

seeed studio

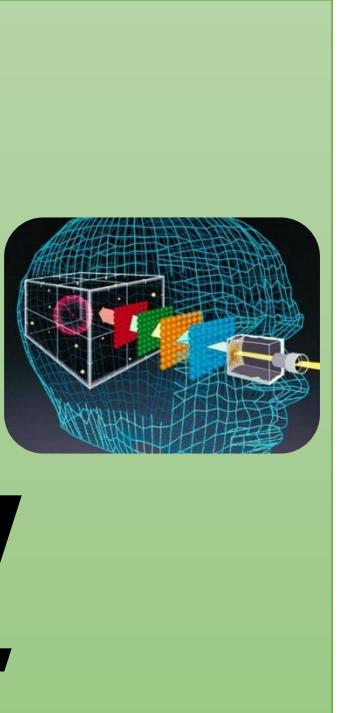
Hands-On Computer Vision

ICTP-UNU Workshop on TinyML
for Sustainable Development



Main Content

1



Introduction to
Machine Vision

2



Arduino
Environment
Preparation

3



Equipment
Introduction &
Demonstration

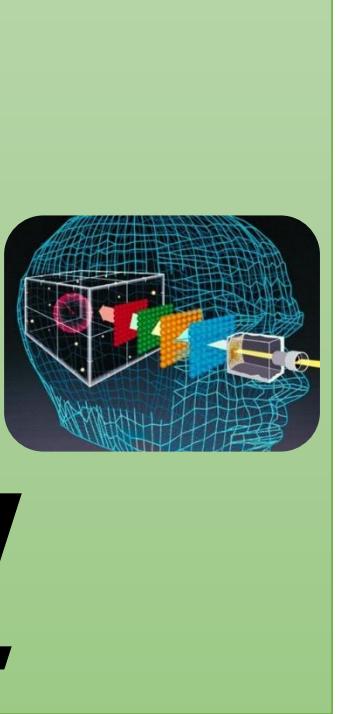
4



MCU + AI Sensors =
Unlimited Creative
Possibilities

Main Content

1

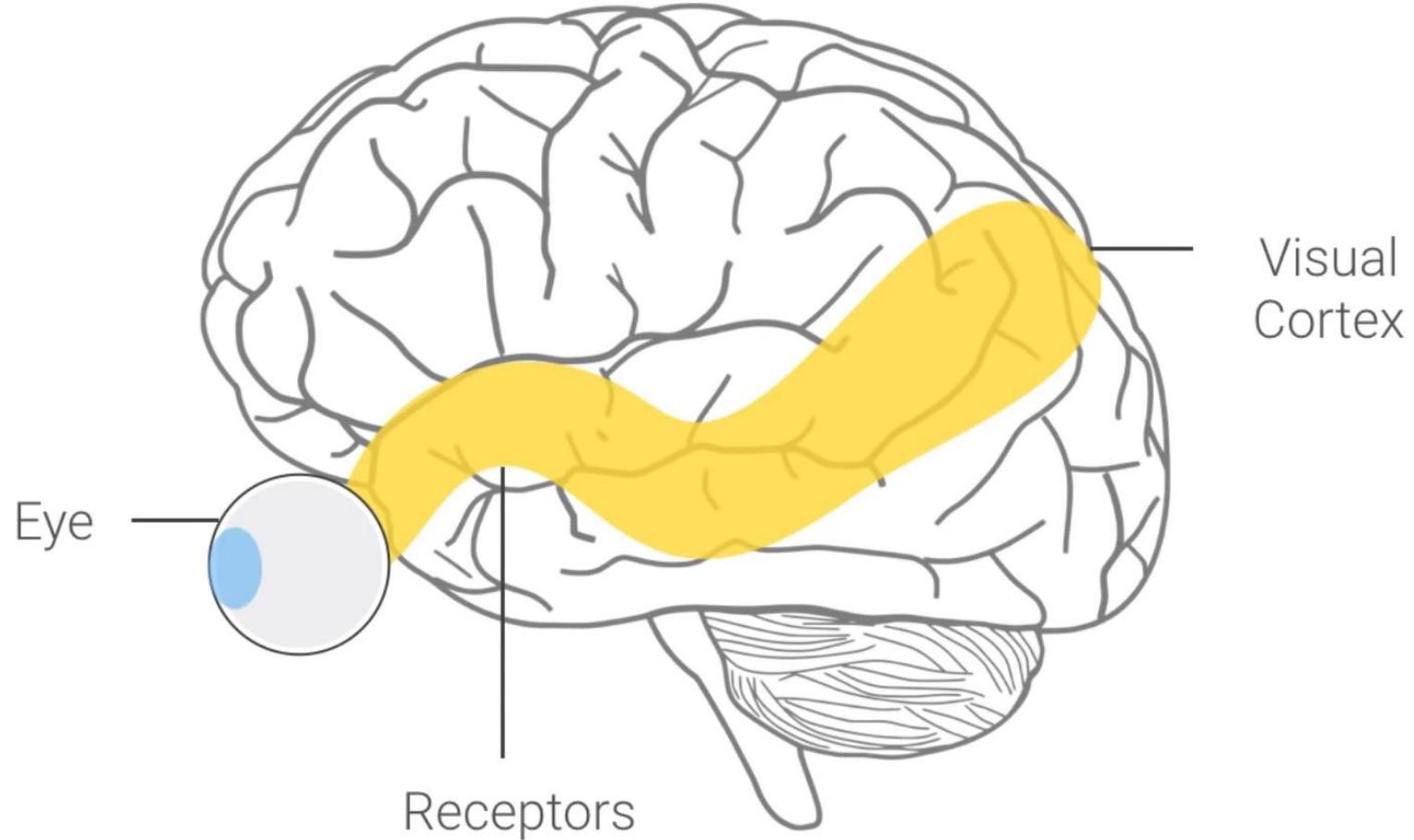


Introduction to
Machine Vision

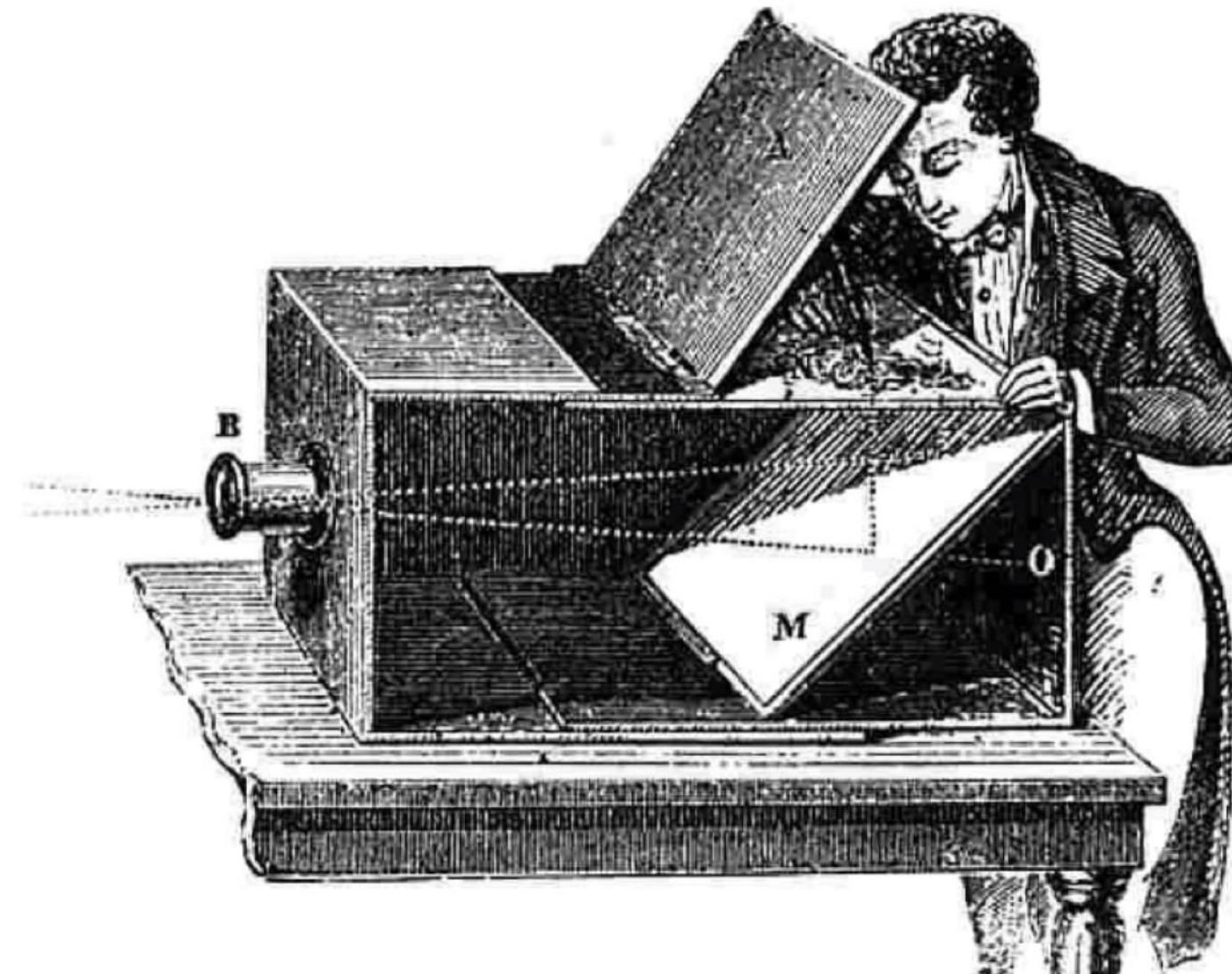
Introduction to Machine Vision



Introduction to Machine Vision



Introduction to Machine Vision

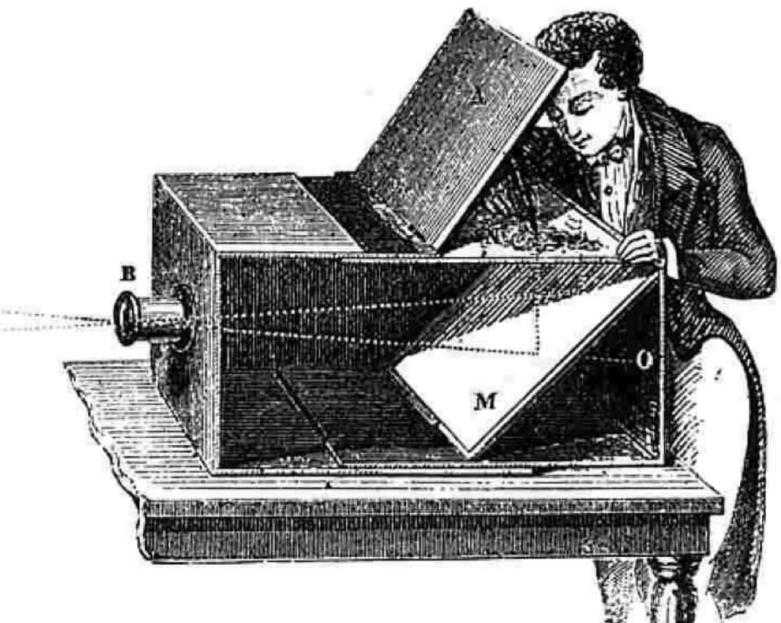
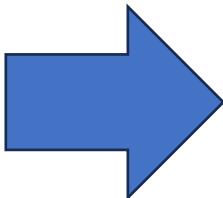
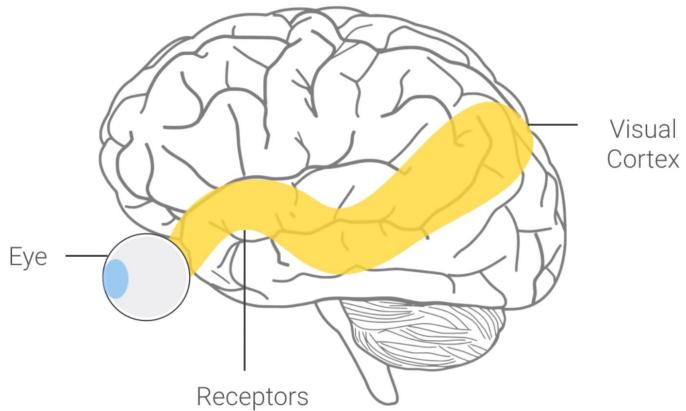


Introduction to Machine Vision

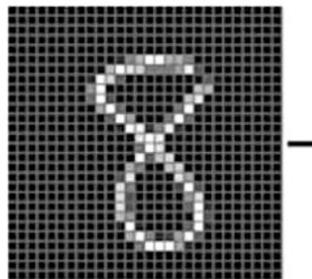


seeed studio

Introduction to Machine Vision



Introduction to Machine Vision

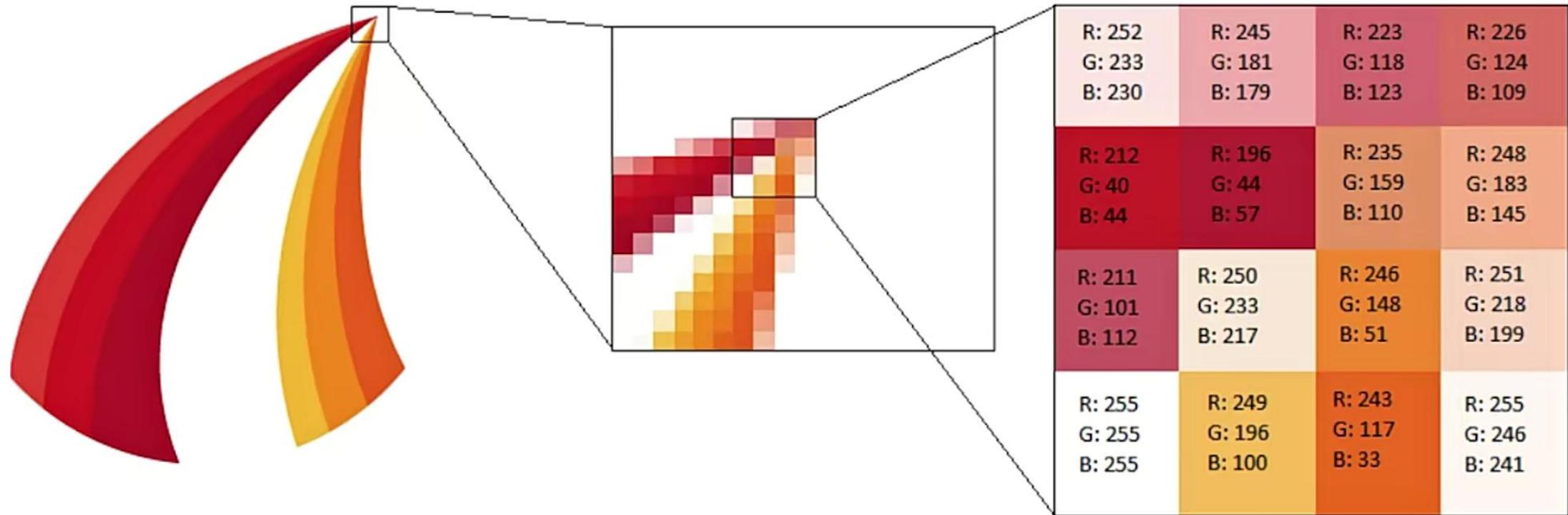


28 x 28
784 pixels

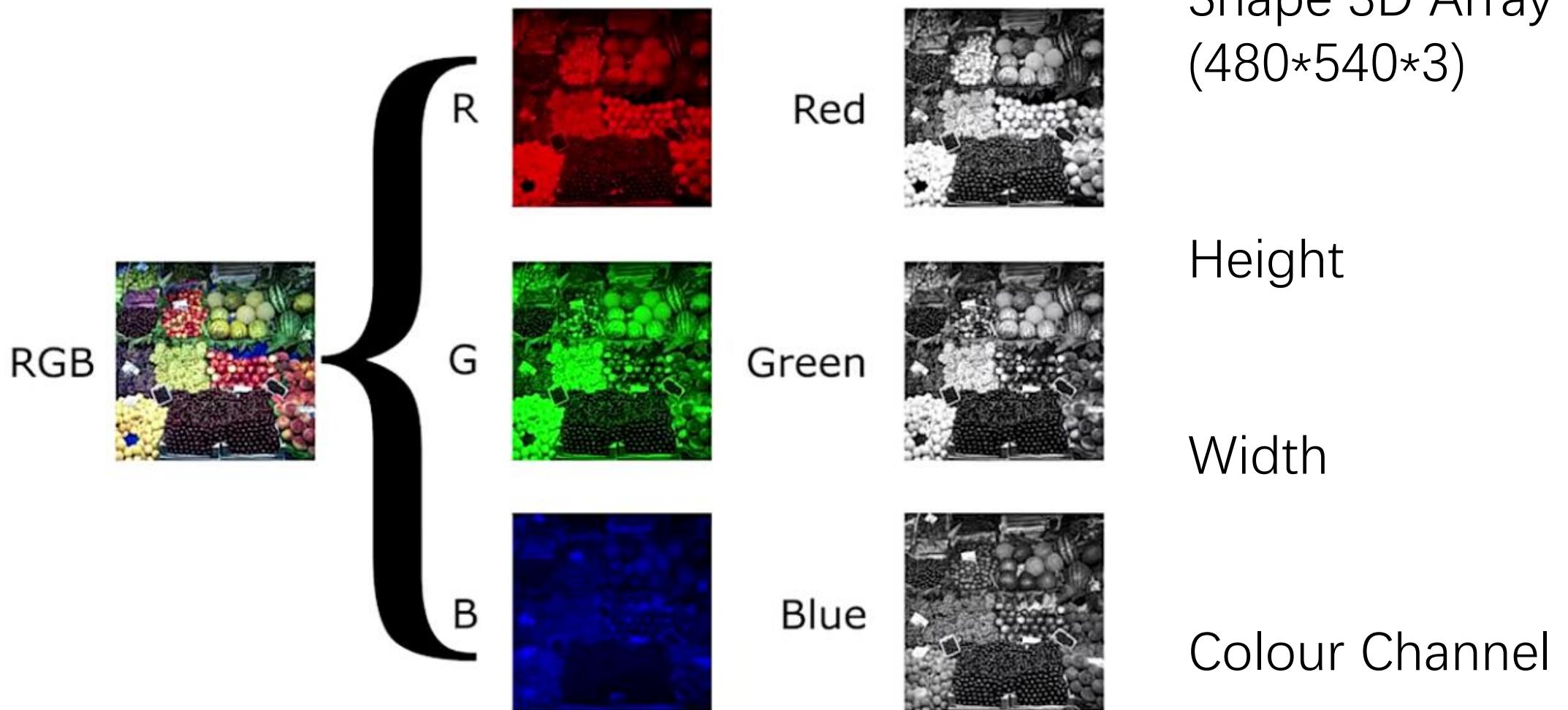
→

A greyscale image can be understood as consisting of dark (0) and light (numbers greater than zero), the closer to 255 means the brighter the area is.

