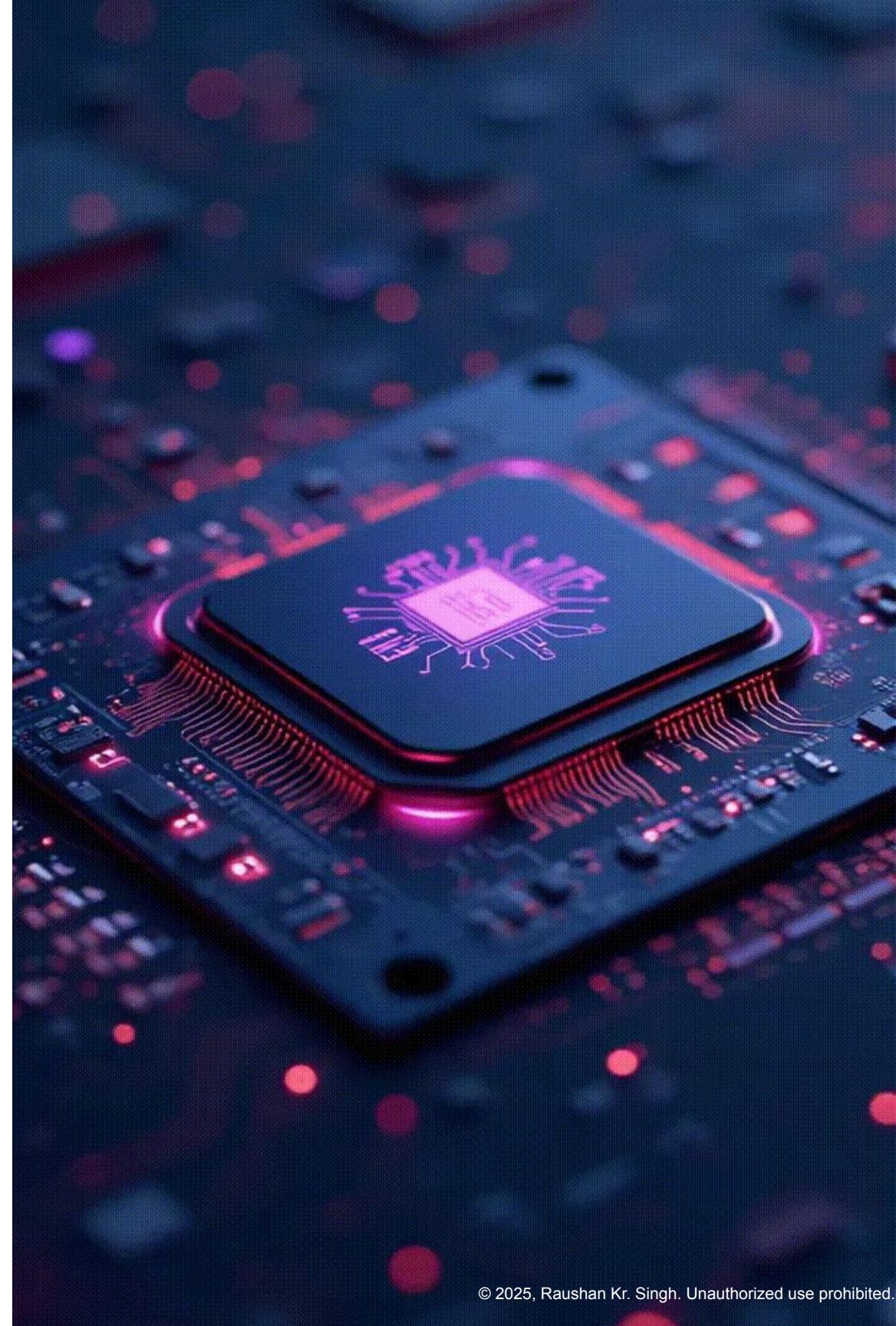


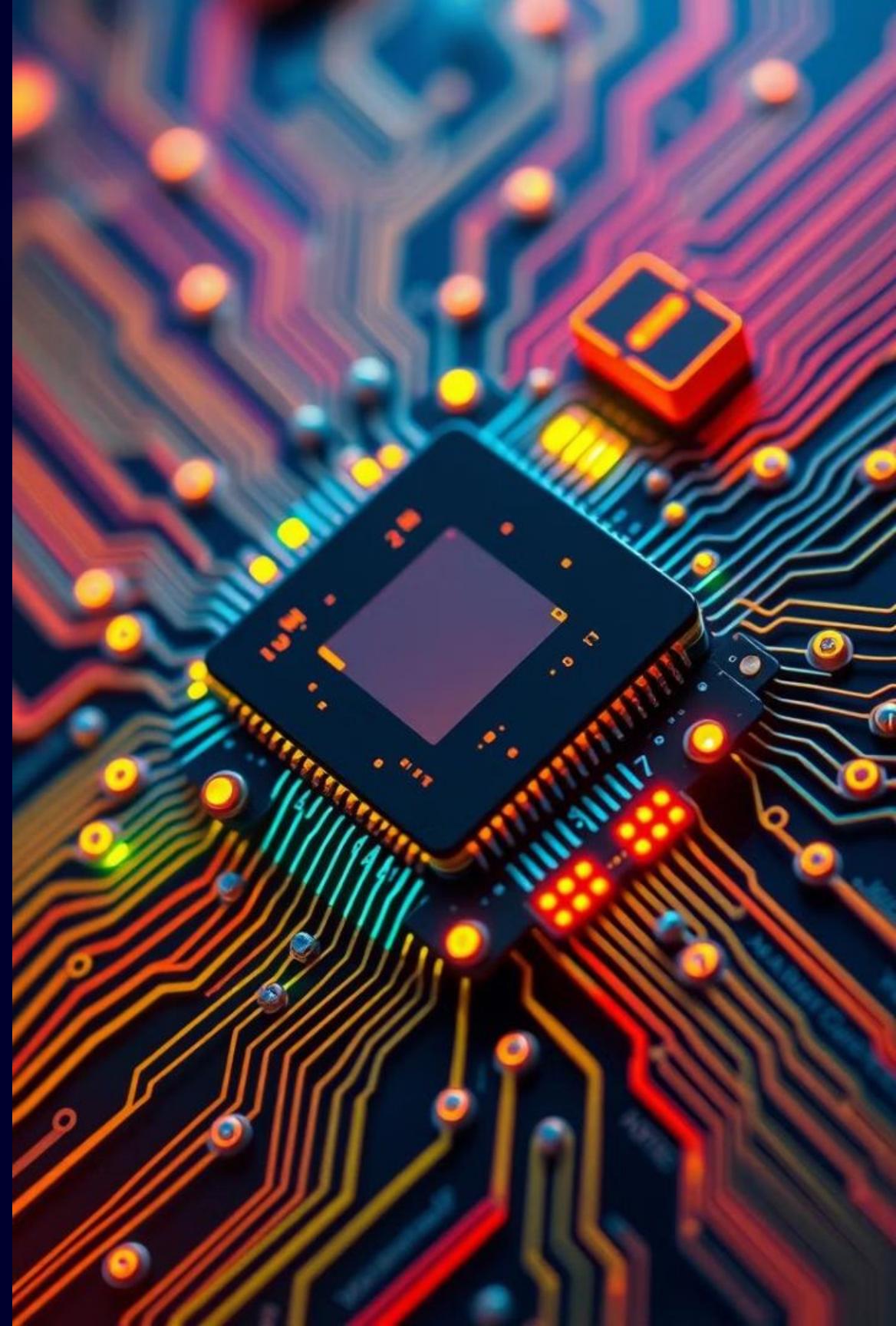
TinyML : Edge AI Bootcamp: Building Intelligent IoT with TinyML (PART A & B)

Raushan Kr. Singh
CEO, Fulectronix Technologies
IIT Ropar



Edge Impulse: Powering TinyML at the Edge

Build, train, and deploy machine learning
models directly on edge devices





Edge Impulse: The Edge AI Platform



End-to-End Development

From data collection to model deployment, Edge Impulse offers a complete development pipeline.



