin	**Connection / Component**	**Target (ESP32 or Peripheral)**	**Purpose / Notes**
0 (RX)	Level Shifter	ESP32 TX2	Hardware Serial (Cross-connected)
1 (TX)	Level Shifter	ESP32 RX2	Hardware Serial (Cross-connected)
2	Level Shifter	ESP32 GPIO (RX Config)	Secondary Serial / Digital Link
3	Level Shifter	ESP32 GPIO (TX Config)	Secondary Serial / Digital Link
4	150Ω Resistor	Red LED	Visual Alert / Status
5	150Ω Resistor	Green LED	Visual Alert / Status
6	Direct Connection	Switch / Push Button	User Input
7	Reserved	Camera Module	Do Not Use for other logic
8	Direct Connection	LED (Optional)	ON/OFF Status Indicator
10	Reserved	Camera Module	Do Not Use for other logic
11	Reserved	Camera Module	Do Not Use for other logic
12	Reserved	Camera Module	Do Not Use for other logic
13	Reserved	Camera Module	Do Not Use for other logic
SDA	Reserved	Camera Module	I2C Data Line
SCL	Reserved	Camera Module	I2C Clock Line
5V	HV (High Voltage)	3.3V (LV)	Powers the shifter's reference rails.
GND	GND	GND	Establish a common ground for all logic.

Task #2 Arducam Setup:

a) Copy library files to Arduino IDE.
b) Enable OV2640_MINI_2MP_Plus in memorysaver.h, disable others.
c) Select correct serial port and camera example in Arduino IDE.
d) Test camera setup using ArduCAM_Host_V2 application.

Figure 6 shows the schematic of Arducam with Arduino and a screenshot of successful configuration of Arducam with Arduino.

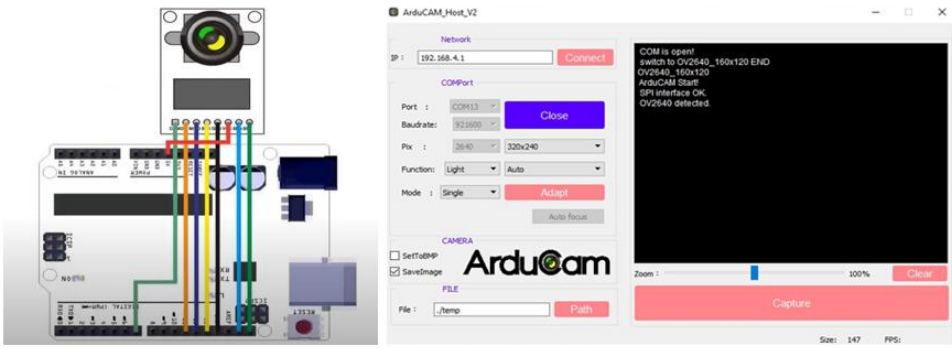

Figure 6. Schematic and Configuration of Arducam with Arduino.

Task #3 Serial Communication Test Run:

a) Use provided Arduino and ESP32 test code to verify TX/RX.
b) Ensure bidirectional message passing between boards.

Task #4 Arduino UNO R3 Setup:

a) Capture image, sent to ESP32.
b) Receive LED control commands from ESP32.
c) Set up and upload provided code; choose camera resolution (<640x480 recommended for MQTT).

Task #5 ESP32 Setup:

a) Receive image from UNO R3; publish to HiveMQ.
b) Receive cloud control command, pass to UNO R3.
c) Download and configure provided code: Wi-Fi, MQTT credentials, root certificate, topic names.

Task 6 HiveMQ Client Setup:

a) Set up MQTT client cluster.
b) Create username/password for broker.
c) Enter credentials in Arduino IDE/ESP32.
d) Subscribe to topics for image and control data.

Task #7 Cloud Connection Test Run:

a) Upload cloud connection test code to ESP32.

b) Test by publishing/receiving string messages before attempting image data.