Introduction to Machine Vision



Introduction to Machine Vision



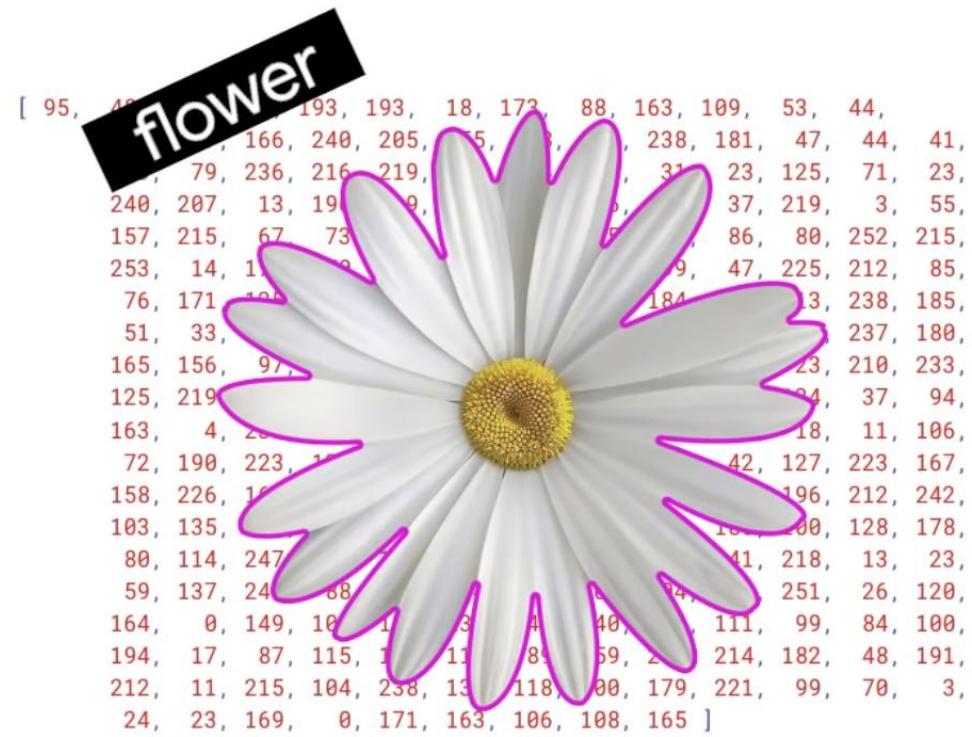
Introduction to Machine Vision



[95, 48, 122, 153, 193, 193, 18, 173, 88, 163, 109, 53, 44, 106, 87, 166, 240, 205, 155, 88, 201, 238, 181, 47, 44, 41, 169, 79, 236, 216, 219, 15, 14, 7, 31, 23, 125, 71, 23, 240, 207, 13, 196, 9, 113, 3, 126, 186, 37, 219, 3, 55, 157, 215, 67, 73, 64, 38, 218, 135, 52, 86, 80, 252, 215, 253, 14, 174, 70, 225, 225, 109, 156, 99, 47, 225, 212, 85, 76, 171, 125, 23, 60, 84, 20, 109, 184, 49, 13, 238, 185, 51, 33, 250, 237, 74, 174, 53, 241, 94, 18, 61, 237, 180, 165, 156, 97, 42, 232, 255, 189, 241, 203, 34, 23, 210, 233, 125, 219, 132, 231, 145, 64, 203, 135, 3, 87, 124, 37, 94, 163, 4, 236, 50, 233, 244, 176, 66, 139, 164, 18, 11, 106, 72, 190, 223, 133, 78, 230, 24, 235, 20, 42, 127, 223, 167, 158, 226, 163, 114, 121, 1, 194, 245, 215, 199, 196, 212, 242, 103, 135, 194, 45, 192, 4, 6, 24, 12, 185, 100, 128, 178, 80, 114, 247, 18, 149, 215, 141, 128, 148, 241, 218, 13, 23, 59, 137, 244, 88, 200, 174, 39, 203, 104, 40, 251, 26, 120, 164, 0, 149, 103, 138, 239, 243, 40, 131, 111, 99, 84, 100, 194, 17, 87, 115, 178, 110, 89, 59, 255, 214, 182, 48, 191, 212, 11, 215, 104, 238, 132, 118, 100, 179, 221, 99, 70, 3, 24, 23, 169, 0, 171, 163, 106, 108, 165]

Introduction to Machine Vision

Machine Learning

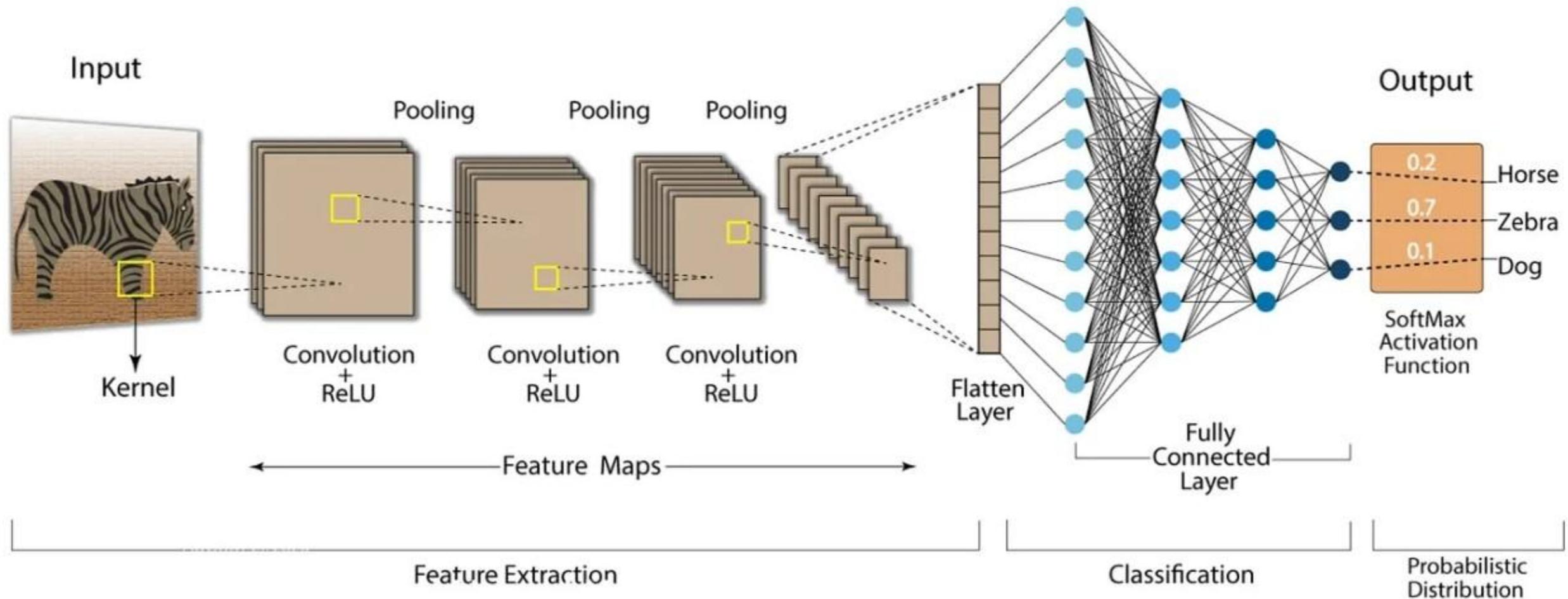


Introduction to Machine Vision

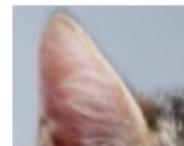
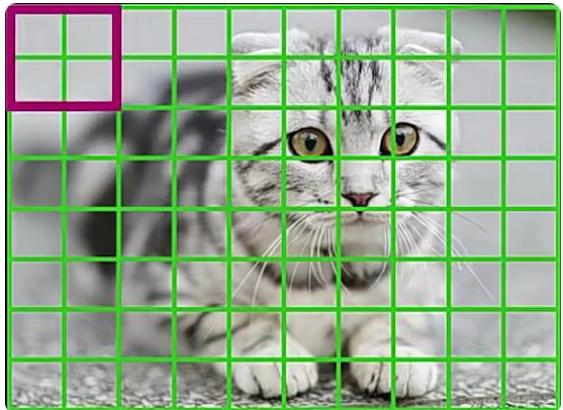


Introduction to Machine Vision

Convolution Neural Network (CNN)



Introduction to Machine Vision



Ear



Ear



Eye



Eye



Fur

Convolutional Layers

- Convolutional layers use a set of learnable filters (also known as convolutional kernels) that slide over the input image and perform convolution operations.
- Each filter extracts specific features from the image, such as edges, textures, or shapes.
- The result of the convolution operation is a set of feature maps that represent the response of different features at various locations in the image.
- By using multiple filters, convolutional layers can extract a variety of features from the image.

Introduction to Machine Vision



1000 x 750 x 3

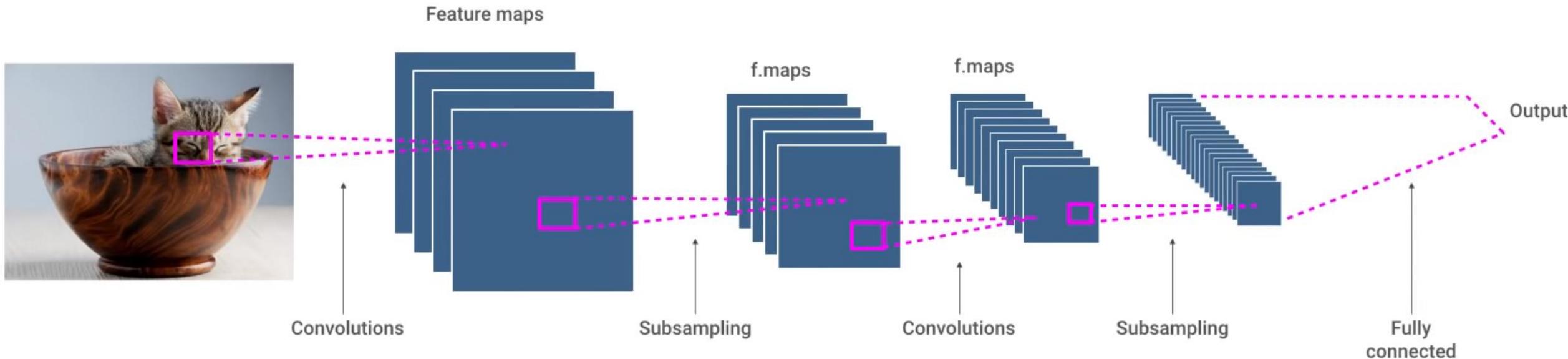


1024 x 1

Pooling Layers

- Pooling layers are used to reduce the spatial dimensions of the feature maps while retaining the most important feature information.
- The most common pooling operation is max pooling, which selects the maximum value within each region of the feature map as the representative of that region.
- Pooling operations help to reduce the size of the feature maps, thereby reducing the computational burden of subsequent layers and providing some level of translation invariance.

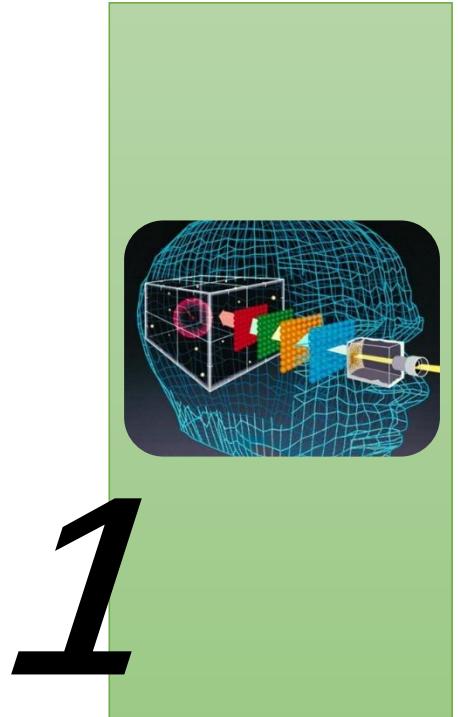
Introduction to Machine Vision



Fully Connected Layers

- After multiple convolutional and pooling layers, CNNs typically employ one or more fully connected layers for the final classification or prediction task.
- Fully connected layers flatten the features extracted by the previous layers and apply a weight matrix to transform them into the final output.
- Fully connected layers can learn complex relationships between features and generate the desired output based on the task requirements, such as the probability of an image belonging to a specific class.

Main Content



1

Introduction to
Machine Vision



2

Arduino
Environment
Preparation

Arduino Environment Preparation

TABLE OF CONTENTS

[What is Arduino?](#)

[Our Mission & Vision](#)

[Maker](#)

[Edu](#)

[Pro](#)

[The Arduino Team](#)

[Founders](#)

[Management team](#)

[Tech Partnerships](#)

[Our Distributors](#)

About Arduino

What is Arduino?

Arduino designs, manufactures, and supports electronic devices and software, allowing people around the world to easily access advanced technologies that interact with the physical world. Our products are straightforward, simple, and powerful, ready to satisfy users' needs from students to makers and all the way to professional developers.

[Find out more](#)

Our Mission & Vision

Arduino's mission is to enable anyone to enhance their lives through accessible electronics and digital technologies. There was once a barrier between the electronics, design, and programming world and the rest of the world. Arduino has broken down that barrier.

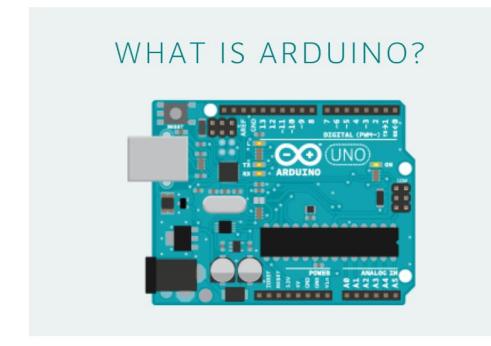
Over the years, our products have been the brains behind thousands of projects, from everyday objects to complex scientific instruments. A worldwide community, comprising students, hobbyists, artists, programmers, and professionals, has gathered around this open-source platform, their contributions adding up to an incredible amount of accessible knowledge.

Our vision is to make Arduino available to everyone, whether you are a student, maker or professional, which is why we now have three segments to our business. These segments work together as an ecosystem with a shared mindset: we started with Maker, and that has evolved into Education and PRO solutions.

Arduino Environment Preparation

<https://www.arduino.cc/>

The screenshot shows the Arduino website's header with navigation links: PROFESSIONAL, EDUCATION, STORE, HARDWARE, SOFTWARE (highlighted with a red box), CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. There is also a search bar and a sign-in link.



BUY AN ARDUINO



LEARN ARDUINO



DONATE



ARDUINO IN THE CLOUD



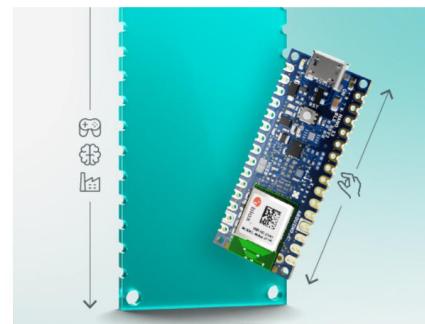
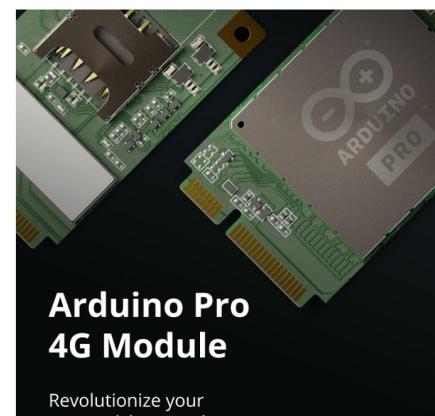
CAREERS



正在等待 sgtm.arduino.cc...



Discover Now



Nano 33 BLE Rev2

Effortless access to essential features,
coupled with an integrated IMU for
expanded experimentation.

[Check it out now!](#)



4 NEW IOT MONITORING
DASHBOARD UPDATES ON

Help

seeed studio

Arduino Environment Preparation

Legacy IDE (1.8.X)



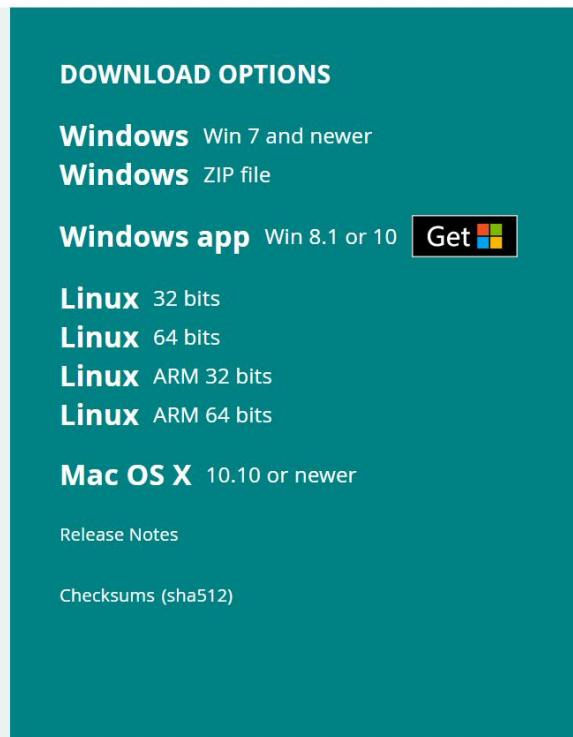
Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Arduino IDE 1.x documentation](#) for installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.



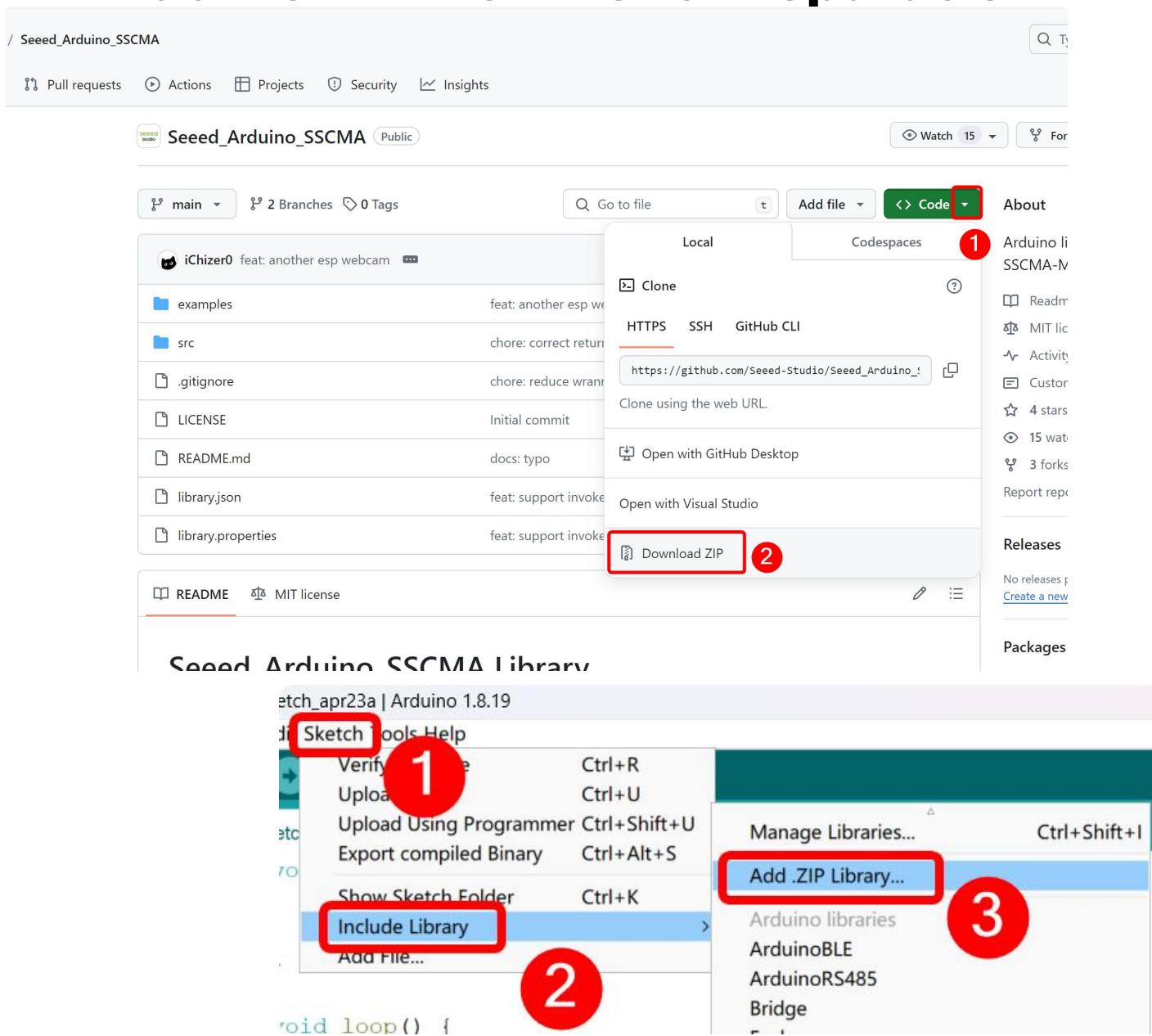
A screenshot of the Arduino IDE download options page. The page has a teal header with the text "DOWNLOAD OPTIONS". Below the header, there are sections for "Windows", "Linux", and "Mac OS X", each listing multiple download links. A "Get" button is visible next to the Windows app link. At the bottom of the page, there are links for "Release Notes" and "Checksums (sha512)".