Device Compatibility

It targets microcontrollers, sensors, and other edge devices, with over 40 sensor integrations.



Real-World Applications

The platform powers diverse applications like predictive maintenance and human sensing.

Why Use Edge Impulse?

Edge Impulse simplifies complex machine learning development, making it accessible to a wider range of developers.

Simplified ML Development

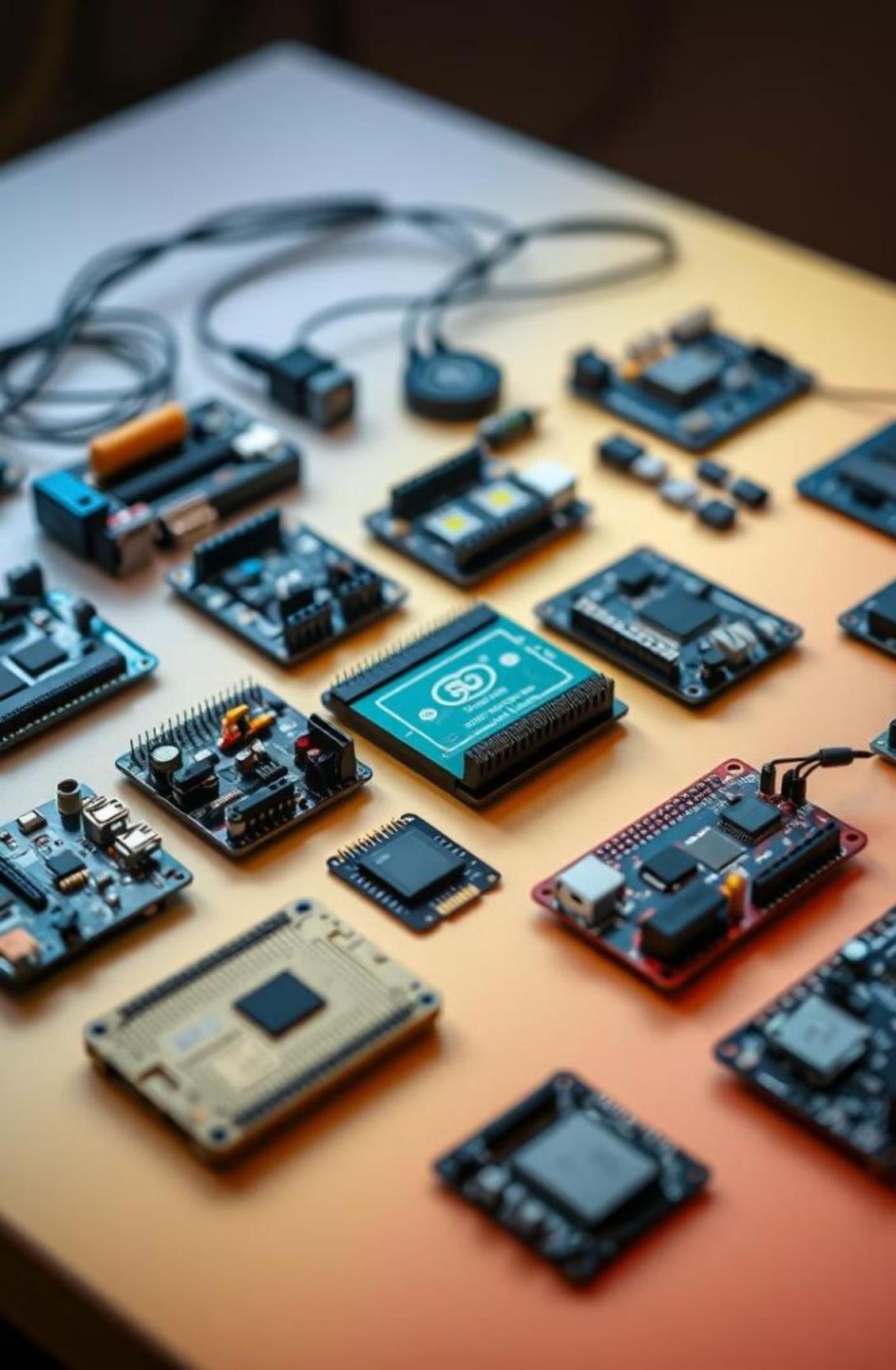
Access powerful no-code/low-code ML tools.

- Visual interface for model building.
- Pre-built machine learning blocks.

Flexible Code Generation

Generate optimized code in Python, C++, and Javascript.

- Supports various programming languages.
- Seamless integration into existing projects.



Edge Impulse Benefits

Edge Impulse offers a range of benefits for efficient and effective edge ML deployment.



Real-time Data

Collect data directly from devices in real time.



Broad Compatibility

Works with Arduino, ESP32, Raspberry Pi, and more.



Optimized Performance

Models are lightweight, fast, and function offline.



Reduced Model Size

Achieve up to an 80% reduction in model size.



Edge Impulse: Streamlined ML Development

Edge Impulse offers specialized tools for each stage of the machine learning development process.

Data Collection

Utilize the integrated data forwarder tool.

Feature Extraction

Access various signal processing blocks.

Model Training

Leverage AutoML, TensorFlow Lite, and CoreML.