Task #8 Compiler (Local Server) Setup:

a) HiveMQ is free, easy-access platform for IoT starter projects. However, it is lightweight.

5 Evaluation and Conclusion

A workshop was hosted on Texas A&M University campus in April 2025 to gauge the acceptance of materials by college instructors who teach related areas. Four instructors participated in the designed lab exercises after a presentation about the Cloud Computing module. A survey was administered; instructors were asked to rate the lab on each of the following attributes on a scale of 1–5 (1: Strongly Disagree, 5: Strongly Agree). The average ratings for the attributes were: (1) *learning objectives* – 4.5, (2) *lab instructions* – 4.0, and (3) *allotted time frame* – 2.5. Only two of the four instructors completed the lab within the 3-hour timeframe. The feedback suggested that the length of the lab should be extended, since there are eight tasks that need to be completed in sequence. If any task is not completed correctly, the system will not work. However, all rated the learning objectives highly. Overall, of the four instructors, three recommended that the lab exercise was ready to be used and one recommended minor revisions.

Future work includes 1) breaking the lab exercise into Part A and B (two lab exercises lasting two hours each); and 2) pilot testing the exercises with a few upper-level undergraduate students. Since eight tasks need to be completed in sequence to see the system working, dividing the lab over two sessions may increase the likelihood of the entire lab being completed correctly. Dividing the lab into two lab periods over two weeks will also provide the instructor with more time to introduce the lab assignment and for students to reflect on what they have learned after the first lab so they can make improvements during the second lab. Lastly, an on-line video about the lab and each task can also be developed as a resource that students can use during the lab time.

Acknowledgement

This material was supported by the National Science Foundation's Advanced Technology Education Program (Award no. 2202201). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group," acatech – National Academy of Science and Engineering, Apr. 2013. [Online]. Available: https://www.acatech.de/wp-content/uploads/2018/03/Final_report__Industrie_4.0_accessible.pdf

[2] S. Hsieh, M. Barger, S. G. Marzano, and J. Song, "Preparing the manufacturing workforce for Industry 4.0 technology implementation," in Proc. 2023 ASEE Annu. Conf. & Expo., Baltimore, MD, June 2023. doi: 10.18260/1-2—43957.

[3] J. L. Romero Gázquez, M. V. Bueno Delgado, J. J. Ortega Gras, J. Garrido Lova, M. V. Gómez Gómez, and M. Zbiec, "Lack of skills, knowledge and competences in Higher Education about Industry 4.0 in the manufacturing sector," RIED. Rev. Iberoam. de Educ. a Distancia, vol. 24, no. 1, 2021.

[4] J. Siddiqi, A. S. Gandy, and S. Hsieh, "Technical Training for Industry 4.0 Technologies: Low-Cost Gantry Candy Sorting System for Education and Outreach," in Proc. ASEE 2024 Annu. Conf., Portland, OR, June 23–26, 2024.

[5] "What are Industry 4.0, the Fourth Industrial Revolution, and 4IR?," McKinsey & Company, Aug. 17, 2022. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir

[6] "The technology behind General Motors' MES 4.0," GE Vernova, Aug. 5, 2025. [Online]. Available: https://www.gevernova.com/software/blog/the-technology-behind-general-motors-mes-40

[7] J. Adams et al., "Cloud Computing Curriculum: Developing Exemplar Modules for General Course Inclusion," in ITiCSE-WGR '20: Proc. Working Group Rep. Innovation and Technol. in Comput. Sci. Education, 2020, pp. 151–172. doi: 10.1145/3437800.3439206.

[8] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Y. Wang, "YOLO-FaceV2: A Scale and Occlusion Aware Face Detector," Pattern Recognit., vol. 122, 2022.

[9] "Ultralytics YOLOv8: Real-Time Face Detection Implementation," Ultralytics, 2023. [Online]. Available: docs.ultralytics.com