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 [Get](#)
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

[Release Notes](#)
[Checksums \(sha512\)](#)

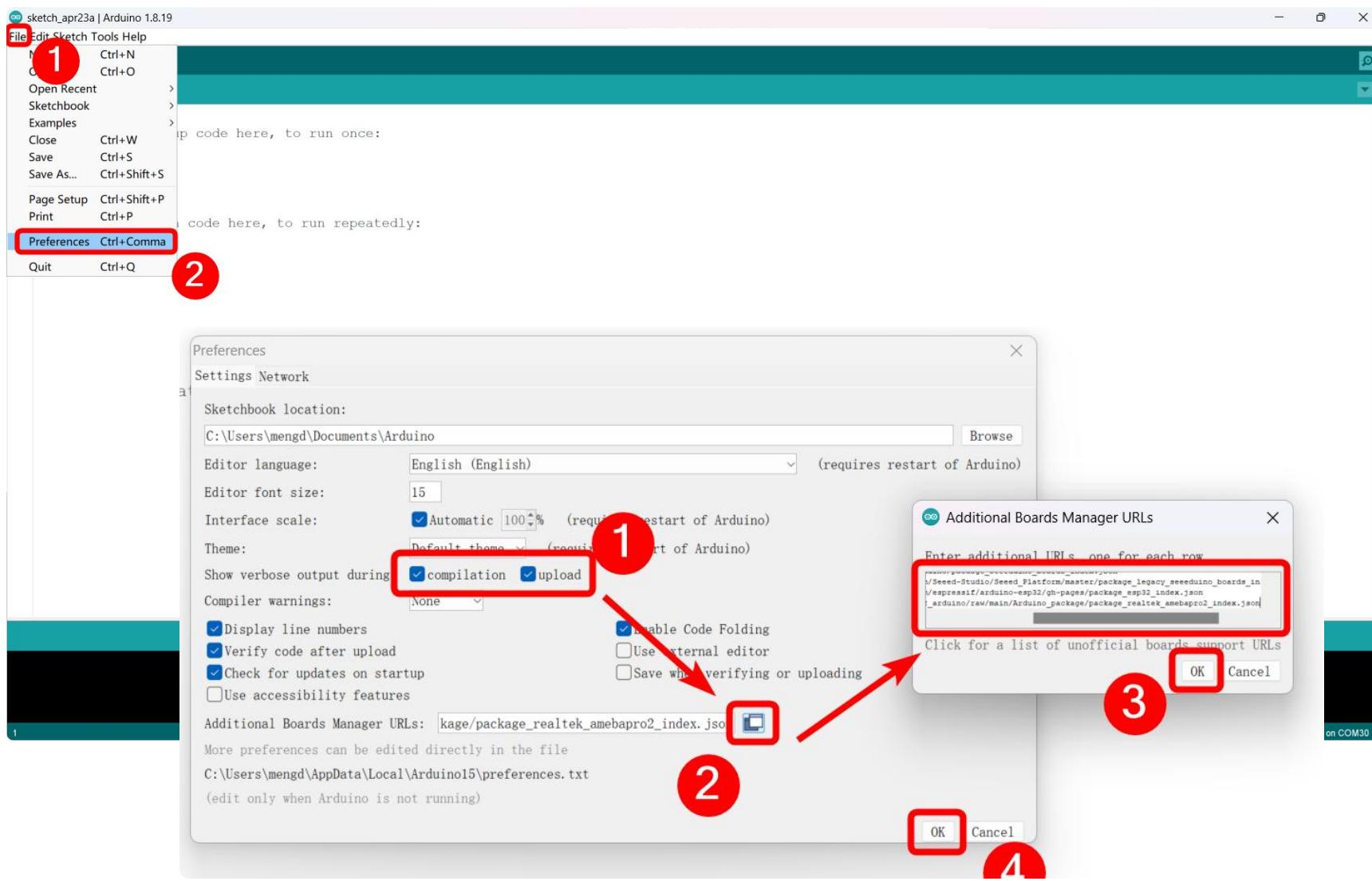
1. Visit the official Arduino website: <https://www.arduino.cc/en/software>
2. Click on the "Windows" or "Mac" button based on your operating system.
3. Download the Arduino IDE 1.8.19 installer.
4. Once the download is complete, run the installer.
5. Follow the installation wizard, accepting the license agreement and choosing the installation directory.
6. If prompted, allow the installer to install device drivers.
7. Once the installation is finished, click "Close" to exit the installer.
8. Open the Arduino IDE from the desktop shortcut or the start menu.
9. You're now ready to start using Arduino IDE 1.8.19!

Arduino Environment Preparation



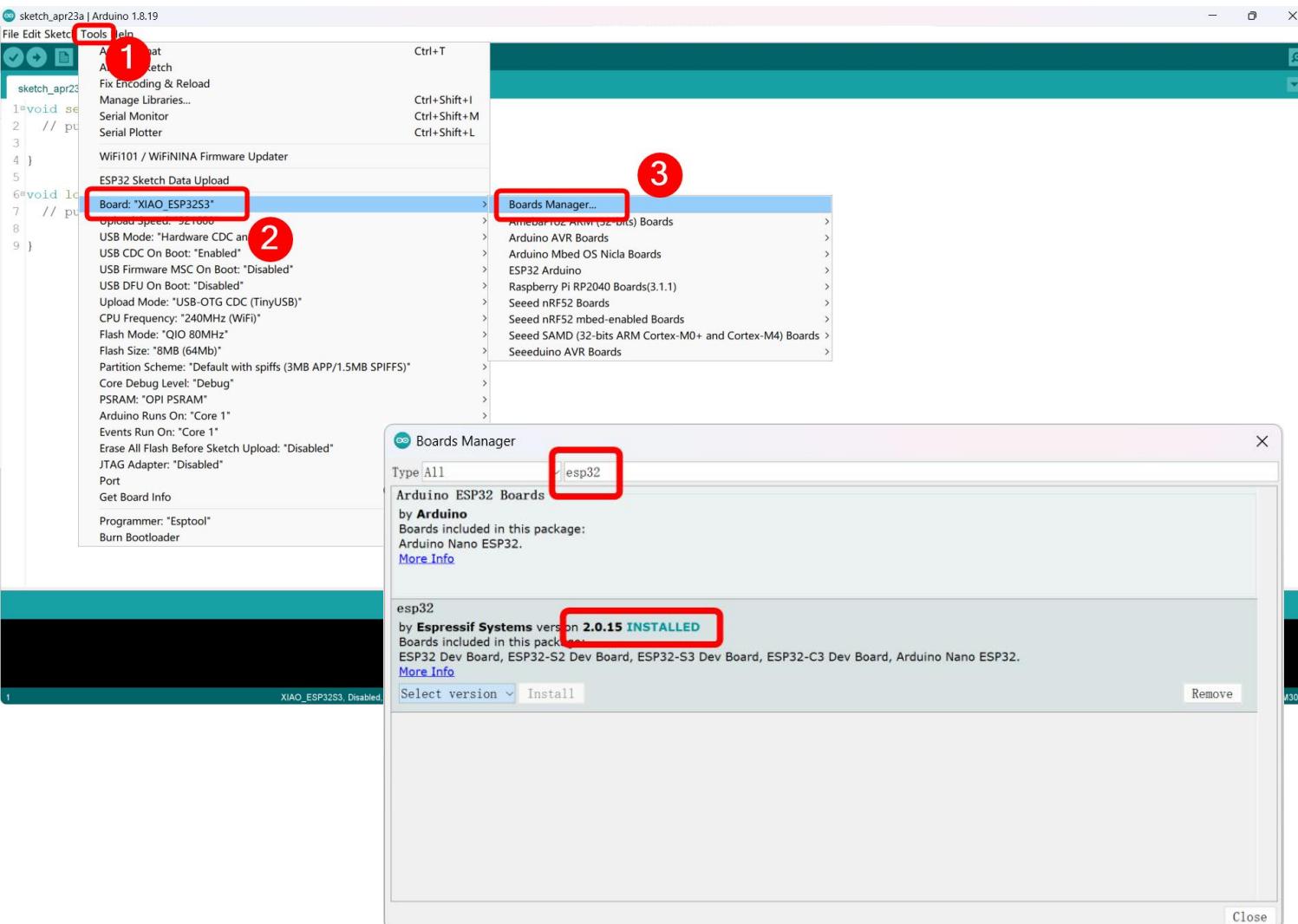
1. Open your web browser and navigate to the GitHub repository:
https://github.com/Seeed-Studio/Seeed_Arduino_SSCMA
2. Click on the green "Code" button and select "Download ZIP" to download the library as a ZIP file.
3. Save the ZIP file to a location on your computer where you can easily find it.
4. Open the Arduino IDE.
5. Go to Sketch > Include Library > Add .ZIP Library.
6. In the file browser window that appears, navigate to the location where you saved the downloaded ZIP file.
7. Select the ZIP file and click "Open" to add the library to your Arduino IDE.
8. The Seeed_Arduino_SSCMA library should now be installed and ready to use.
9. To verify the installation, go to Sketch > Include Library and check if "Seeed_Arduino_SSCMA" appears in the list of installed libraries.

Arduino Environment Preparation



1. Open the Arduino IDE.
2. Go to File > Preferences.
3. In the "Additional Boards Manager URLs" field, enter the following URL:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
4. Click "OK" to close the Preferences window.

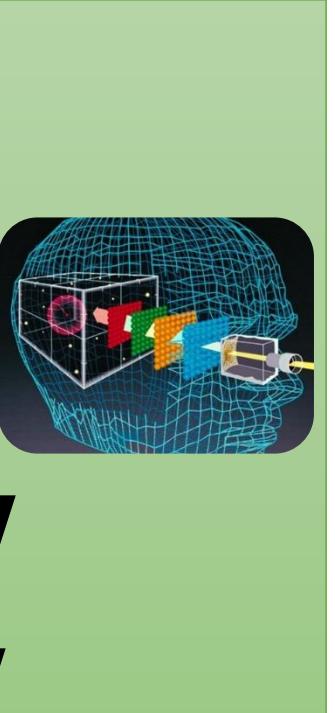
Arduino Environment Preparation



5. Navigate to Tools > Board > Boards Manager.
6. In the Boards Manager window, search for "ESP32".
7. Locate the "ESP32 by Espressif Systems" entry and click on it.
8. Select the latest version from the dropdown menu and click "Install".
9. Wait for the installation process to complete. This may take a few minutes.
10. Once the installation is finished, close the Boards Manager window.

Main Content

1



Introduction to
Machine Vision

2



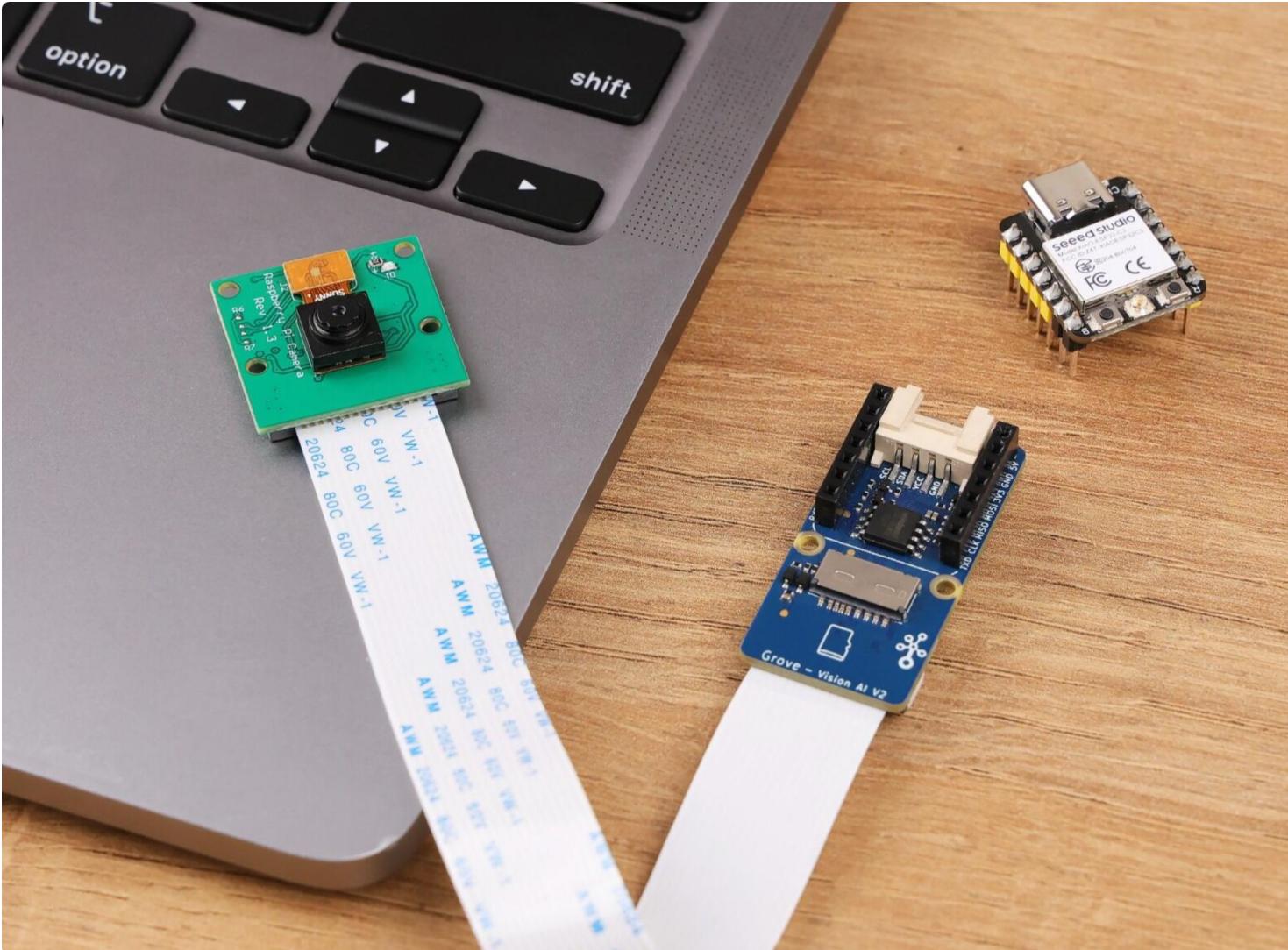
Arduino
Environment
Preparation

3



Equipment
Introduction &
Demonstration

Equipment Introduction & Demonstration



1. Grove Vision AI V2

2. XIAO ESP32C3

Equipment Introduction & Demonstration

Open Source

World First MCU-Based Vision AI Module Powered by Himax WiseEye™ WE2 HX6538, based on Arm Cortex-M55 & Ethos-U55

- Cortex -M55 Processor (Big)
- Cortex -M55 Processor (Little)
- Ethos-U55 microNPU
- Security

Grove Vision AI v2

\$15.99 (camera excluded)

Versatile AI Model Supported

TensorFlow PyTorch

Work as a Smart Expansion XIAO / Arduino / Raspberry Pi / BeagleBoard, ESP32 and other mainstream controllers

IIC, UART, SPI

SD Card Slot

Support Multimodel Input

Compatible with all Pi cameras

On-board PDM Microphone

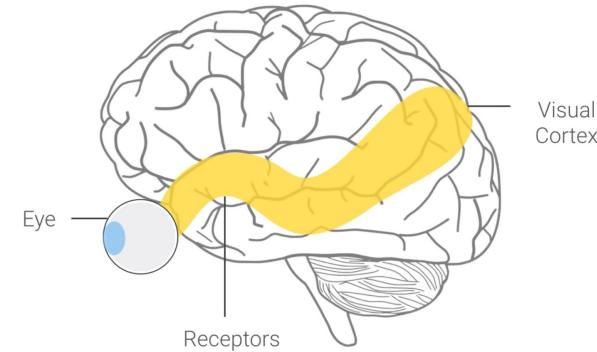
Tailored for battery-powered endpoint AI applications

Significant boost in on-device ML performance

Award-winning low power consumption

Significant boost in on-device ML performance

arm | Himax | seeed studio



Lighting: Lighting illuminates the object or scene to make its features visible.

Lens: This captures the image and delivers it to the sensor in the camera as light.

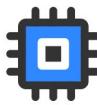
Capture board, frame grabber or sensor: These devices work together to process the image from the camera and convert it to a digital format as pixels.

Processor: The processor runs software and related algorithms that process the digital image and extract the required information.