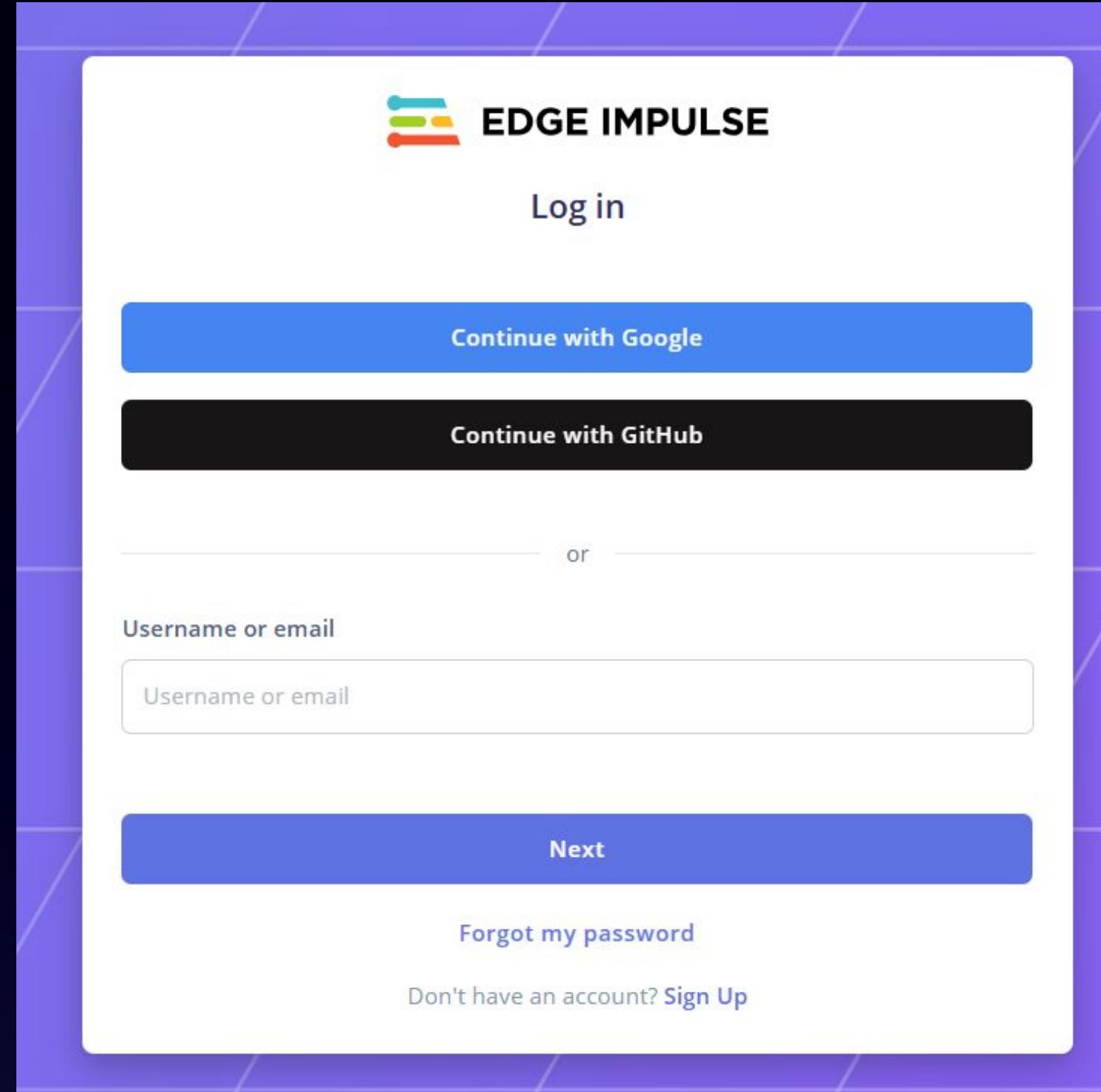
Deployment

One-click deployment to various target devices.



Simulation

Step 1: Create account on Edge Impulse Studio & Login



Step 2: Create a new project

The screenshot shows a user interface for managing projects. On the left, there is a sidebar with the following sections:

- User Profile:** A large blue circle with a white 'R' inside, labeled "Raushan".
- Role:** DEVELOPER.
- Two-Factor Authentication (MFA):** A button to "Enable MFA" with a note that it's available for all users.
- Organizations:** A section for managing shared workspaces.
- Enterprise Offer:** A button to "Try Enterprise free".

The main area is titled "Projects" and contains the following content:

- A message: "0 of 3 private projects remaining. Want access to more? Try Enterprise free."
- A "Sort" dropdown and a red-bordered "Create new project" button.
- A list of seven projects, each with a blue circular icon containing a white 'R':
 - Raushan / IoT
 - Raushan / TinyML
 - Raushan / Raushan_IoBT_Glove_Voice
 - Raushan / IoT Glove GESTURE
 - Raushan / IoT_CLASS
 - Raushan / iot_class_1
 - Raushan / defender_glove_new PUBLIC

Step 3: Enter Project Name and Select Setting

 Create a new project X

Enter the name for your new project:

Choose your project type:

Personal
60 min job limit, 4GB or 4 hours of data, limited collaboration.

Enterprise
No job or data size limits, higher performance, custom blocks.

Choose your project setting:

Public
Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users will be able to edit.

Private (0 of 3 remaining)
Only invited users can edit and view your project.
To request additional projects, [contact sales](#)

 Want full-feature access and unlimited projects? [Try Enterprise free.](#)

Create new project

Step 4: Enter Project Name and Select Setting

 Create a new project ×

Enter the name for your new project:

Choose your project type:

Personal
60 min job limit, 4GB or 4 hours of data, limited collaboration.

Enterprise
No job or data size limits, higher performance, custom blocks.

Choose your project setting:

Public
Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users will be able to edit.

Private (0 of 3 remaining)
Only invited users can edit and view your project.
To request additional projects, contact sales

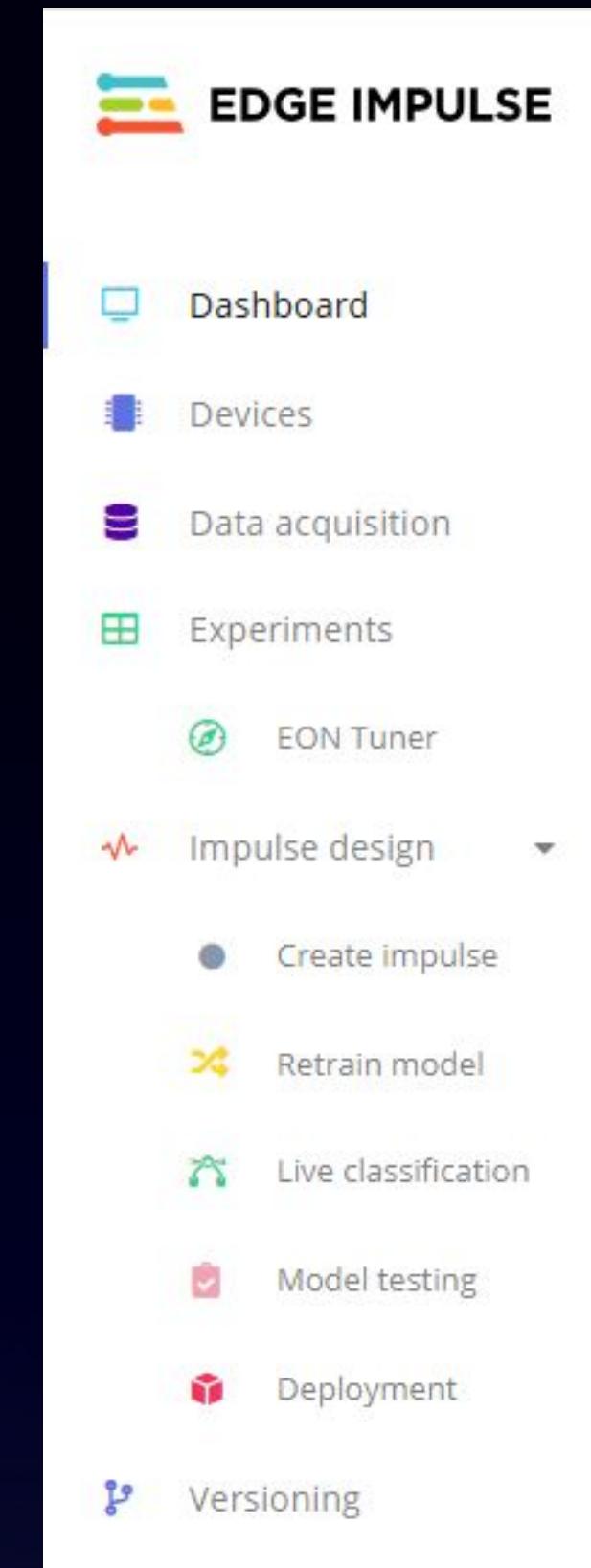
 Want full-feature access and unlimited projects? [Try Enterprise free.](#)

Create new project

Step 5: Getting Started

The screenshot shows the Edge Impulse Studio interface for a project titled "TinyML and IoT". The left sidebar, highlighted with a red border, contains a navigation menu with sections like Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment), Versioning, and GETTING STARTED. Below this is an "Upgrade Plan" section with a "View plans" button. The main content area features a "Project info" header with tabs for Project info, Keys, Export, and Jobs. The "Project info" tab is active, showing the project name "Raushan / TinyML and IoT" and a "PERSONAL" badge. A "Target: Cortex-M4F 80MHz" badge is also present. The central part of the screen displays the "TinyML and IoT" project summary, which includes a "Getting started" section with three buttons: "Add existing data" (with a database icon), "Collect new data" (with a microphone icon), and "Upload your model" (with a right-pointing arrow icon). Below this is a "Start with a tutorial" section featuring three circular icons: "Motion: Gesture recognition" (blue background with a hand icon), "Images: Object detection" (teal background with a camera icon), and "Audio: Audio classification" (orange background with a sound wave icon). The right side of the screen contains a "Sharing" section with a "Public" dropdown set to "Public", a note about internet access, a "SHARE LINK" button with the URL <https://studio.edgeimpulse.com/public/712880/live>, a "Published versions (0)" section with a "Publish a version of your project" button, and a "Collaborators (1/4)" section with a user profile for "Raushan" (OWNER).

Step 6: Getting Started



Step 7: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The left sidebar contains navigation links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment), Versioning, and an Upgrade Plan section. The main content area features a "Project info" header with tabs for Keys, Export, and jobs, and a "PERSONAL" status indicator. A "Target: Cortex-M4F 80MHz" and a blue circular "R" icon are also present. The central area displays the "TinyML and IoT" project title and a brief description: "This is your Edge Impulse project. From here you acquire new training data, design impulses and train models." Below this is a "Getting started" section with three options: "Add existing data" (highlighted with a red border), "Collect new data", and "Upload your model". A "Sharing" section indicates the project is public, with a link: <https://studio.edgeimpulse.com/public/712880/live>. The "Published versions (0)" section shows no published versions and a "Publish a version of your project" button. The "Collaborators (1/4)" section lists "Raushan OWNER". The bottom right corner includes a copyright notice: "© 2025, Raushan Kr. Singh. Unauthorized use prohibited."

Step 8: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The dashboard has a dark blue header with the project name and a "PERSONAL" badge. On the left, there's a sidebar with various navigation options like Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with Create impulse and Retrain model), Live classification, Model testing, Deployment, Versioning, and an Upgrade Plan section. The main content area features a "Getting started" section with three buttons: "Add existing data", "Collect new data" (which is highlighted with a red box), and "Upload your model". Below this, there's a "Start with a tutorial" section showing three cards: "Motion: Gesture recognition", "Images: Object detection", and "Audio: Audio classification". To the right, there's a "Sharing" section with a "Public" dropdown set to "Public", a note about internet visibility, a "SHARE LINK" button with the URL <https://studio.edgeimpulse.com/public/712880/live>, and a "Published versions (0)" section with a "Publish a version of your project" button. At the bottom, there's a "Collaborators (1/4)" section with one collaborator listed as "Raushan OWNER".

Raushan / TinyML and IoT PERSONAL

Target: Cortex-M4F 80MHz R

EDGE IMPULSE

Project info Keys Export Jobs

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

- Add existing data
- Collect new data
- Upload your model

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

- Motion: Gesture recognition
- Images: Object detection
- Audio: Audio classification

Sharing

Public

Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users can edit.

SHARE LINK

<https://studio.edgeimpulse.com/public/712880/live>

Published versions (0)

This project has no published versions.

Publish a version of your project

Collaborators (1/4)

R Raushan OWNER

Upgrade Plan

Get access to higher job limits and more collaborators.

View plans

Step 9: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The left sidebar contains navigation links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment), Versioning, and an Upgrade Plan section. The main content area features a "Getting started" section with three buttons: "Add existing data", "Collect new data", and "Upload your model" (which is highlighted with a red box). Below this is a "Start with a tutorial" section featuring three cards: "Motion: Gesture recognition", "Images: Object detection", and "Audio: Audio classification". The top right shows project details like "Target: Cortex-M4F 80MHz" and a "PERSONAL" badge. The right side includes sections for "Sharing" (Public link: <https://studio.edgeimpulse.com/public/712880/live>) and "Published versions (0)". The bottom right shows the owner information "Raushan OWNER".

Project info Keys Export jobs

Raushan / TinyML and IoT PERSONAL

Target: Cortex-M4F 80MHz R

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

- Add existing data
- Collect new data
- Upload your model

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

- Motion: Gesture recognition
- Images: Object detection
- Audio: Audio classification

Sharing

Public

Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users can edit.

SHARE LINK

<https://studio.edgeimpulse.com/public/712880/live>

Published versions (0)

This project has no published versions.

PUBLISH A VERSION

Collaborators (1/4)

R Raushan OWNER

Step 10: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The left sidebar contains navigation links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment), Versioning, and an Upgrade Plan section. The main content area features a "Project info" header with tabs for Keys, Export, and jobs, and a "PERSONAL" status indicator. A "Target: Cortex-M4F 80MHz" and a blue circular "R" icon are also present. The central part of the dashboard is titled "TinyML and IoT" and includes a sub-header: "This is your Edge Impulse project. From here you acquire new training data, design impulses and train models." Below this is a "New tag" button. The "Getting started" section provides three initial steps: "Add existing data" (with a database icon), "Collect new data" (with a camera icon), and "Upload your model" (with a gear icon). A "Sharing" panel indicates the project is public, allowing anyone on the internet to view and clone it under a 3-Clause BSD license. It includes a "SHARE LINK" button with the URL <https://studio.edgeimpulse.com/public/712880/live>. The "Published versions (0)" section shows no published versions and has a "Publish a version of your project" button. The "Collaborators (1/4)" section lists "Raushan" as the owner. The bottom right corner features a copyright notice: "© 2025, Raushan Kr. Singh. Unauthorized use prohibited."

EDGE IMPULSE

Raushan / TinyML and IoT PERSONAL

Target: Cortex-M4F 80MHz R

Project info Keys Export jobs

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

- Add existing data
- Collect new data
- Upload your model

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

- Motion: Gesture recognition
- Images: Object detection
- Audio: Audio classification

Upgrade Plan

Get access to higher job limits and more collaborators.

View plans

PERSONAL

Sharing

Public

Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users can edit.

SHARE LINK

<https://studio.edgeimpulse.com/public/712880/live>

Published versions (0)

This project has no published versions.

PUBLISH A VERSION

Collaborators (1/4)

Raushan OWNER

© 2025, Raushan Kr. Singh. Unauthorized use prohibited.

Step 11: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The left sidebar contains navigation links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment), Versioning, and an Upgrade Plan section. The main content area features a "Getting started" section with three buttons: "Add existing data", "Collect new data", and "Upload your model". Below this is a "Start with a tutorial" section with three options: "Motion: Gesture recognition" (purple circle), "Images: Object detection" (red box), and "Audio: Audio classification" (orange circle). The right sidebar includes sections for "Sharing" (Public link: <https://studio.edgeimpulse.com/public/712880/live>), "Published versions (0)", and "Collaborators (1/4)" (Raushan, OWNER). A top header bar shows the user "Raushan / TinyML and IoT PERSONAL", target "Cortex-M4F 80MHz", and a profile icon.