Communication: These systems enable the machine vision cameras and processing system to communicate with other elements of the bigger system.

seeed studio

Equipment Introduction & Demonstration



Popular & Powerful SoCs Integrated

ESP32, RP2040, nRF52840, SAMD21, and more to come



Extra Tiny Size With SMD

Sized at 21x17.5mm with single-sided surface mount design, ready to blend in space-constrained designs



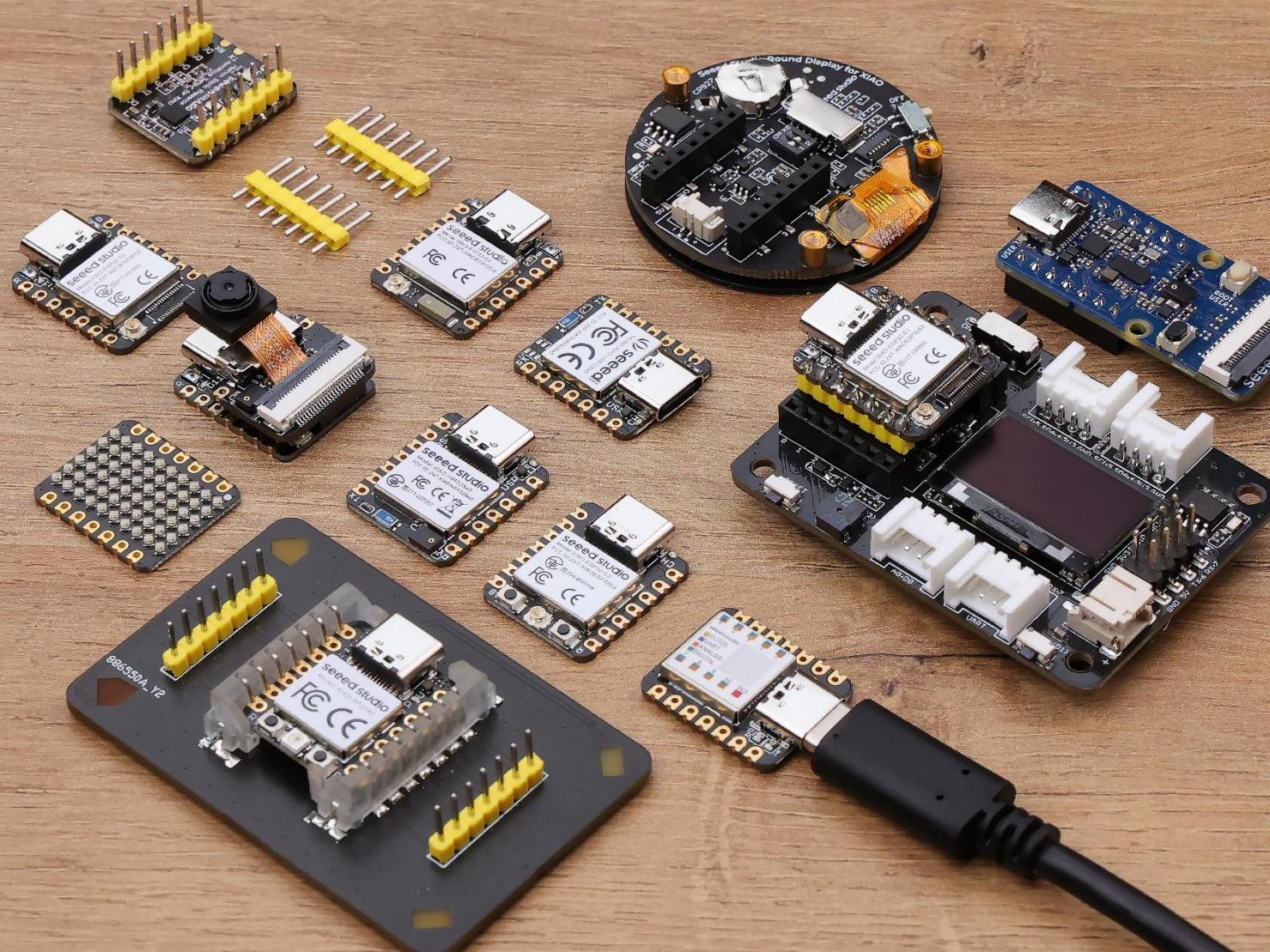
Developer-Friendly

Natively compatible with Arduino, supporting MicroPython, and CircuitPython



IoT Cloud Integration

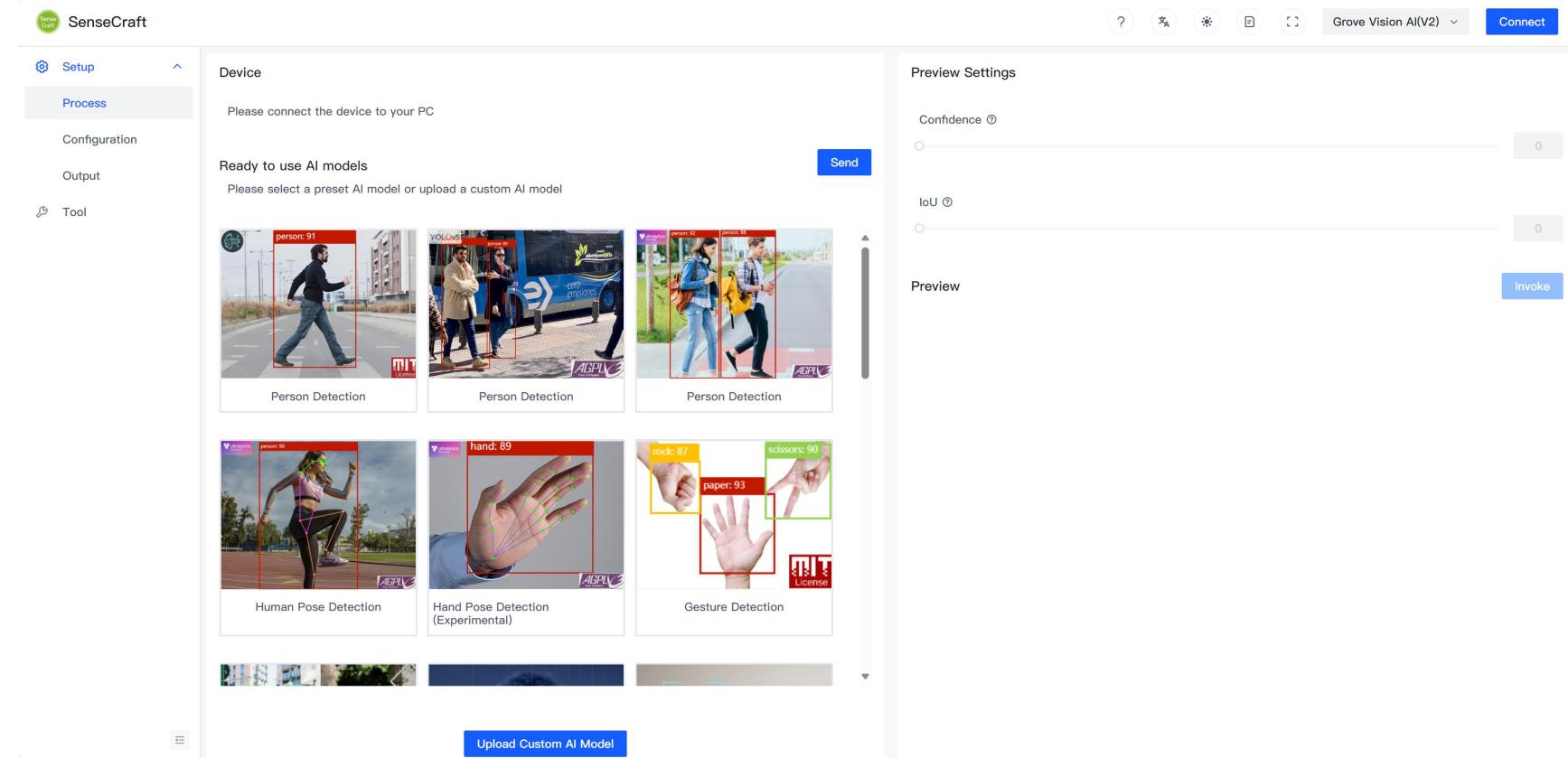
A Seamless integration with AWS, Microsoft Azure, Google Cloud and more



The Seeed Studio XIAO Series is a collection of thumb-sized, powerful microcontroller units (MCUs) tailor-made for space-conscious projects requiring high performance and wireless connectivity. Embodying the essence of popular hardware platforms such as ESP32, RP2040, nRF52840, and SAMD21, the Arduino-compatible XIAO series is the perfect toolset for you to embrace tiny machine learning (TinyML) on the Edge.

Equipment Introduction & Demonstration

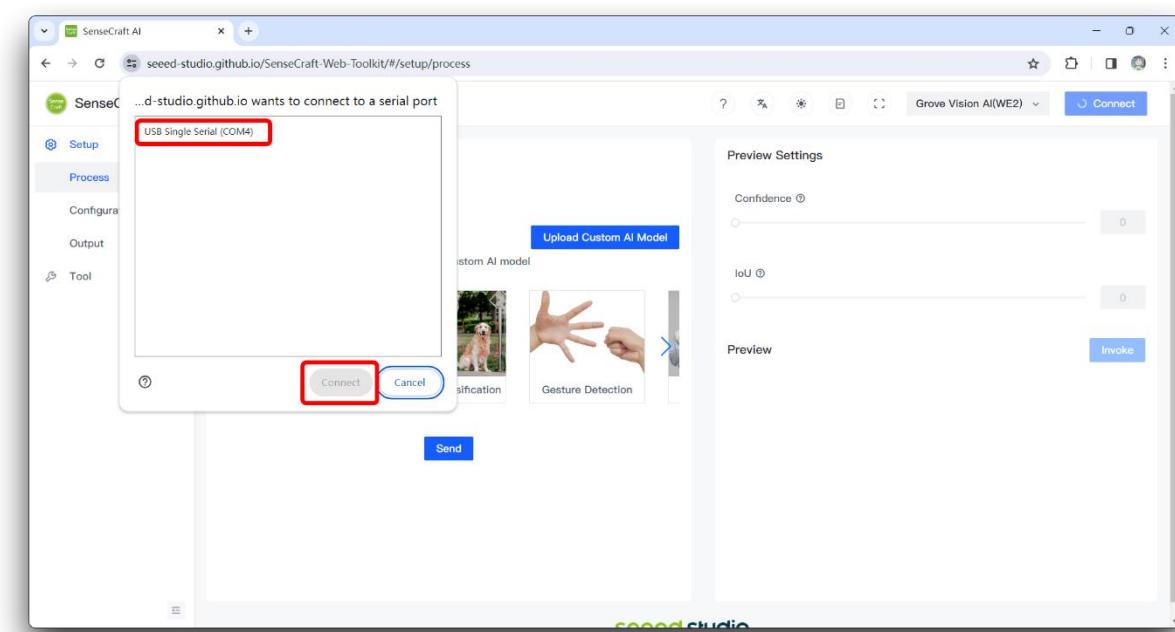
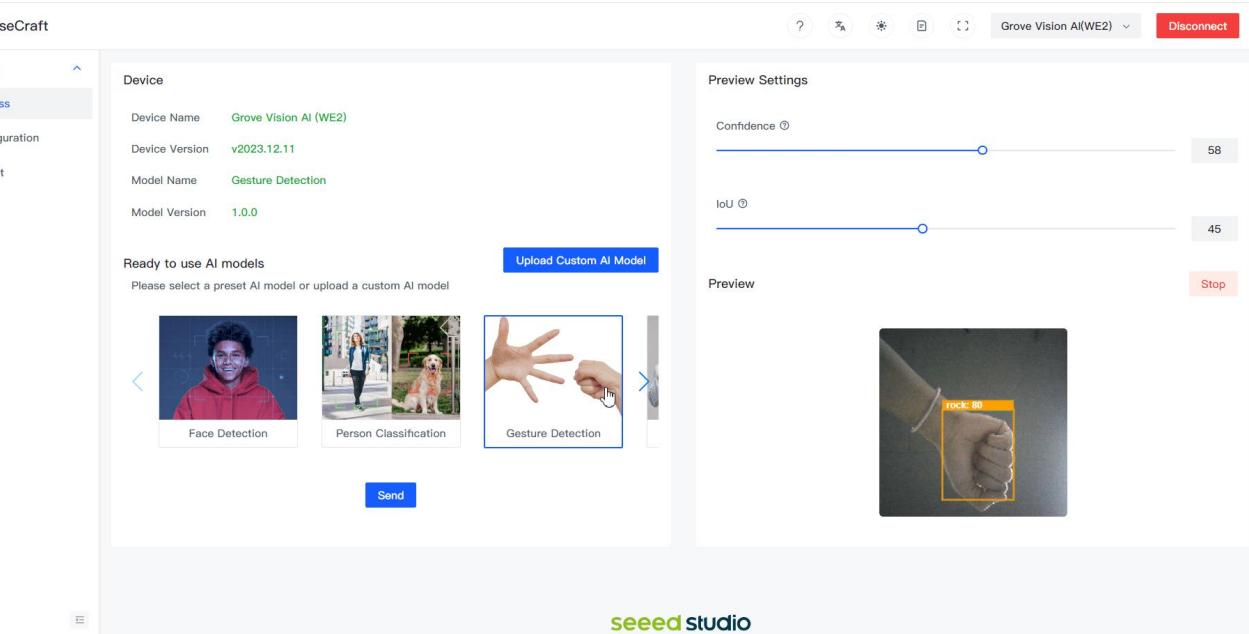
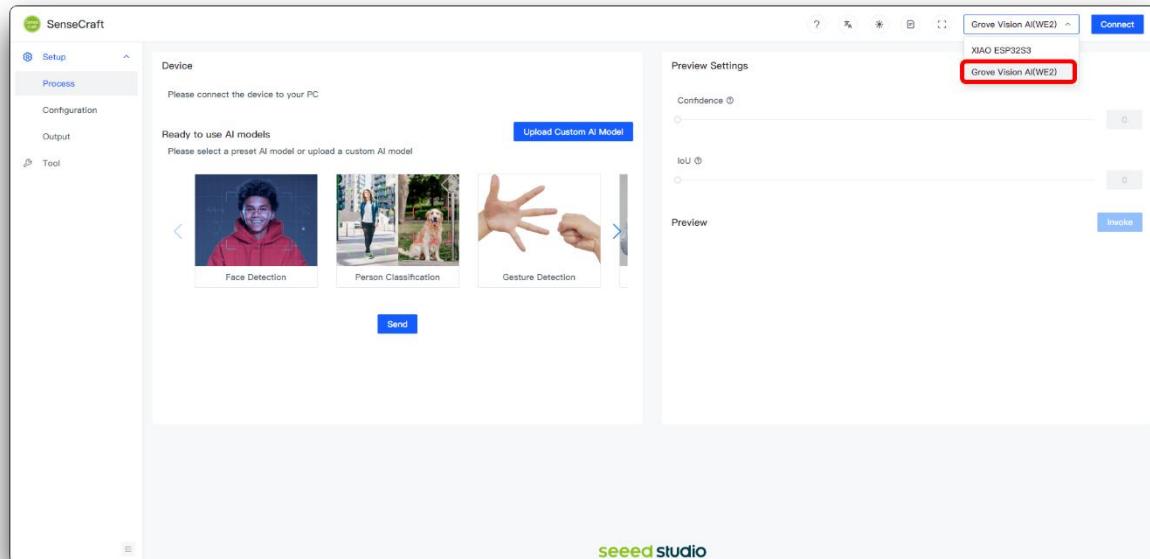
SenseCraft Model Assistant



<https://seeed-studio.github.io/SenseCraft-Web-Toolkit/#/setup/process>

SenseCraft AI empowers users to effortlessly deploy a vast library of publicly available AI models onto their edge devices such as Recomputer (Jetson), XIAO S3, and more, and provides a seamless and user-friendly experience, allowing you to deploy public AI models directly onto your edge devices with just a few clicks. Say goodbye to complex configurations and coding – with SenseCraft AI, you can effortlessly unlock the power of AI on your devices. SenseCraft AI also allows you to upload and share your own trained AI models with the community. By publishing your models, you contribute to a growing library of shared knowledge, fostering collaboration and innovation among AI enthusiasts.

Equipment Introduction & Demonstration



First, we need to open the main SenseCraft AI Model Assistant page.

Please use a Type-C type cable to connect Grove Vision AI V2 to your computer.

In the upper right corner of the SenseCraft AI Model Assistant page, you can select **Grove Vision AI (WE2)**. Then click the **Connect** button on the far right.

In the new window that pops up, select the correct COM port for the device and click the **Connect** button.

Then, just select a model you want to use and click **Send** below. Here is an example of Gesture Detection.

Equipment Introduction & Demonstration

Preview Settings

Confidence ?

96

IoU ?

45

Preview

Stop



Confidence

Confidence refers to the level of certainty or probability assigned by a model to its predictions.

Equipment Introduction & Demonstration

Preview Settings

Confidence ⓘ

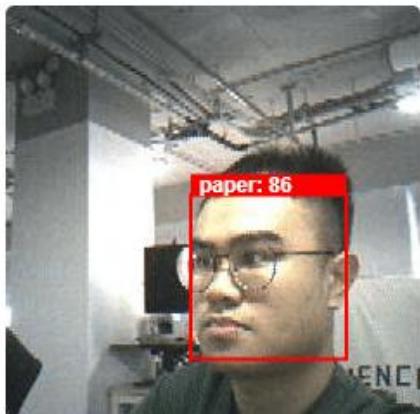


IoU ⓘ



Preview

Stop

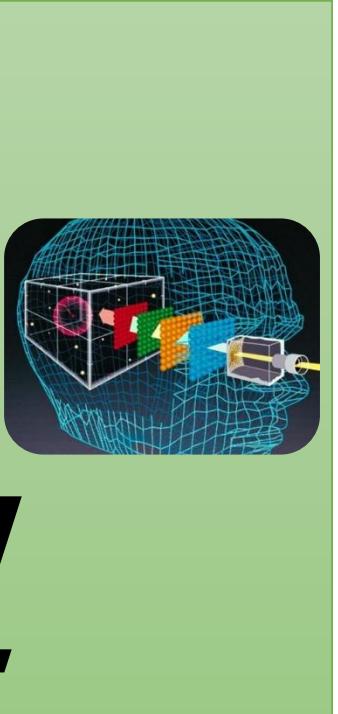


IoU

IoU is used to assess the accuracy of predicted bounding boxes compared to truth bounding boxes.

Main Content

1



Introduction to
Machine Vision

2



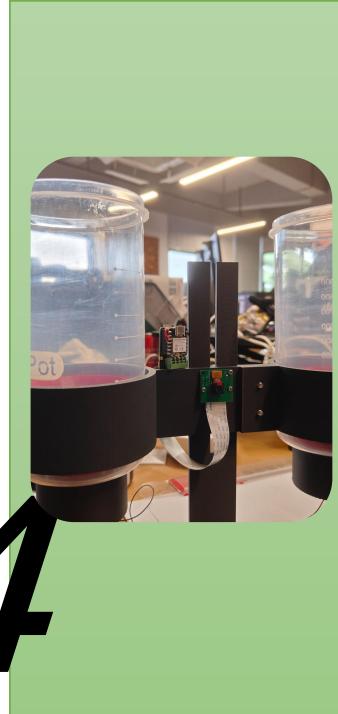
Arduino
Environment
Preparation

3



Equipment
Introduction &
Demonstration

4

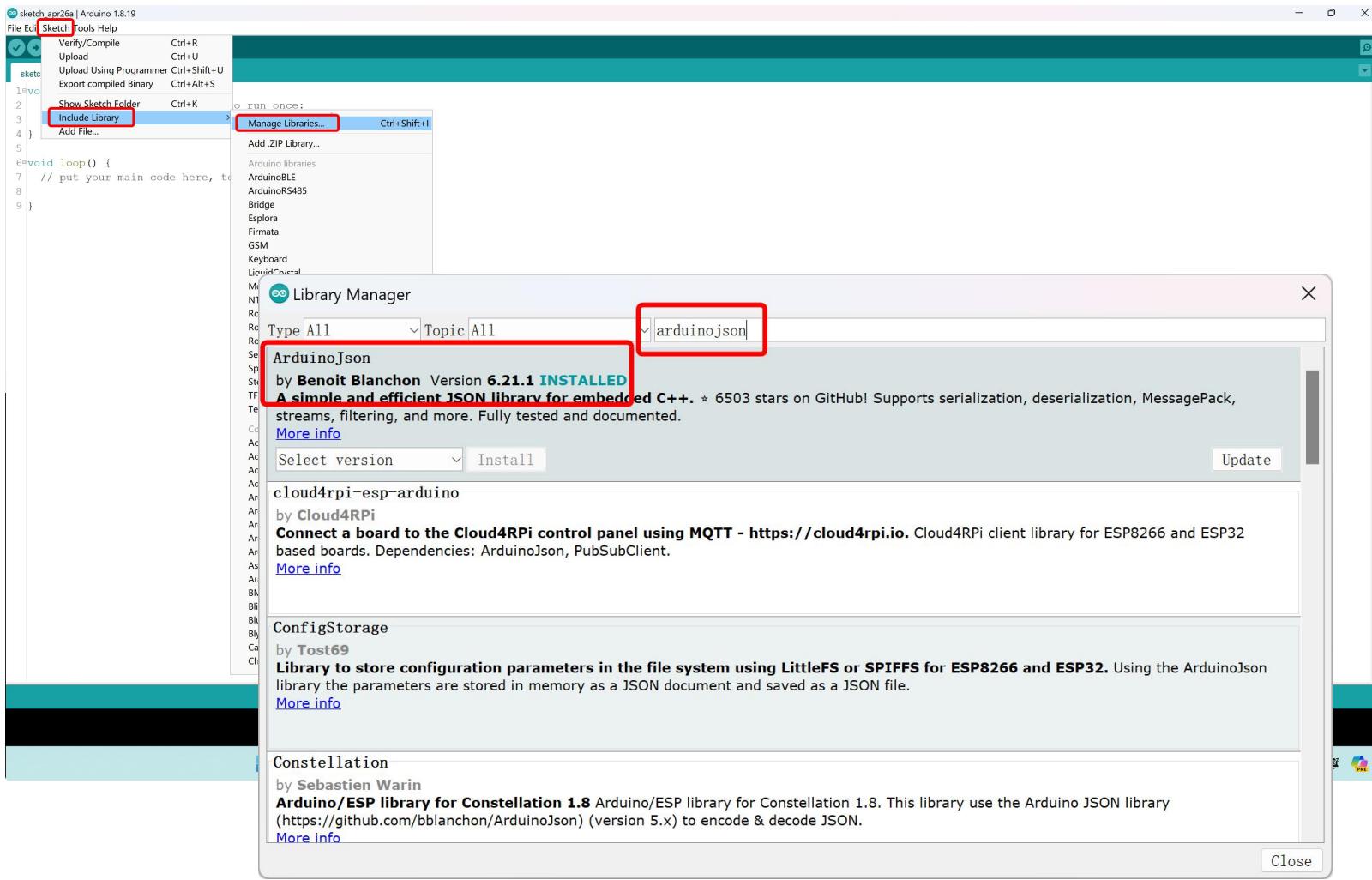


MCU + AI Sensors =
Unlimited Creative
Possibilities

MCU + AI Sensors = Unlimited Creative Possibilities

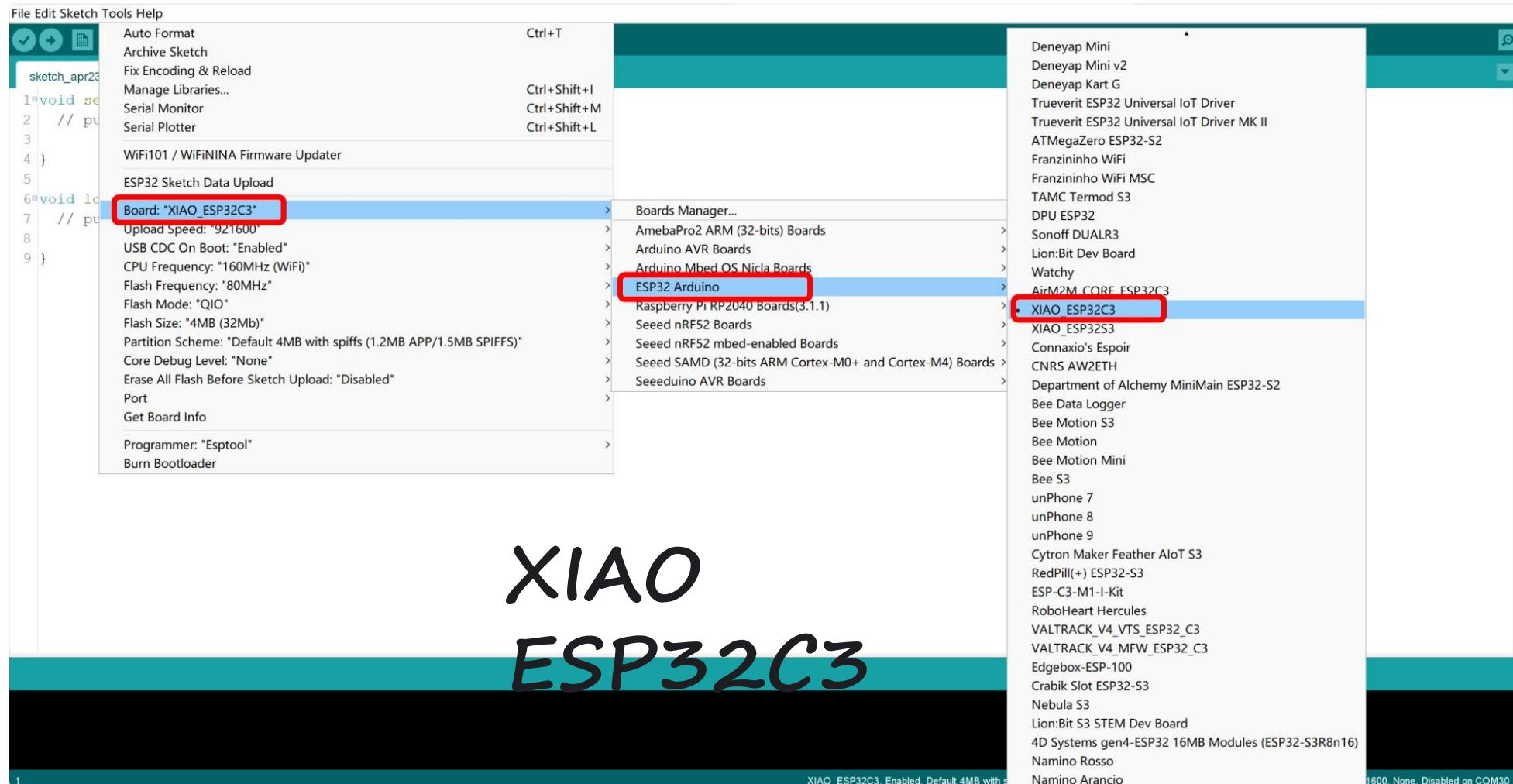


MCU + AI Sensors = Unlimited Creative Possibilities



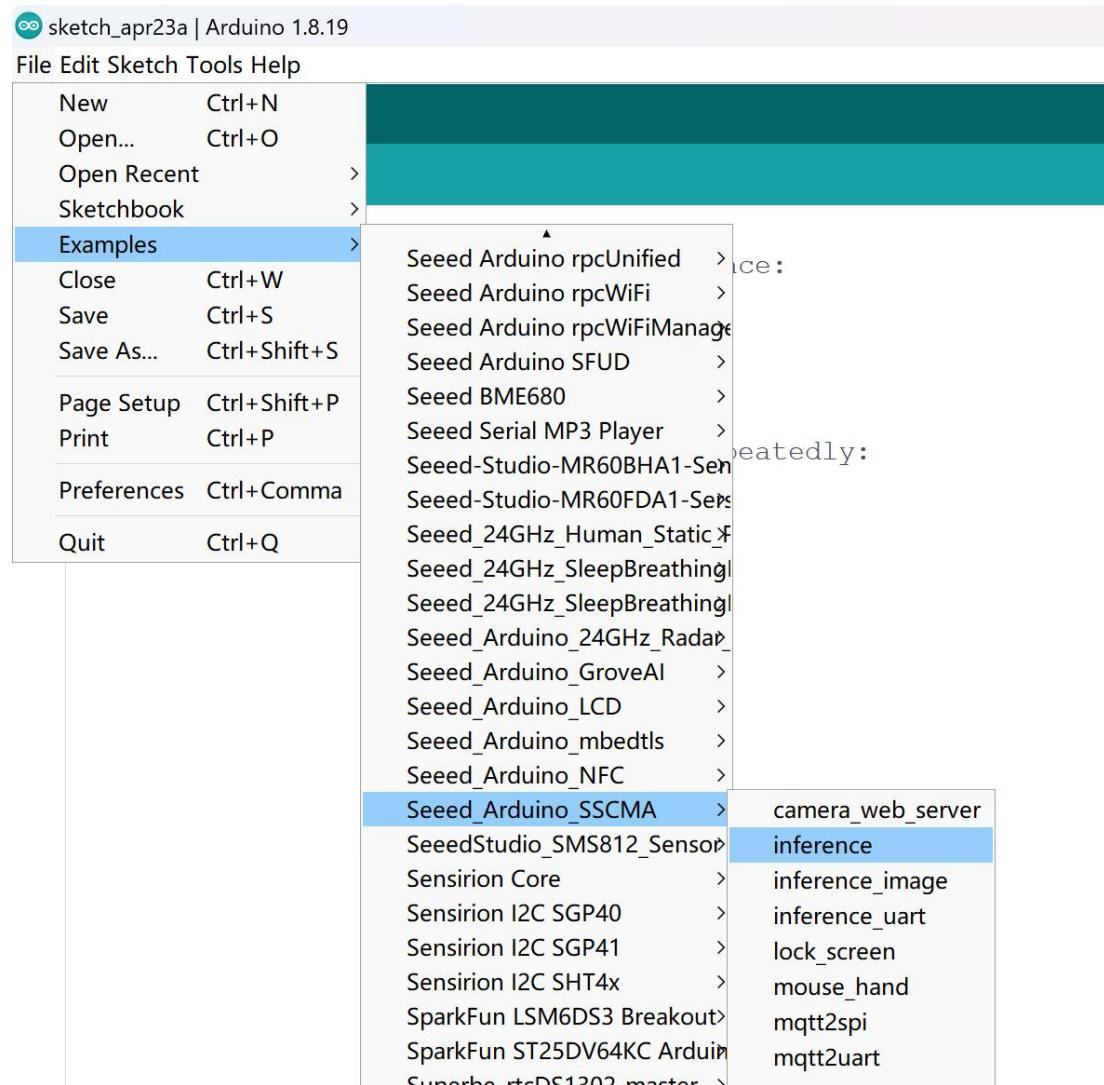
Go to the Sketch menu, then select **Include Library > Manage Libraries...**. This will open the Library Manager. In the search bar at the top of the Library Manager, type in **ArduinoJSON**. The search results will list the ArduinoJSON library. There will be an **Install** button next to the library. Click the **Install** button. The Arduino IDE will automatically download and install the library into your Arduino development environment.

MCU + AI Sensors = Unlimited Creative Possibilities



XIAO
ESP32C3

MCU + AI Sensors = Unlimited Creative Possibilities



https://wiki.seeedstudio.com/grove_vision_ai_v2_software_support/#demo-1-use-xiao-to-get-recognition-results

The screenshot shows the Arduino IDE interface with the Tools menu open. The title bar reads "inference | Arduino 1.8.19". The Tools menu is highlighted with a red box. A red circle with the number 1 is positioned over the Tools menu icon. The menu lists various options: Auto Format, Archive Sketch, Fix Encoding & Reload, Manage Libraries..., Serial Monitor, Serial Plotter, WiFi101 / WiFiNINA Firmware Updater, and ESP32 Sketch Data Upload. Under "Board", it shows "XIAO_ESP32C3", "Upload Speed: 921600", "USB CDC On Boot: Enabled", "CPU Frequency: 160MHz (WiFi)", "Flash Frequency: 80MHz", "Flash Mode: QIO", "Flash Size: 4MB (32Mb)", "Partition Scheme: Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)", "Core Debug Level: None", and "Erase All Flash Before Sketch Upload: Disabled". A red circle with the number 2 is positioned over the "Port" option. The "Port" dropdown menu is open, showing "Serial ports" and a list of COM ports: COM18, COM19, COM49, COM50, and COM6 (ESP32S3 Dev Module). A red box highlights the "COM6 (ESP32S3 Dev Module)" option. A red circle with the number 3 is positioned over the "COM6 (ESP32S3 Dev Module)" entry. The main workspace shows a portion of the "inference" sketch code.

```
#include "camera_web_server.h"
#include "inference.h"

void setup() {
    AI.setAIModel("inference");
    if (AI.begin(1)) {
        Serial.println("AI initialized");
    }
}

void loop() {
    if (AI.isImageAvailable()) {
        inference_image();
    }
}
```

MCU + AI Sensors = Unlimited Creative Possibilities

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. A toolbar with various icons is visible above the code area. The code tab is titled "inference". The code itself is as follows:

```
inference | Arduino 1.8.19
File Edit Sketch Tools Help
inference
1 #include <Seeed_Arduino_SSCMA.h>
2
3 SSCMA AI;
4
5 void setup()
6 {
7     AI.begin();
8     Serial.begin(9600);
9 }
10
11 void loop()
12 {
13     if (!AI.invoke())
14     {
15         Serial.println("invoke success");
16     }
}
Done compiling.
"C:\\\\Users\\\\mengd\\\\AppData\\\\Local\\\\Arduino15\\\\packages\\\\esp32\\\\tools\\\\riscv
"C:\\\\Users\\\\mengd\\\\AppData\\\\Local\\\\Arduino15\\\\packages\\\\esp32\\\\tools\\\\esptool
esptool.py v4.5.1
Creating esp32c3 image...
Merged 2 ELF sections
ESP32C3, Enabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 160MHz (WiFi), QIO, 80MHz, 4MB (32Mb), 921600, None, Disabled on COM5
```

https://wiki.seeedstudio.com/grove_vision_ai_v2_software_support/#demo-1-use-xiao-to-get-recognition-results

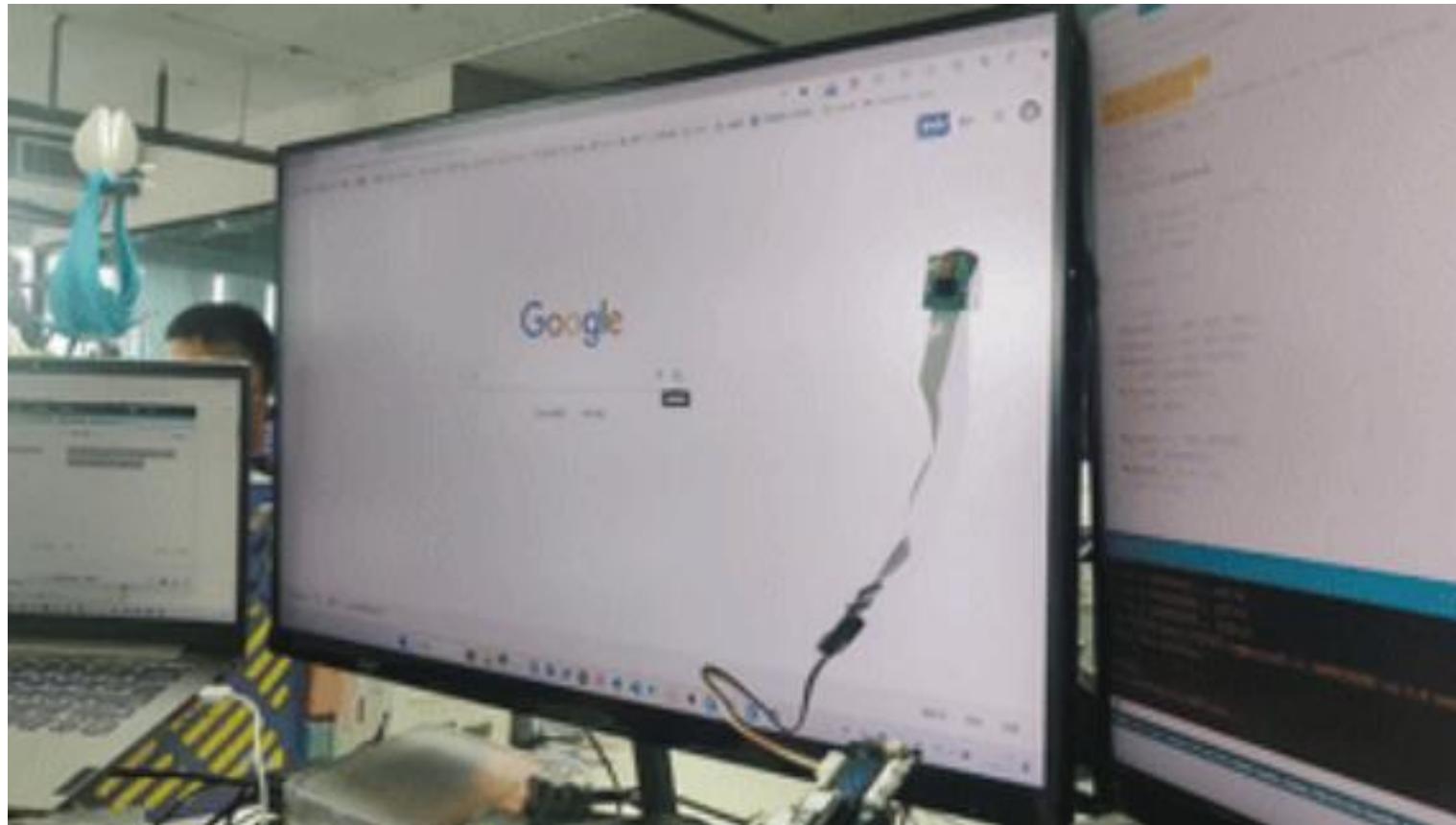
The screenshot shows a terminal window titled "COM5". The window displays a series of log entries indicating AI processing and recognition results. The entries show multiple iterations of preprocessing, inference, and postprocessing steps, along with specific bounding box detections for a target object.

```
COM5
17:02:56.750 -> perf: preprocess=10, inference=45, postprocess=1
17:02:56.841 -> invoke success
17:02:56.841 -> perf: preprocess=10, inference=45, postprocess=0
17:02:56.978 -> invoke success
17:02:56.978 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.114 -> invoke success
17:02:57.114 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.114 -> Box[0] target=1, score=62, x=96, y=135, w=108, h=100
17:02:57.252 -> invoke success
17:02:57.252 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.252 -> Box[0] target=1, score=68, x=105, y=101, w=107, h=95
17:02:57.342 -> invoke success
17:02:57.342 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.481 -> invoke success
17:02:57.481 -> perf: preprocess=10, inference=46, postprocess=0
17:02:57.481 -> Box[0] target=1, score=67, x=120, y=91, w=95, h=100
17:02:57.618 -> invoke success
17:02:57.618 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.618 -> Box[0] target=1, score=77, x=118, y=95, w=95, h=85
17:02:57.709 -> invoke success
17:02:57.709 -> perf: preprocess=10, inference=46, postprocess=0
17:02:57.709 -> Box[0] target=1, score=68, x=111, y=96, w=95, h=85
17:02:57.847 -> invoke success
17:02:57.847 -> perf: preprocess=10, inference=45, postprocess=1
17:02:57.847 -> Box[0] target=1, score=68, x=107, y=105, w=95, h=85
17:02:57.983 -> invoke success
17:02:57.983 -> perf: preprocess=10, inference=45, postprocess=1
```

At the bottom of the terminal window, there are checkboxes for "Autoscroll" and "Show timestamp", and dropdown menus for "Newline", "9600 baud", and "Clear output".

MCU + AI Sensors = Unlimited Creative Possibilities

https://wiki.seeedstudio.com/grove_vision_ai_v2_demo/#step-2-xiao-connects-to-the-computer-and-uploads-the-programme-for-xiao-1



MCU + AI Sensors = Unlimited Creative Possibilities

More incredible creations and ideas...





seeed studio

Vision Challenge

Add a smart eye to your XIAO

- 20+ pre-trained models and no code deployment
- Home assistant supported
- Multi-modal with 400+ Grove extensions
- Yolo v5 & v8 33 fps - a gaming-like smooth experience
- Used as a sub-processor dedicated for AI tasks like skeleton detection



JOIN NOW

FREE products & \$300+ prizes

seeed studio

Thanks you!

seeed studio

Hands-On Keyword Spotting

ICTP-UNU Workshop on TinyML
for Sustainable Development



Main Content



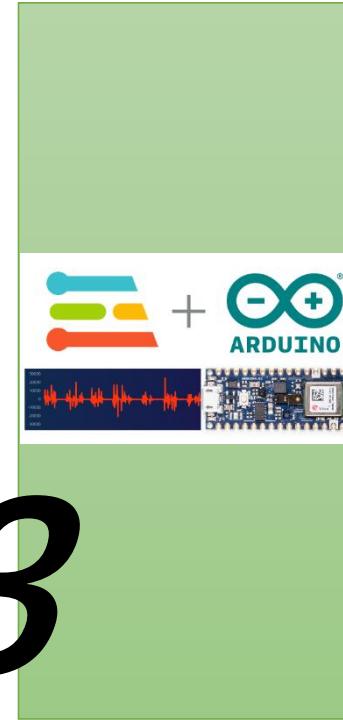
1

Edge Impulse &
XIAO ESP32S3 Sense
Introduction



2

Recording Audio
with XIAO ESP32S3
Sense



3

Training
Exported Models
with Edge
Impulse



4

Deploying models to
XIAO ESP32S3 Sense

Main Content



Edge Impulse &
XIAO ESP32S3 Sense
Introduction

Edge Impulse & XIAO ESP32S3 Sense Introduction



Edge Impulse is a platform for developing machine learning models specifically designed for edge devices and embedded systems. It provides a comprehensive set of tools and services that enable developers to quickly create, train, and deploy machine learning models without requiring deep expertise in machine learning.