Project info Keys Export Jobs

Raushan / TinyML and IoT PERSONAL

Target: Cortex-M4F 80MHz

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

- Create impulse
- Retrain model
- Live classification
- Model testing
- Deployment

Versioning

Upgrading Plan

Get access to higher job limits and more collaborators.

[View plans](#)

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

Add existing data Collect new data Upload your model

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

Motion: Gesture recognition

Images: Object detection

Audio: Audio classification

Sharing

Public

Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users can edit.

SHARE LINK

<https://studio.edgeimpulse.com/public/712880/live>

Published versions (0)

This project has no published versions.

Publish a version of your project

Collaborators (1/4)

R Raushan OWNER

© 2025, Raushan Kr. Singh. Unauthorized use prohibited.

Step 12: Getting Started

The screenshot shows the Edge Impulse project dashboard for a project named "TinyML and IoT". The left sidebar includes links for Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Retrain model, Live classification, Model testing, Deployment, and Versioning), and an Upgrade Plan section. The main content area features a "Project info" header with tabs for Project info, Keys, Export, and Jobs, and a "PERSONAL" badge. It displays the target as "Cortex-M4F 80MHz" and a user icon labeled "R". The central area has a title "TinyML and IoT" and a sub-section "Getting started" with three options: "Add existing data", "Collect new data", and "Upload your model". Below this is a "Start with a tutorial" section featuring three cards: "Motion: Gesture recognition" (purple), "Images: Object detection" (teal), and "Audio: Audio classification" (orange, highlighted with a red border). The right sidebar contains sections for "Sharing" (Public link: <https://studio.edgeimpulse.com/public/712880/live>), "Published versions (0)", and "Collaborators (1/4)" (Raushan Kr. Singh, OWNER).

Project info / TinyML and IoT PERSONAL

Target: Cortex-M4F 80MHz R

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

- Create impulse
- Retrain model
- Live classification
- Model testing
- Deployment

Versioning

GETTING STARTED

Upgrade Plan

Get access to higher job limits and more collaborators.

[View plans](#)

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Add existing data

Collect new data

Upload your model

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

Motion: Gesture recognition

Images: Object detection

Audio: Audio classification

Sharing

Public

Anyone on the internet can view and clone this project under the [3-Clause BSD license](#). Only invited users can edit.

SHARE LINK

<https://studio.edgeimpulse.com/public/712880/live>

Published versions (0)

This project has no published versions.

[Publish a version of your project](#)

Collaborators (1/4)

R Raushan OWNER



Data Collection

Step 13: Collect new data and Scan QR to use Phone

Raushan / TinyML and IoT

Project info Keys Export Jobs

TinyML and IoT

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

+ New tag

Getting started

Start building your dataset or validate your model's on-device performance:

Add existing data Collect new data

Start with a tutorial

Not sure where to start? Follow a tutorial to build your first model in just minutes!

Motion: Gesture recognition Images: Object detection

Collect new data

Collect data directly from your phone, computer, device, or development board.

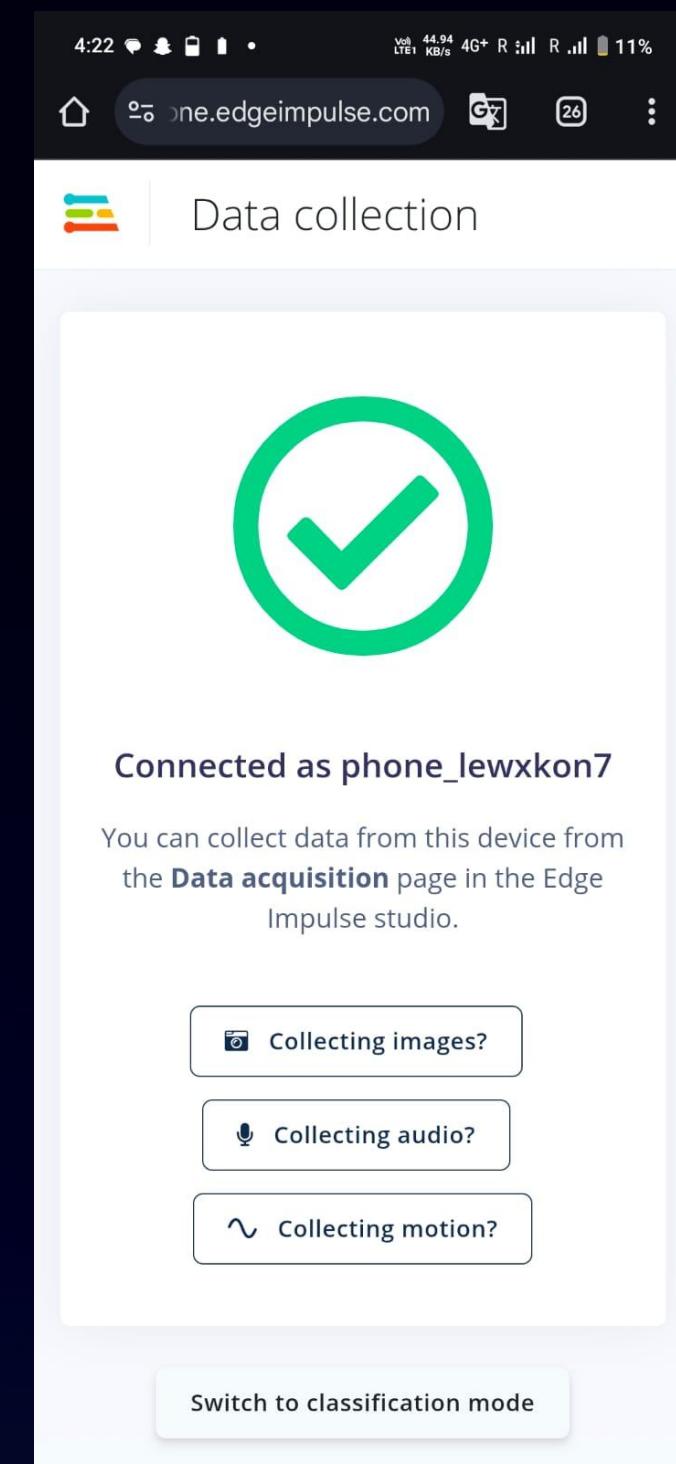
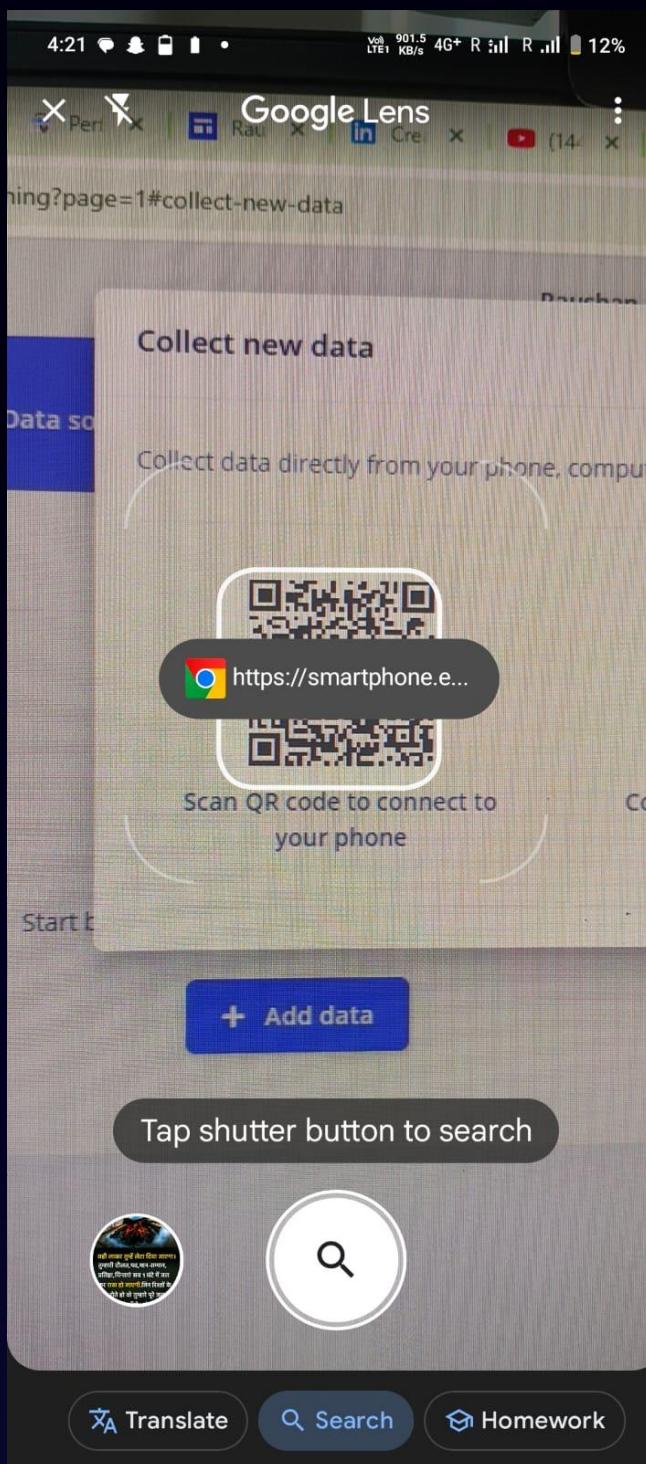
Scan QR code to connect to your phone