Edge Impulse & XIAO ESP32S3 Sense Introduction

Data acquisition and management

The screenshot shows the Edge Impulse project interface for the project "Citric / Lesson1". The left sidebar contains a navigation menu with the following items:

- Dashboard
- Devices
- Data acquisition
 - Create impulse
 - Raw data
 - NN Classifier
- Impulse design
 - EON Tuner
 - Retrain model
 - Live classification
 - Model testing
 - Versioning
 - Deployment
- Try Enterprise Free
- Start free trial

The main content area displays the project details for "Citric / Lesson1". It includes a message about updated resampling algorithm, a maintenance window notice, and a "New tag" button. The "Getting started" section provides links to add existing data, collect new data, and upload a model. The "Sharing" section indicates the project is private. A "Published versions" section lists one version: "Built-in Light Sensor (Rock,Vulcan) LATEST". The bottom right corner features the "seeed studio" logo.

Edge Impulse & XIAO ESP32S3 Sense Introduction

Data processing and feature extraction

EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Impulse design
- Create impulse
- MFCC
- Classifier
- EON Tuner
- Retrain model
- Live classification
- Model testing
- Performance calibration
- Versioning
- Deployment

Try Enterprise Free

Get access to high job limits and training on GPUs.

Start free trial

Citric / xiao-esp32s3-speech PERSONAL

#1 ▾ Click to set a description for this version

Parameters Generate features

Raw data

Show: All label: noise.19.s4 (noi)

audio

Raw features

Label noise

5140, 5160, 5160, 5148, 5152, 5176, 5204, 5196, 5168, 5168, 5184, 5172, 5184, 5180, 5168,...

DSP result

Cepstral Coefficients

Time [sec]

Processed features

?

The screenshot shows the Edge Impulse web interface for a project titled 'xiao-esp32s3-speech'. The main area displays 'Raw data' as a waveform from 0ms to 954ms, with a 'noise' label selected. Below it, 'Raw features' are listed as a series of numerical values, and 'Processed features' are shown as a spectrogram heatmap. On the left, a sidebar lists various project management and development tools. A prominent 'Try Enterprise Free' section offers high job limits and GPU training. At the bottom, a 'Start free trial' button is available.

Edge Impulse & XIAO ESP32S3 Sense Introduction

Model training and optimization

The screenshot shows the Edge Impulse web interface for a project named "xiao-esp32s3-speech". The left sidebar includes options like Dashboard, Devices, Data acquisition, Impulse design, Create impulse, MFCC, Classifier, EON Tuner, Retrain model, Live classification, Model testing (selected), Performance calibration, Versioning, Deployment, and a Try Enterprise Free button.

The main area displays "Test data" and "Model testing output". The "Test data" section lists samples with their expected outcomes, lengths, accuracy, and results. The "Model testing output" section shows accuracy (63.35%) and a confusion matrix:

	HELLO	NOISE	STOP	UNCERTAIN
HELLO	19.3%	78.9%	0%	1.8%
NOISE	0%	99.2%	0.8%	0%
STOP	0%	68.9%	26.2%	4.9%
F1 SCORE	0.32	0.75	0.41	

The "Feature explorer" section contains a scatter plot with points colored by category: hello - correct (yellow), noise - correct (green), stop - correct (blue), hello - incorrect (pink), noise - incorrect (red), and stop - incorrect (orange).

Edge Impulse & XIAO ESP32S3 Sense Introduction

Model deployment and integration

The screenshot shows the Edge Impulse web interface for a project named "xiao-esp32s3-speech". The left sidebar includes links for Dashboard, Devices, Data acquisition, Impulse design (Create impulse, MFCC, Classifier), EON Tuner, Retrain model, Live classification, Model testing, Performance calibration, Versioning, Deployment, and a "Try Enterprise Free" section with a "Start free trial" button. The main area displays the "Configure your deployment" screen, which lists various deployment options:

- C++ library**: A portable C++ library with no external dependencies, which can be compiled with any modern C++ compiler.
- Arduino library**: An Arduino library with examples that runs on most Arm-based Arduino development boards.
- BrainChip MetaTF Model**: A MetaTF converted model (.fbz) for use with the BrainChip Akida™ runtime.
- Cube.MX CMSIS-PACK**: A STM32Cube.MX CMSIS-PACK, for fast inferencing on many ST MCUs.
- Custom block**: A ZIP file containing everything your custom deploy block will receive.

Below the deployment options is a table comparing resource usage for the selected "Selected" deployment method:

	LATENCY	289 ms.	5 ms.	294 ms.
RAM	15.6K		3.7K	15.6K
FLASH	-		26.7K	-
ACCURACY				-

At the bottom, there is another table for "Unoptimized (float32)" deployment:

	MFCC	CLASSIFIER	TOTAL
LATENCY	289 ms.	42 ms.	331 ms.
RAM	15.6K	7.0K	15.6K

The right side of the interface shows the "Latest build" (v5 (C++ library) from Jul 19 2023, 14:32:43) and a "Run this model" section with a QR code and a "Launch in browser" button.

Edge Impulse & XIAO ESP32S3 Sense Introduction

Seeed Studio XIAO ESP32S3 Sense

Ultra-small ESP32-S3 development board with OV2640 camera, a rising star in the fields of TinyML and Smart Home

HIGH-PERFORMANCE

240MHz Xtensa 32-bit LX7 dual-core processor

MEMORY

8MB PSRAM + 8MB FLASH

MULTI-FUNCTIONAL

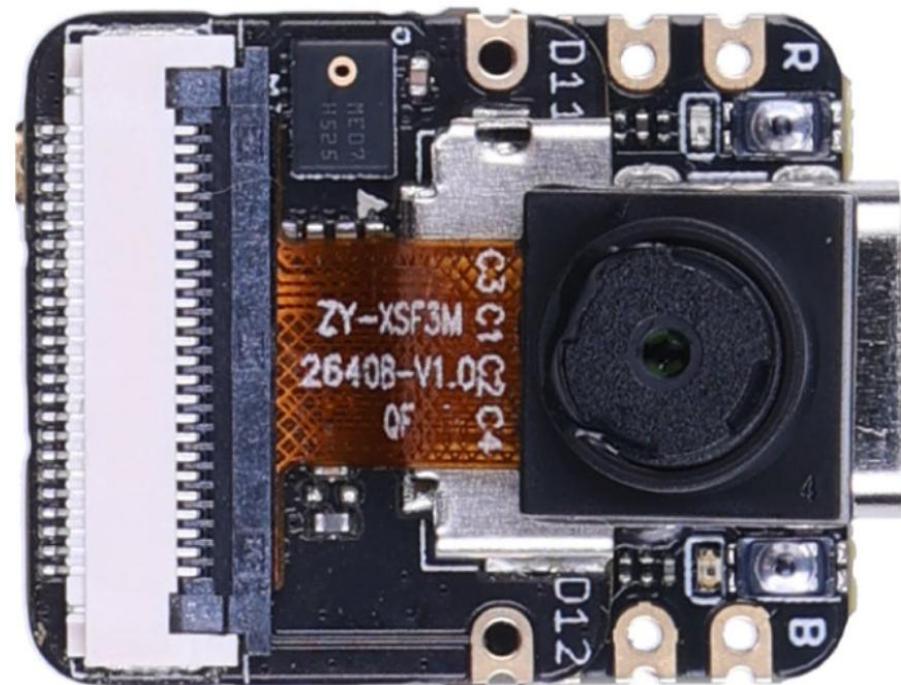
Microphone/SD card slot/Detachable OV2640

WIRELESS

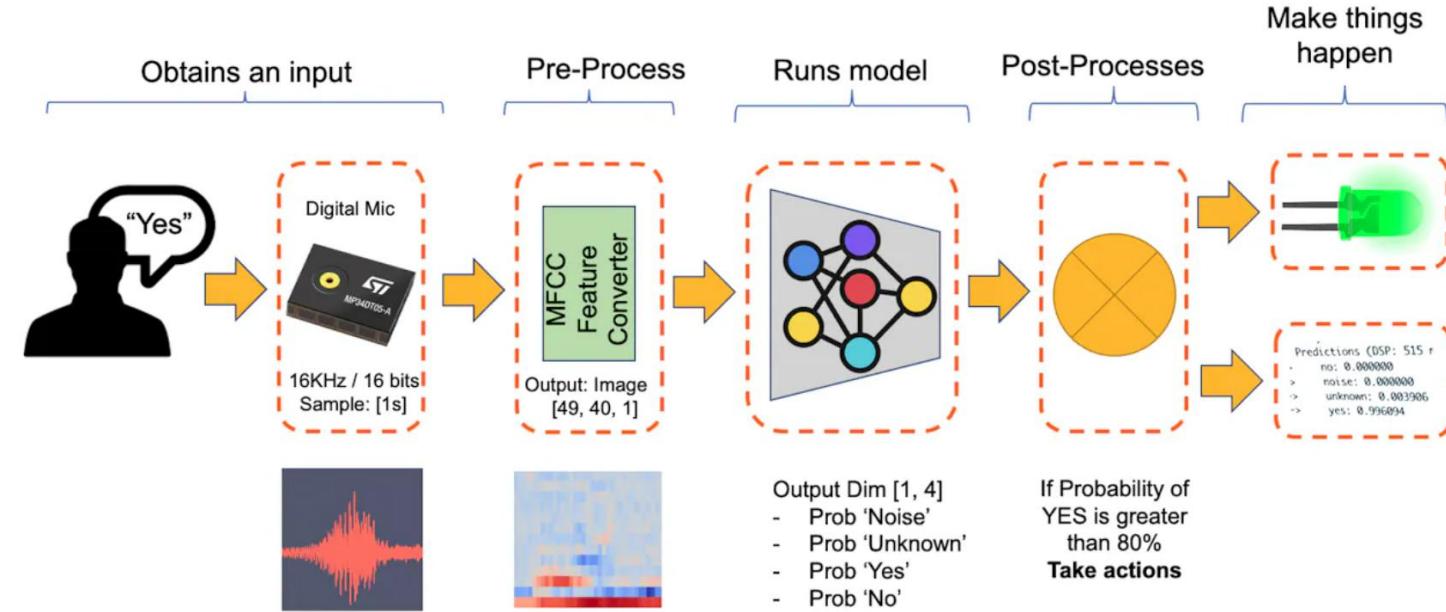
2.4GHz WiFi and BLE 5

TINYML-SUPPORTED

Image Processing/Speech Recognition



Edge Impulse & XIAO ESP32S3 Sense Introduction

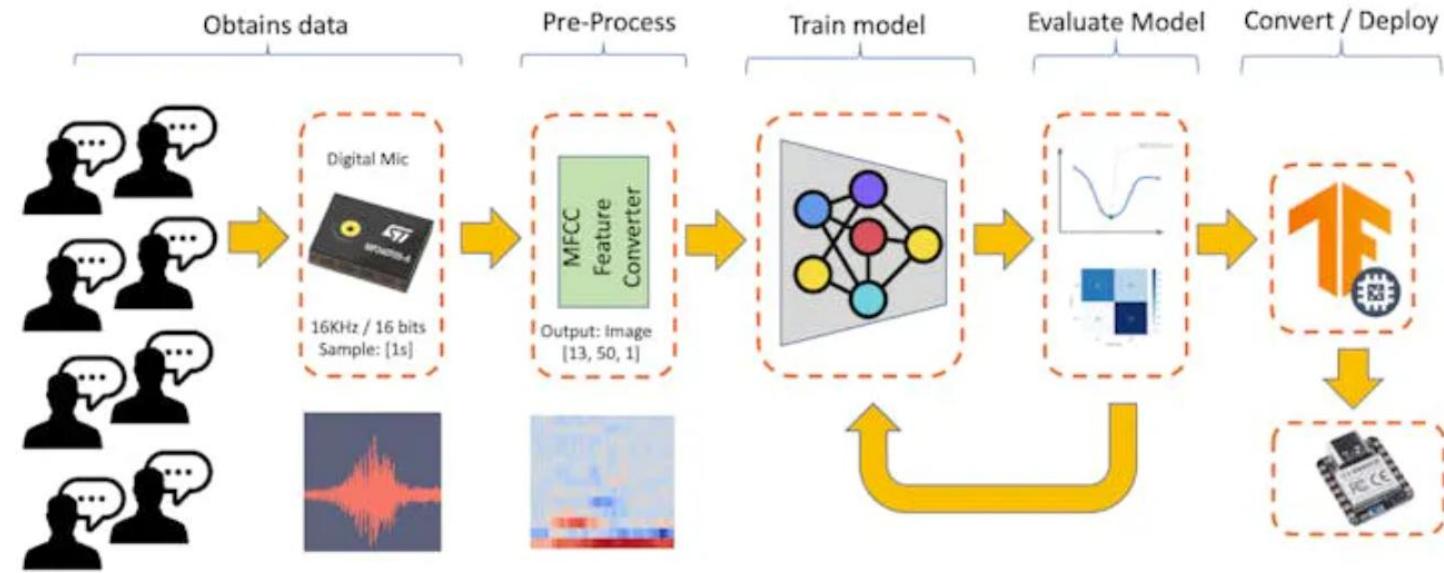


The principle of keyword spotting is based on machine learning and speech recognition technologies. First, a training dataset containing samples of keywords and non-keywords is constructed. The audio data is preprocessed and features are extracted to convert it into a suitable input format for machine learning models.

Next, the keyword detection model is trained using the training dataset. The model learns to distinguish between keyword and non-keyword samples by establishing a discriminative decision boundary. Common model architectures include CNN, RNN, and DNN, which effectively capture temporal and frequency features in speech signals.

During the inference stage, the pre-trained keyword detection model takes the real-time recorded audio data as input, extracts features, and performs classification. If the model's output probability exceeds a set threshold, it is considered that the keyword has been detected, triggering the wake-up mechanism of the voice assistant.

Edge Impulse & XIAO ESP32S3 Sense Introduction



Data Acquisition: The device's microphone continuously records audio data and transmits it to the keyword detection module.

Preprocessing: The recorded audio data undergoes preprocessing, including noise reduction, silence removal, normalization, etc., to improve audio quality and reduce the impact of noise.

Feature Extraction: The preprocessed audio data is converted into a feature representation suitable for machine learning models. Common feature extraction methods include Mel-Frequency Cepstral Coefficients (MFCC) and Mel-Frequency Energy (MFE).

Keyword Detection: A pre-trained keyword detection model is used to classify the extracted features and determine whether the current audio data contains the keyword. Popular keyword detection models include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN).

Post-processing: The results of keyword detection are post-processed, such as setting thresholds and smoothing, to improve detection accuracy and stability.

Wake-up: If the keyword is detected, the wake-up mechanism of the voice assistant is triggered, and subsequent voice commands are transmitted to the cloud for further processing.

Main Content



1

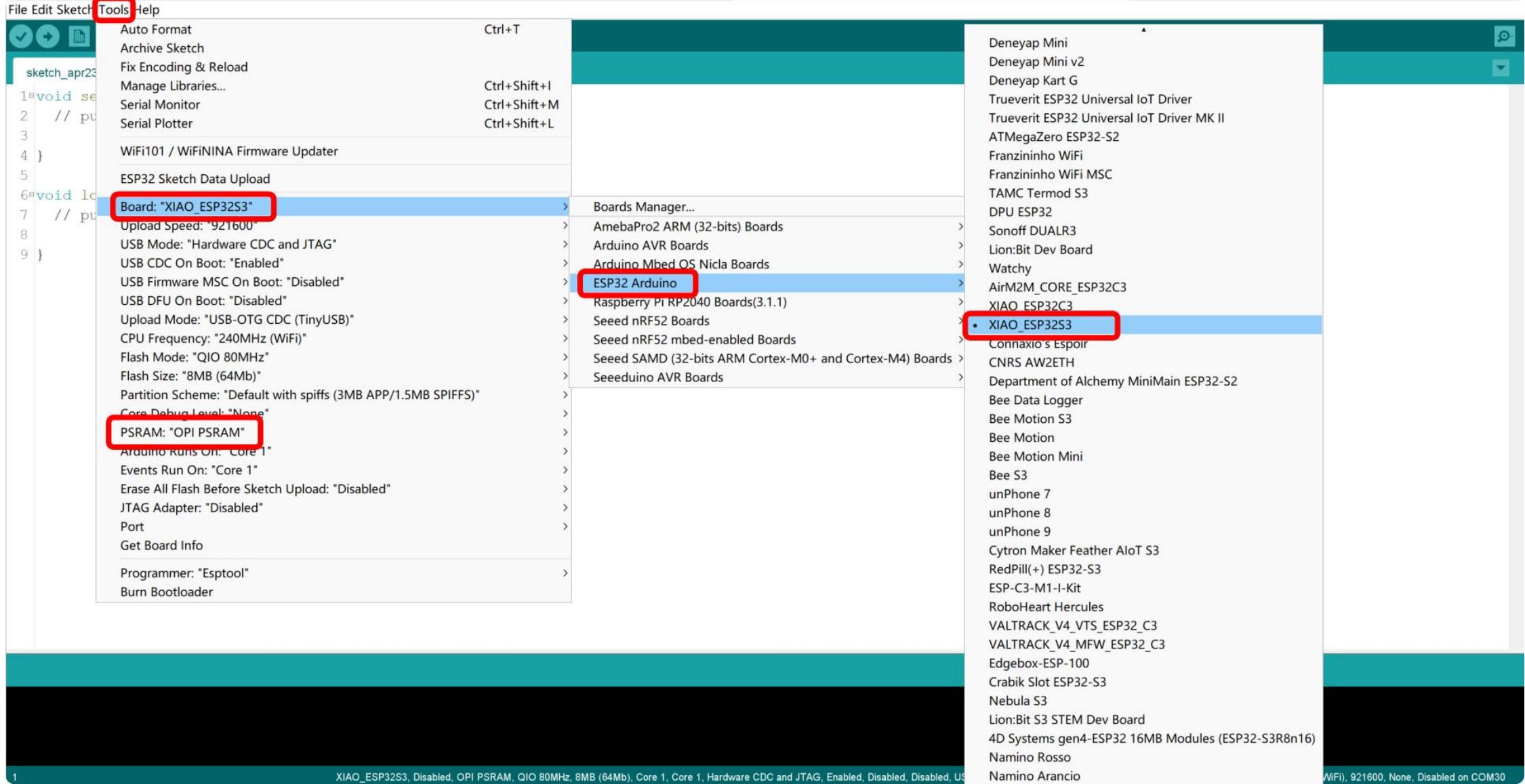
Edge Impulse &
XIAO ESP32S3 Sense
Introduction



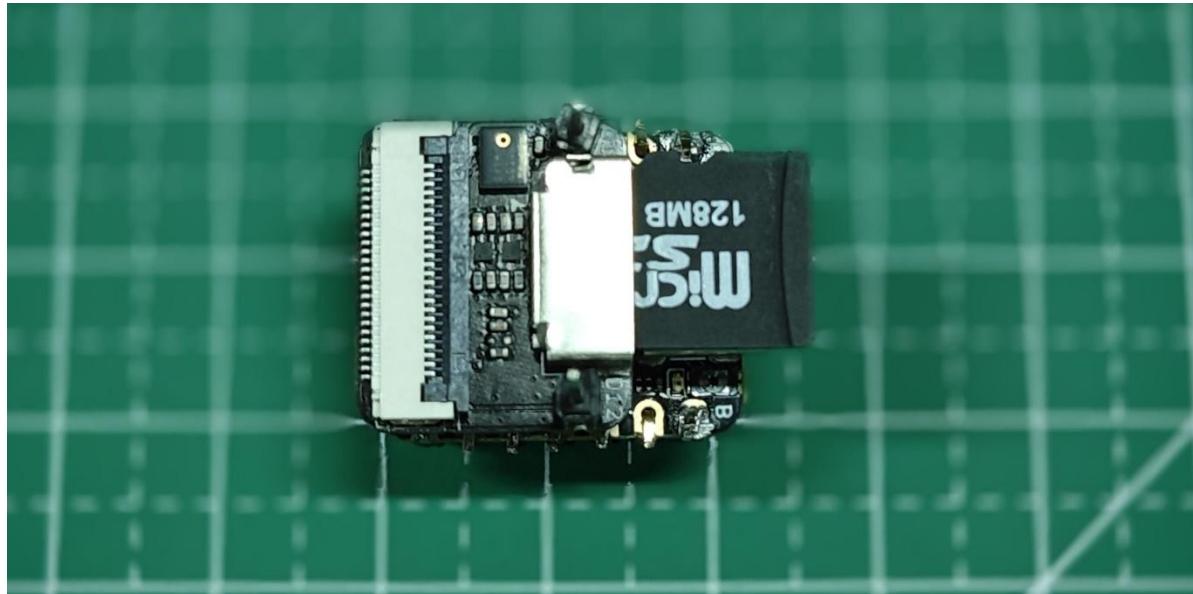
2

Recording Audio
with XIAO ESP32S3
Sense

Recording Audio with XIAO ESP32S3 Sense



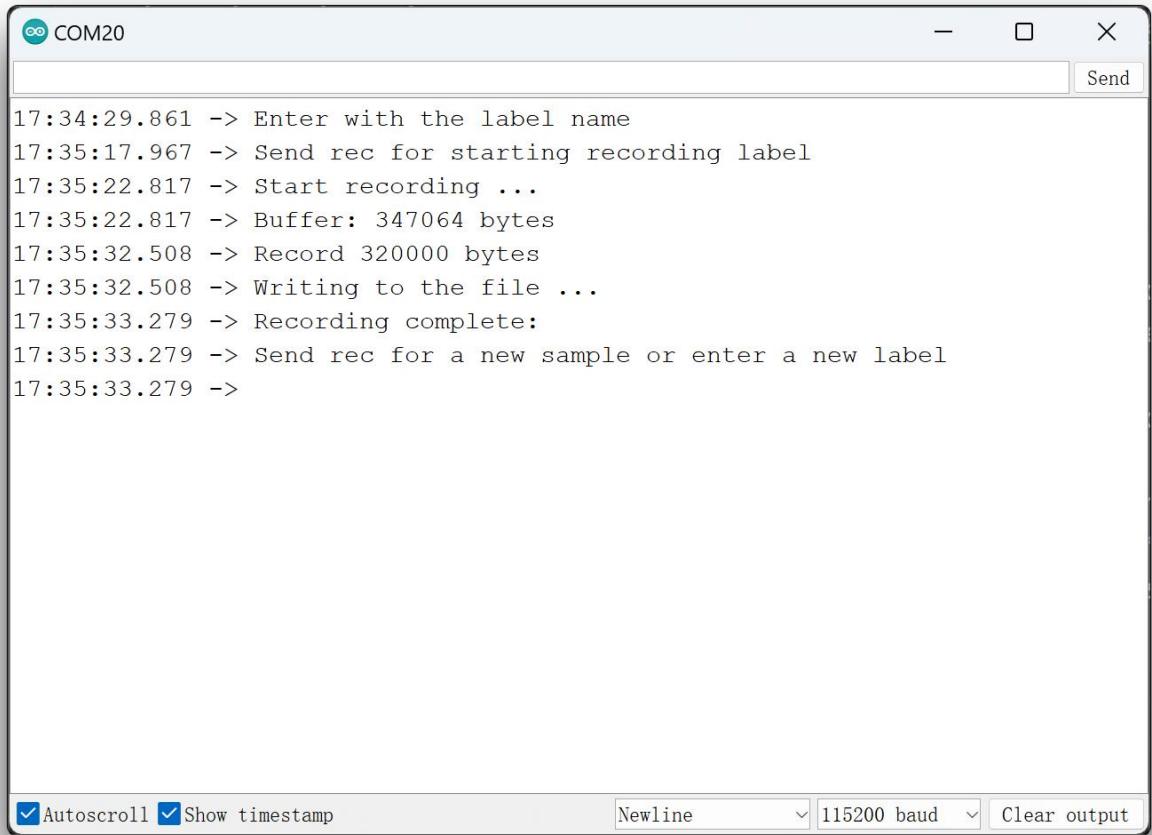
Recording Audio with XIAO ESP32S3 Sense



Insert the microSD card into the microSD card slot. Please note the direction of insertion, the side with the gold finger should face inward.

Recording Audio with XIAO ESP32S3 Sense

https://wiki.seeedstudio.com/tinyml_course_Key_Word_Spotting/#step-1-save-recorded-sound-samples-as-wav-audio-files-to-a-microsd-card



The screenshot shows the Arduino Serial Monitor window titled "COM20". The window displays a log of commands sent to the XIAO ESP32S3 Sense board. The log includes:

```
17:34:29.861 -> Enter with the label name
17:35:17.967 -> Send rec for starting recording label
17:35:22.817 -> Start recording ...
17:35:22.817 -> Buffer: 347064 bytes
17:35:32.508 -> Record 320000 bytes
17:35:32.508 -> Writing to the file ...
17:35:33.279 -> Recording complete:
17:35:33.279 -> Send rec for a new sample or enter a new label
17:35:33.279 ->
```

At the bottom of the window, there are checkboxes for "Autoscroll" and "Show timestamp", and dropdown menus for "Newline", "115200 baud", and "Clear output".

Now, upload the code to the XIAO and get samples from the keywords (hello and stop). You can also capture noise and other words. The Serial monitor will prompt you to receive the label to be recorded.

Send the label (for example, hello). The program will wait for another command: rec.

And the program will start recording new samples every time a command rec is sent. The files will be saved as hello.1.wav, hello.2.wav, hello.3.wav, etc. until a new label (for example, stop) is sent. In this case, you should send the command rec for each new sample, which will be saved as stop.1.wav, stop.2.wav, stop.3.wav, etc.

Ultimately, we will get the saved files on the SD card.

Recording Audio with XIAO ESP32S3 Sense

Use a card reader to save all the sound samples stored inside the SD card to your computer.

Main Content



1

Edge Impulse &
XIAO ESP32S3 Sense
Introduction



2

Recording Audio
with XIAO ESP32S3
Sense



3

Training
Exported Models
with Edge
Impulse

Training Exported Models with Edge Impulse

Join us May 14th for a synthetic data with NVIDIA Omniverse webinar. [Register Here](#) X

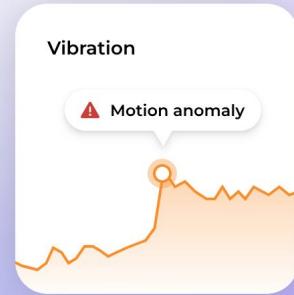
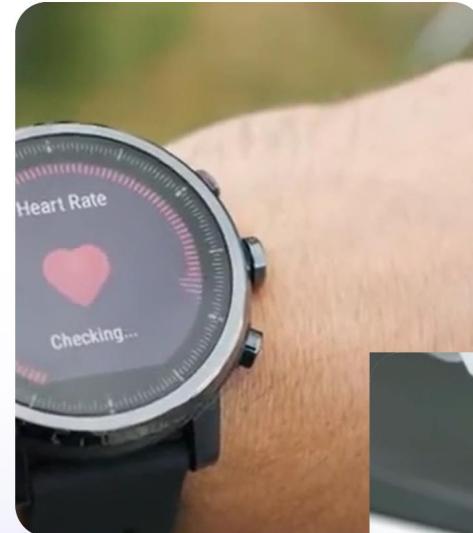
 **EDGE IMPULSE** Product ▾ Solutions ▾ Developers ▾ Pricing Company ▾ Blog [Login](#) [Get started](#)

<https://edgeimpulse.com/>

Build. Train. Optimize. AI for the edge.

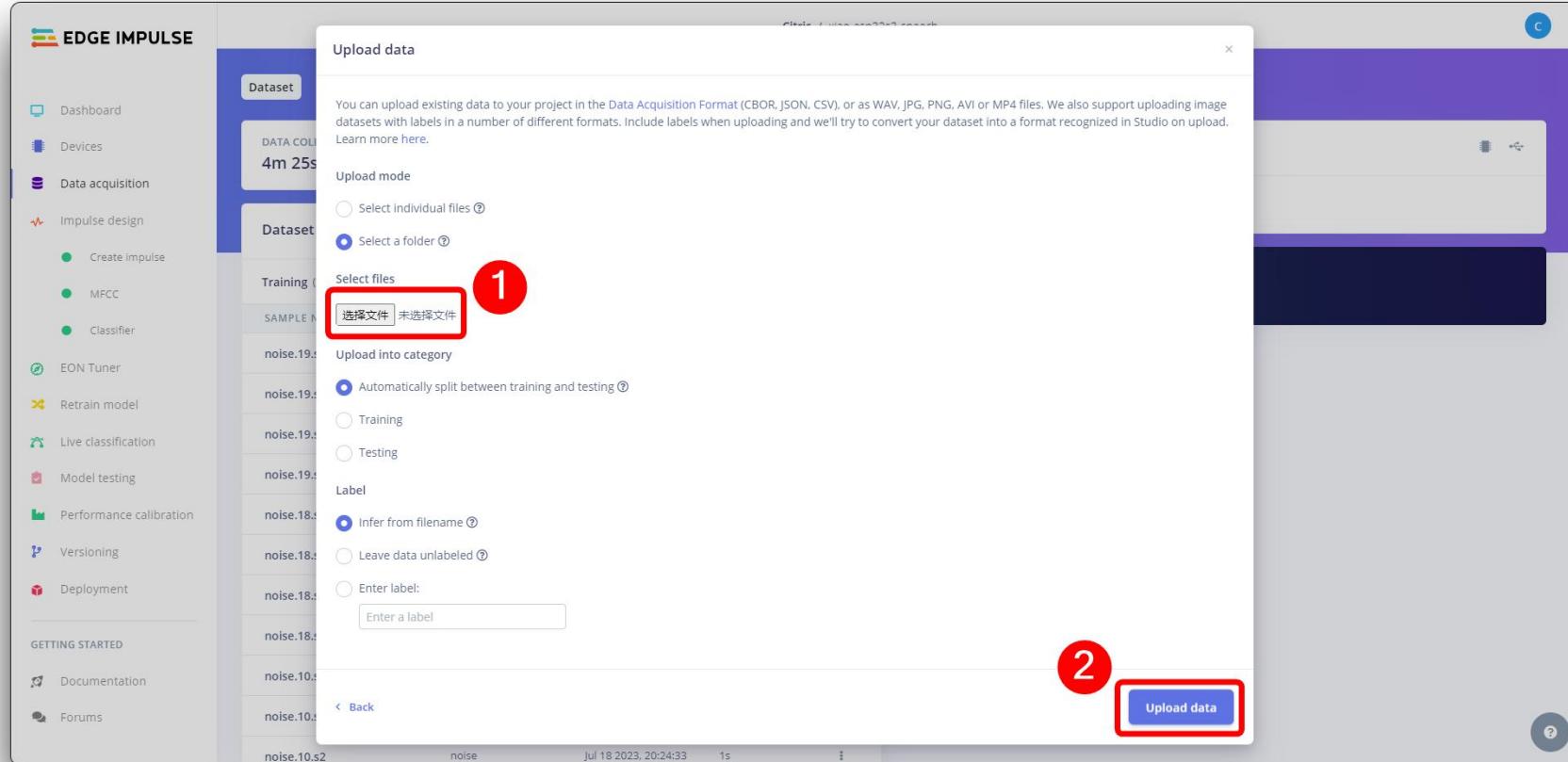
Build datasets, train models, and optimize libraries to run on any edge device, from extremely low-power MCUs to efficient Linux CPU targets and GPUs.

[Get Started](#) [Schedule a demo](#)



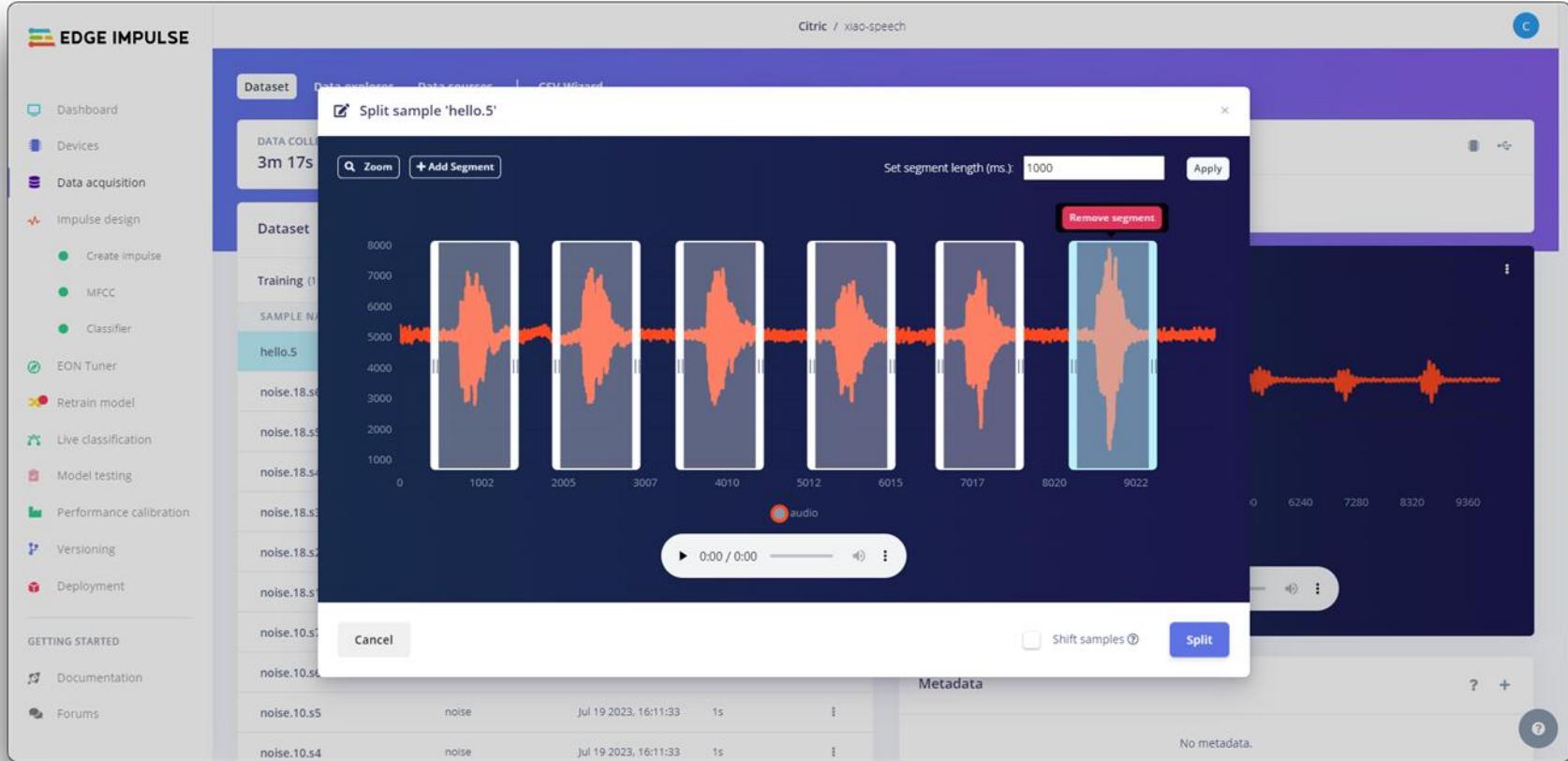
seeed studio

Training Exported Models with Edge Impulse



When the raw dataset is defined and collected, we should initiate a new project at [Edge Impulse](#). Once the project is created, select the **Upload Existing Data** tool in the **Data Acquisition** section. Choose the files to be uploaded.

Training Exported Models with Edge Impulse

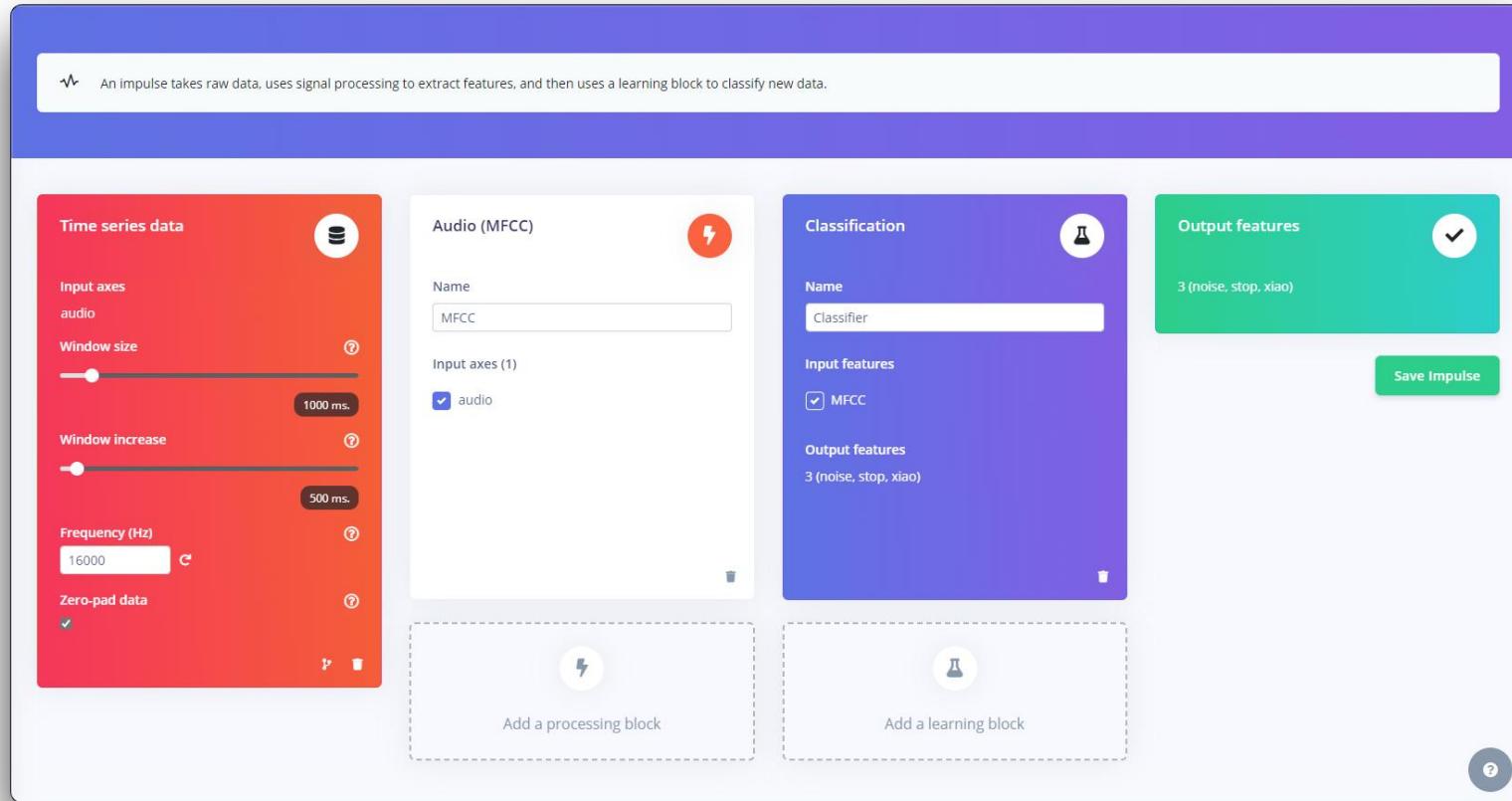


All data on dataset have a 1s length, but the samples recorded in the previous section have 10s and must be split into 1s samples to be compatible. Click on three dots after the sample name and select Split sample.