Connect to your computer

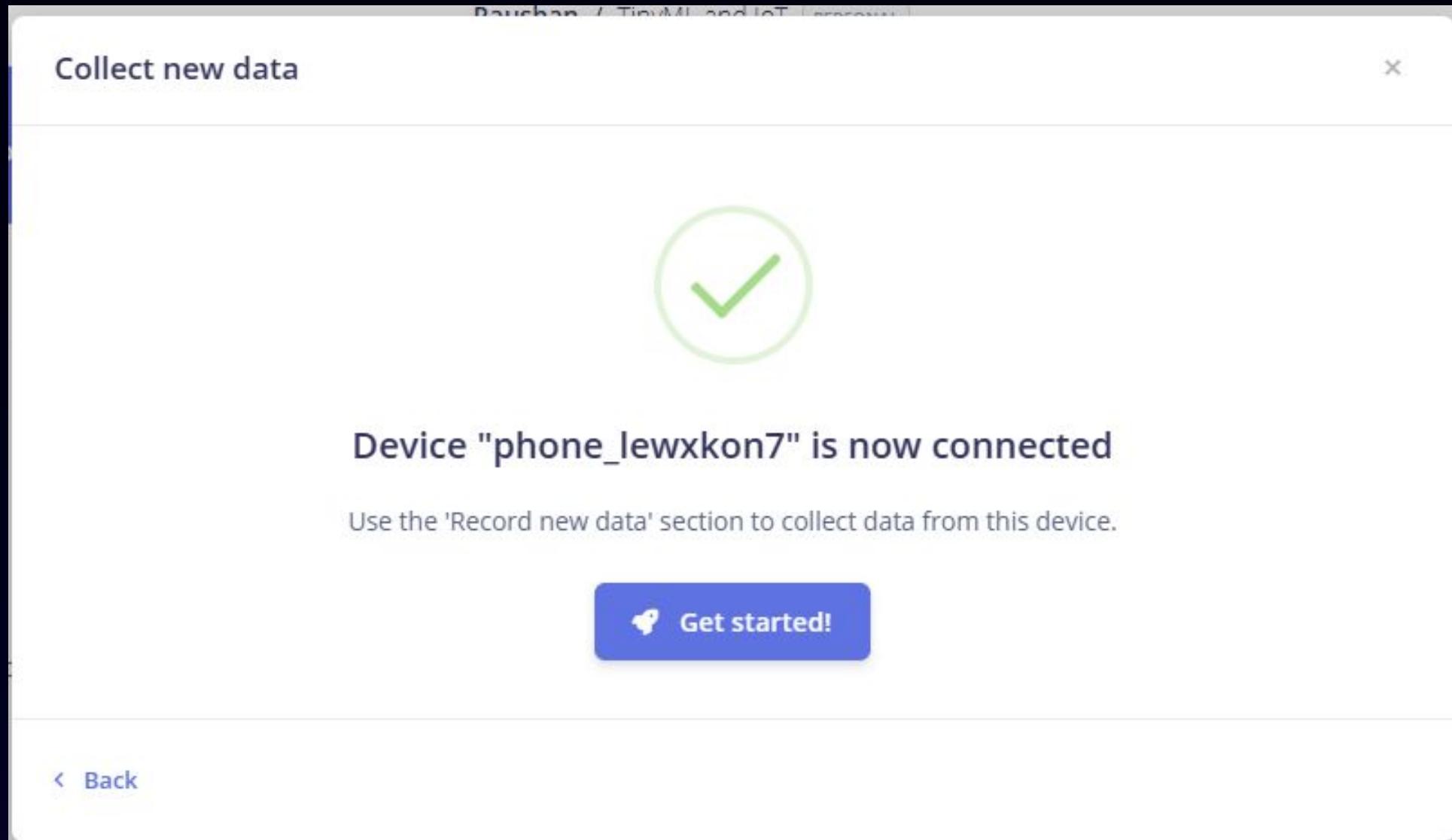
Connect your device or development board

Step 14: Phone Screen: Scan QR in Google Lens and

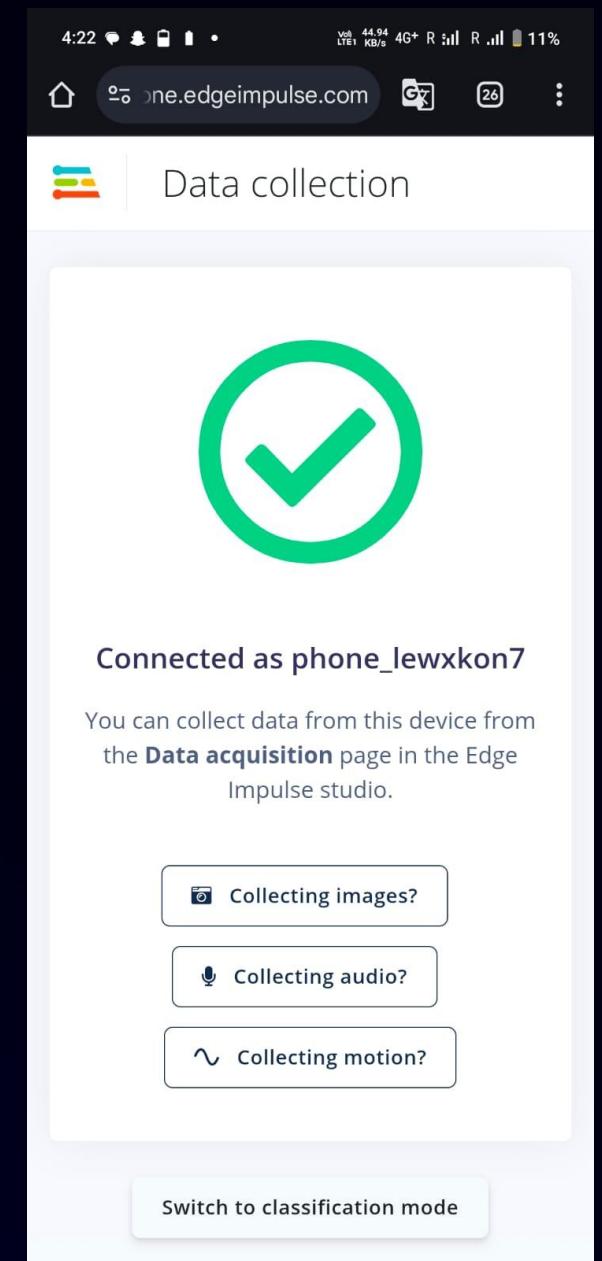
Cont.