Once inside de tool, split the data into 1-second records. If necessary, add or remove segments.

This procedure should be repeated for all samples.

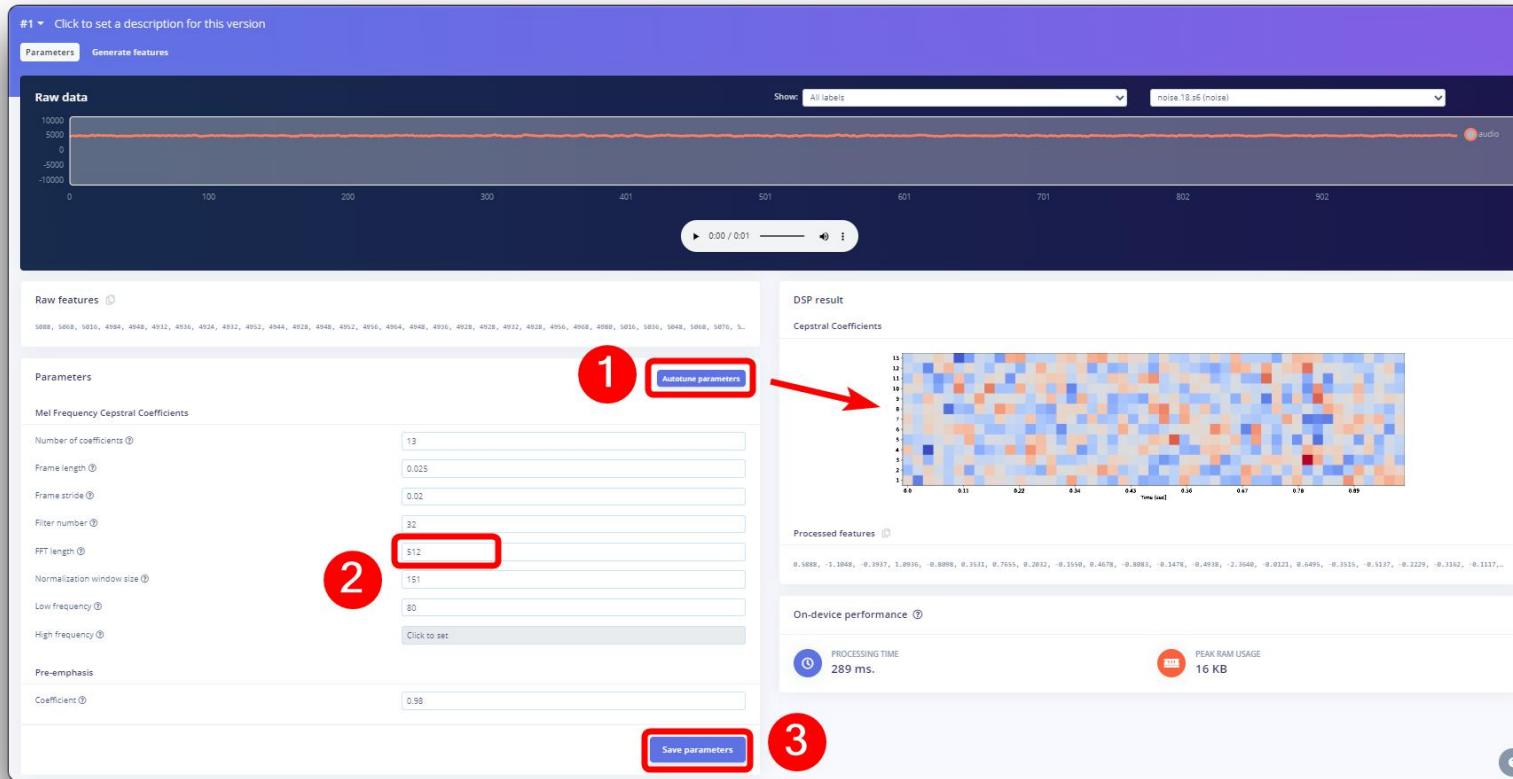
Training Exported Models with Edge Impulse



An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

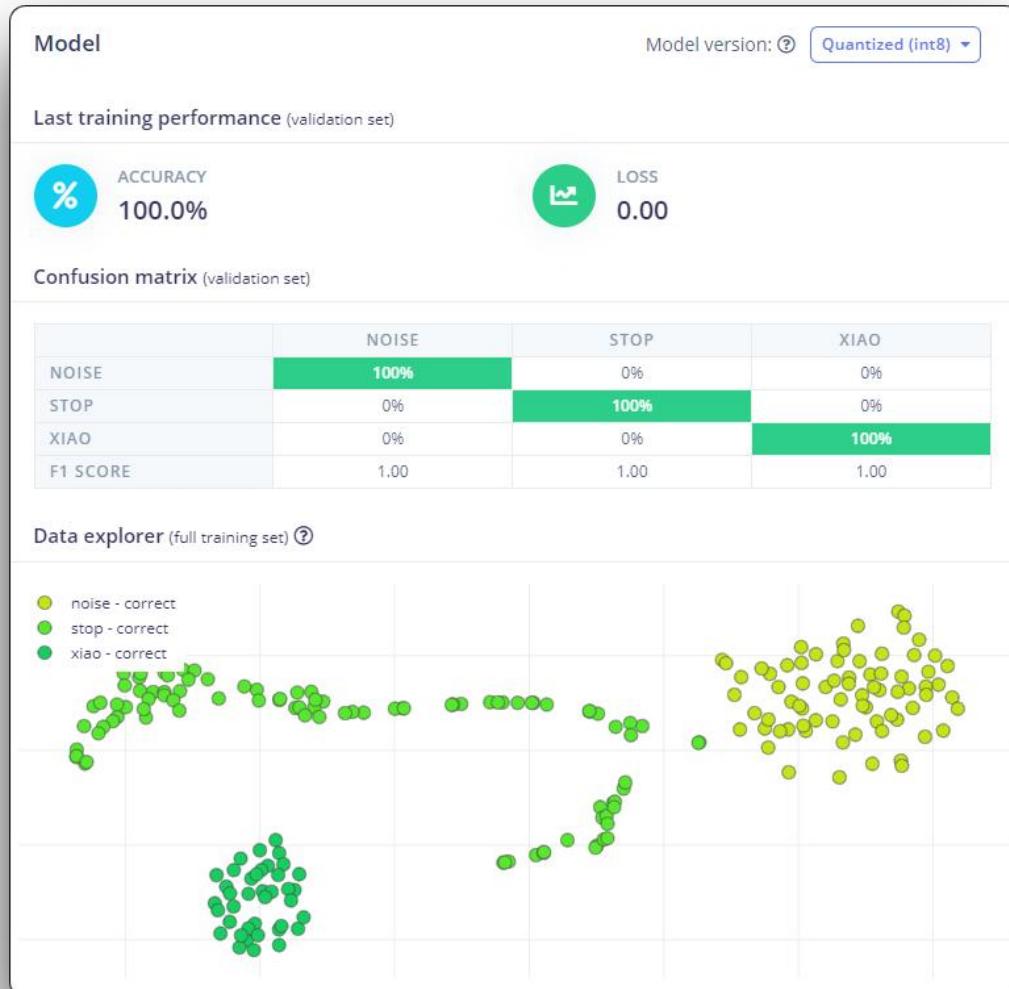
First, we will take the data points with a 1-second window, augmenting the data, sliding that window each 500ms. Note that the option zero-pad data is set. This is important to fill with zeros samples smaller than 1 second (in some cases, I reduced the 1000 ms window on the split tool to avoid

Training Exported Models with Edge Impulse



The next step is to create the images to be trained in the next phase. We can keep the default parameter values or take advantage of the **DSP Autotuneparameters** option, which we will do.

Training Exported Models with Edge Impulse



We will use a Convolution Neural Network (CNN) model. The basic architecture is defined with two blocks of Conv1D + MaxPooling (with 8 and 16 neurons, respectively) and a 0.25 Dropout. And on the last layer, after Flattening four neurons, one for each class.

Training Exported Models with Edge Impulse

Configure your deployment

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more](#).

Arduino library

SELECTED DEPLOYMENT
Arduino library
An Arduino library with examples that runs on most Arm-based Arduino development boards.

MODEL OPTIMIZATIONS
Model optimizations can increase on-device performance but may reduce accuracy.

Enable EON™ Compiler Same accuracy, up to 50% less memory. [Learn more](#)

	MFCC	CLASSIFIER	TOTAL
LATENCY	289 ms.	6 ms.	295 ms.
RAM	15.6K	6.0K	15.6K
FLASH	-	49.5K	-
ACCURACY			-

Quantized (int8) Selected ✓

	MFCC	CLASSIFIER	TOTAL
LATENCY	289 ms.	6 ms.	295 ms.
RAM	15.6K	6.0K	15.6K
FLASH	-	49.5K	-
ACCURACY			-

Unooptimized (float32) Select

	MFCC	CLASSIFIER	TOTAL
LATENCY	289 ms.	50 ms.	339 ms.
RAM	15.6K	10.5K	15.6K
FLASH	-	52.2K	-
ACCURACY			-

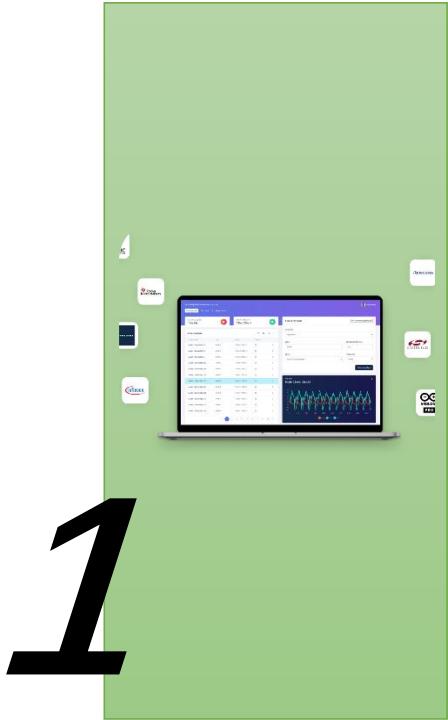
To compare model accuracy, run model testing for all available optimizations.

Run model testing

Edge Impulse will package all the needed libraries, preprocessing functions, and trained models, downloading them to your computer. You should select the option Arduino Library and at the bottom, select Quantized (Int8) and press the button Build.

When the Build button is selected, a Zip file will be created and downloaded to your computer.

Main Content



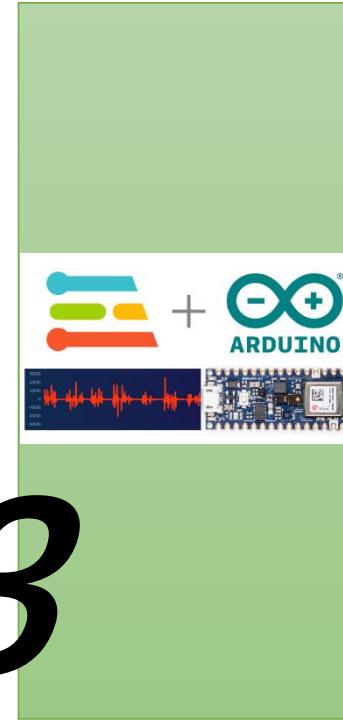
1

Edge Impulse &
XIAO ESP32S3 Sense
Introduction



2

Recording Audio
with XIAO ESP32S3
Sense



3

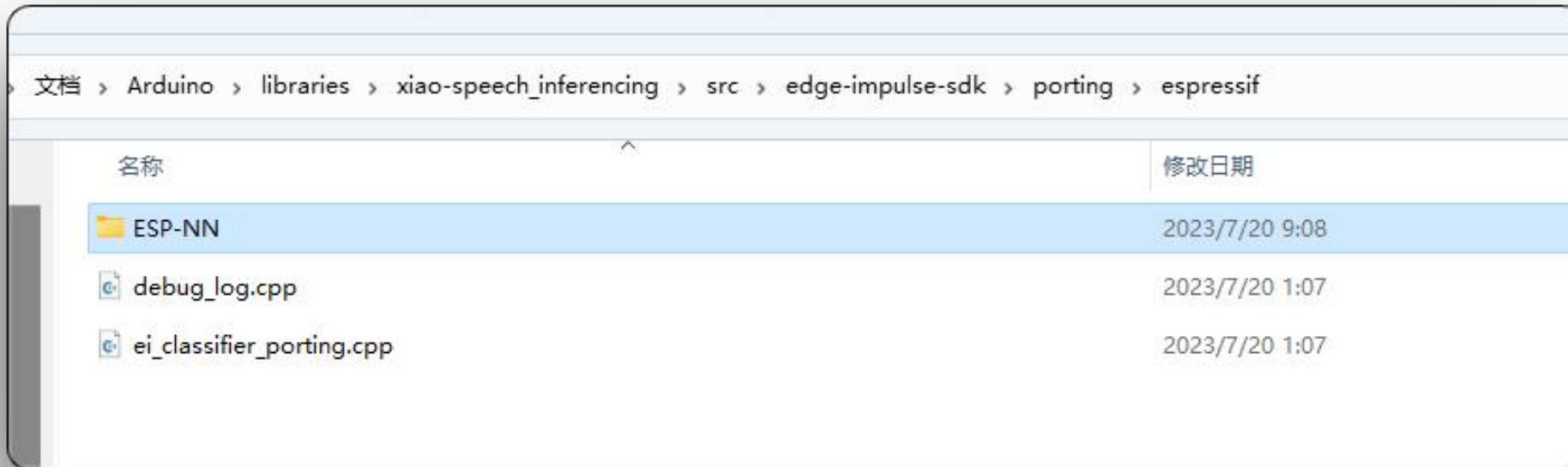
Training
Exported Models
with Edge
Impulse



4

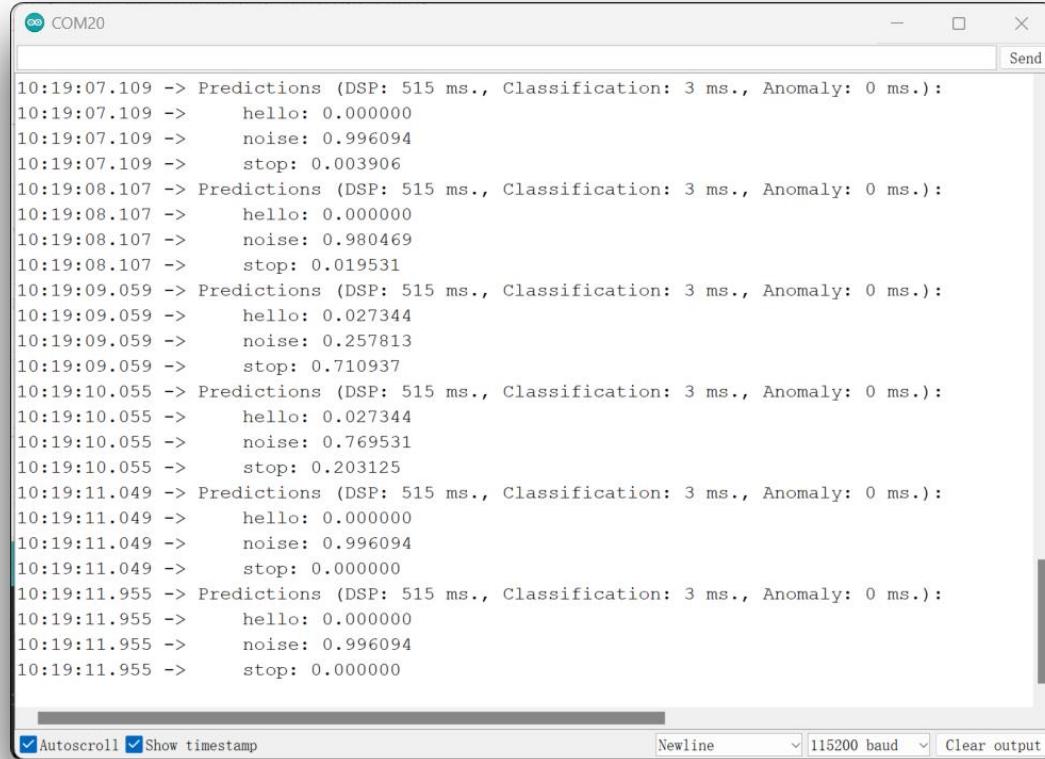
Deploying models to
XIAO ESP32S3 Sense

Deploying models to XIAO ESP32S3 Sense



Before we use the downloaded library, we need to enable the ESP NN Accelerator. For that, you can download a preliminary version from the [project GitHub](#), unzip it, and replace the ESP NN folder with it under: src/edge-impulse-sdk/porting/espressif/ESP-NN, in your Arduino library folder.

Deploying models to XIAO ESP32S3 Sense



The screenshot shows a terminal window titled "COM20" displaying a series of predictions made by a model. The output is timestamped and includes the prediction class and its confidence score. The classes are "hello", "noise", and "stop". The confidence scores range from 0.000000 to 0.996094.

```
10:19:07.109 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:07.109 ->    hello: 0.000000
10:19:07.109 ->    noise: 0.996094
10:19:07.109 ->    stop: 0.003906
10:19:08.107 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:08.107 ->    hello: 0.000000
10:19:08.107 ->    noise: 0.980469
10:19:08.107 ->    stop: 0.019531
10:19:09.059 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:09.059 ->    hello: 0.027344
10:19:09.059 ->    noise: 0.257813
10:19:09.059 ->    stop: 0.710937
10:19:10.055 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:10.055 ->    hello: 0.027344
10:19:10.055 ->    noise: 0.769531
10:19:10.055 ->    stop: 0.203125
10:19:11.049 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:11.049 ->    hello: 0.000000
10:19:11.049 ->    noise: 0.996094
10:19:11.049 ->    stop: 0.000000
10:19:11.955 -> Predictions (DSP: 515 ms., Classification: 3 ms., Anomaly: 0 ms.):
10:19:11.955 ->    hello: 0.000000
10:19:11.955 ->    noise: 0.996094
10:19:11.955 ->    stop: 0.000000
```

Autoscroll Show timestamp Newline Clear output

You can find the complete code on the [project's GitHub](#). Upload the sketch to your board and test some real inferences.

https://github.com/Mjrovai/XIAO-ESP32S3-Sense/tree/main/xiao_esp32s3_microphone_led

Deploying models to XIAO ESP32S3 Sense

```
144         // Display inference result  
145         if ((pred_index == 3) && (pred_value > 0.8)){  
146             digitalWrite(LED_BUILT_IN, LOW); //Turn on  
147         }  
148         else{  
149             digitalWrite(LED_BUILT_IN, HIGH); //Turn off  
150         }
```

Pred_index: Index of identified tags

Pred_value: Confidence level

LED_BUILT_IN: Pin numbering of on-board LED

Thanks you!