Step 15: Create account on Edge Impulse Studio & Login

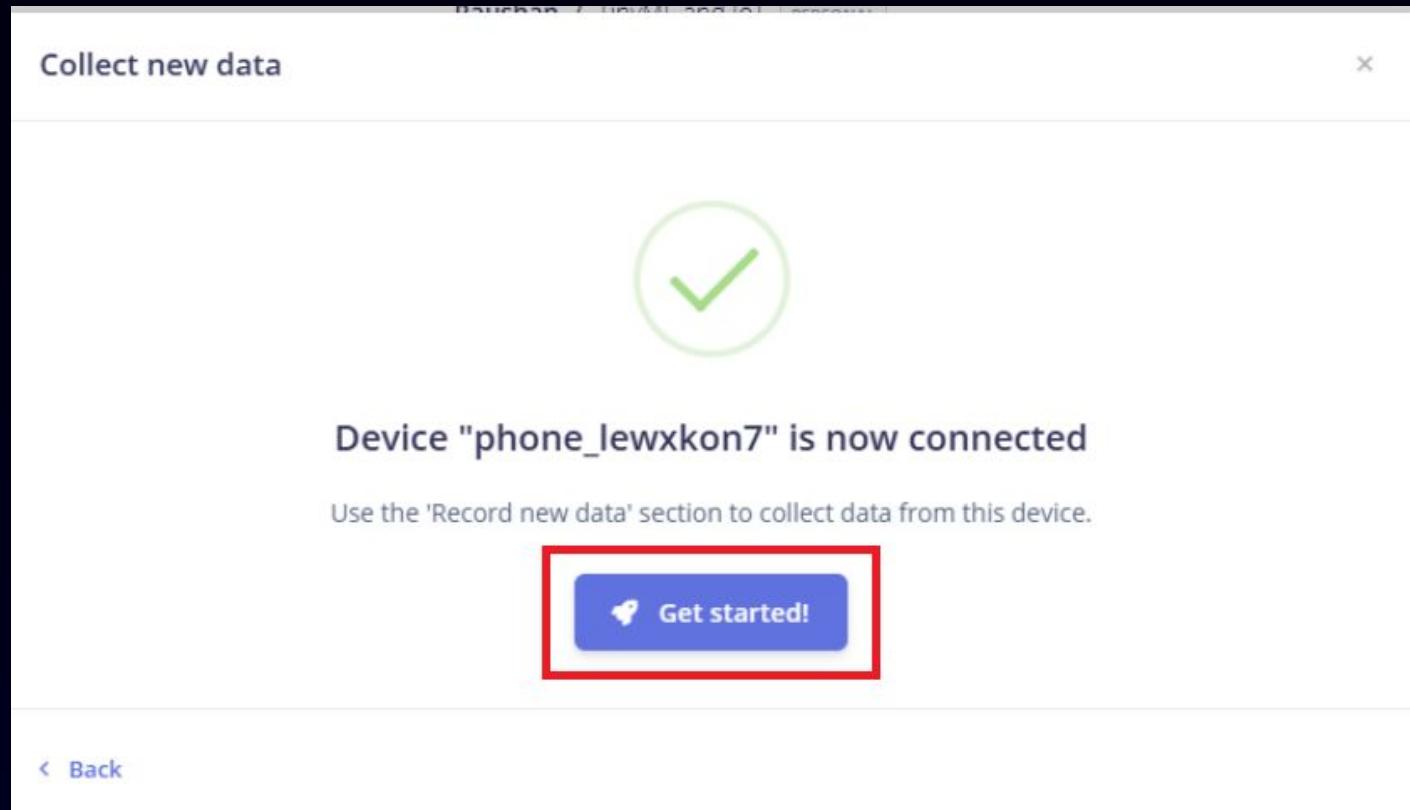


Computer Screen

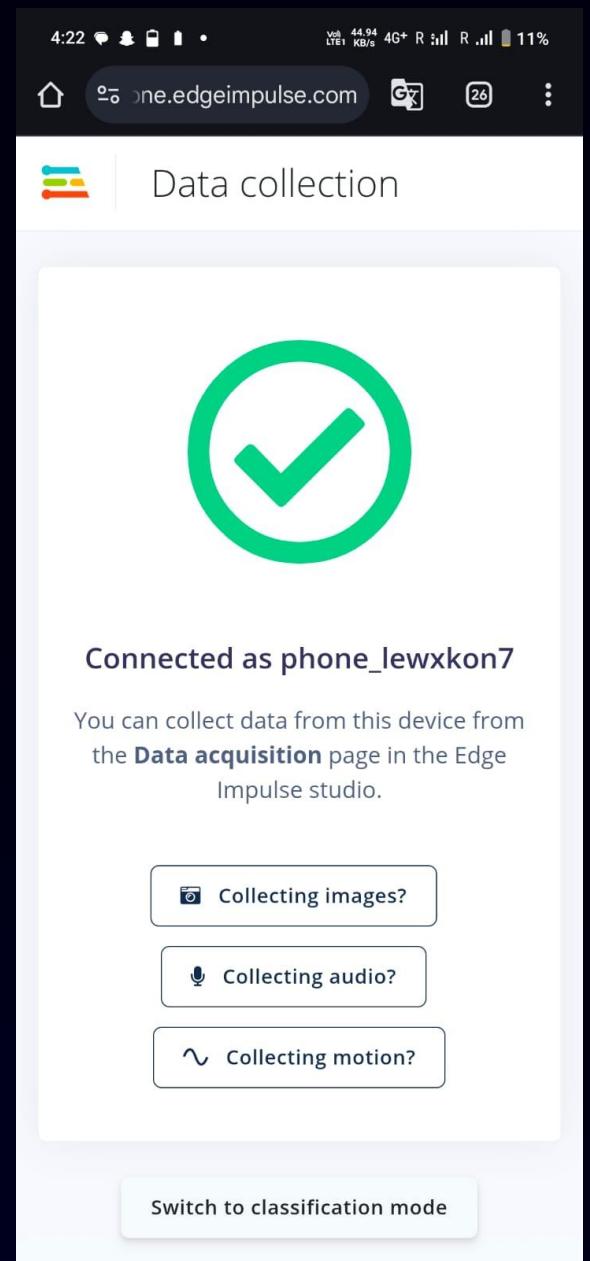


Phone Screen

Step 16: Create account on Edge Impulse Studio & Login



Computer Screen



Phone Screen

Step 17: Verify both the phone and PC Screen

Collect data

Device ⓘ
phone_lewxkon7

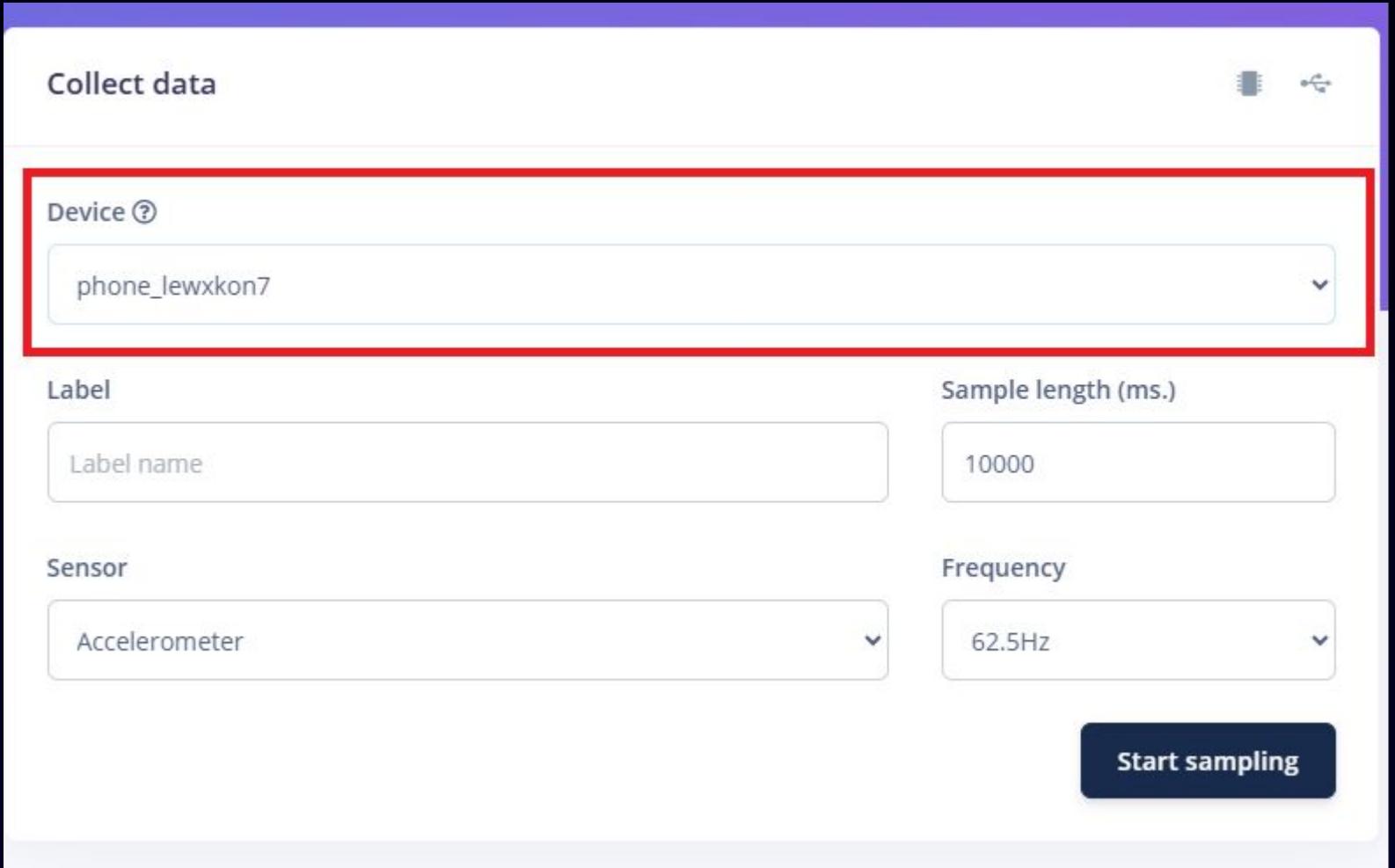
Label
Label name

Sensor
Accelerometer

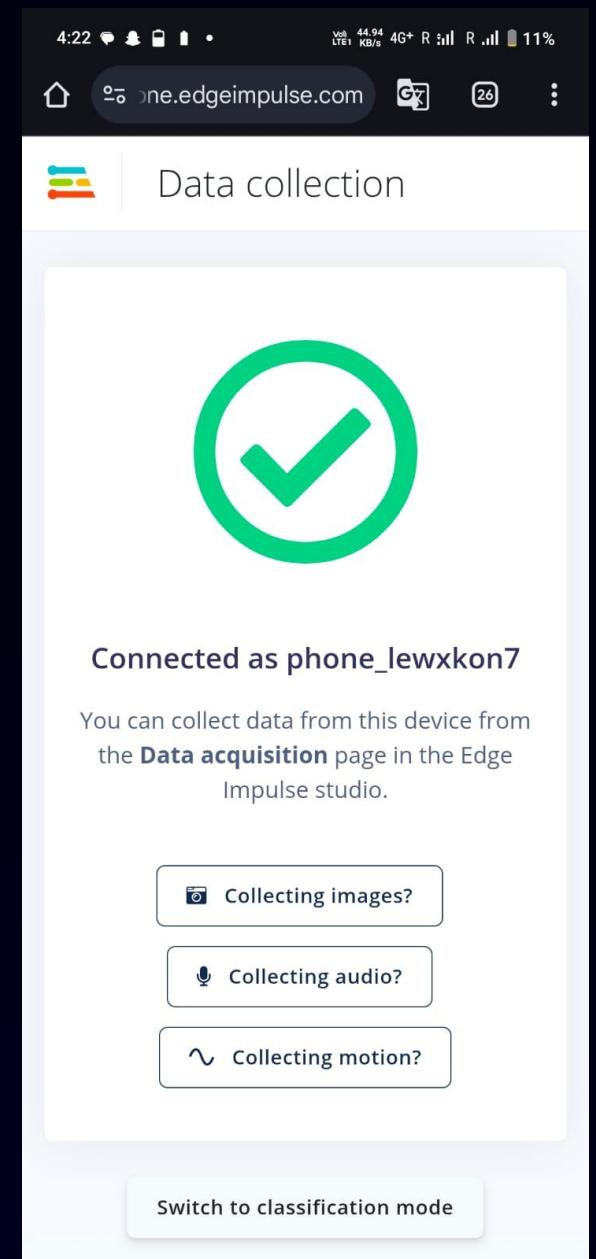
Sample length (ms.)
10000

Frequency
62.5Hz

Start sampling



Computer Screen



Phone Screen



Experiment 1: Fall Detection

Step 18: Verify both the phone and PC Screen

Collect data

Device ⓘ
phone_lewxkon7

Label
Safe

Sensor

- Positional
- Accelerometer
- Microphone
- Camera
- Positional

Sample length (ms.)
10000

Frequency
62.5Hz

Start sampling

This screenshot shows the 'Collect data' interface. It includes fields for Device (set to 'phone_lewxkon7'), Label ('Safe'), and Sensor ('Positional'). Sampling parameters are set to 10000 ms and 62.5Hz. A prominent red box highlights the 'Sensor' dropdown menu, which lists Positional, Accelerometer, Microphone, Camera, and another Positional option.

Select Sensor

Collect data

Device ⓘ
phone_lewxkon7

Label
Safe

Sample length (ms.)
10000

Frequency
62.5Hz

Start sampling

This screenshot shows the 'Collect data' interface after a sample has been labeled. The 'Label' field now contains 'Safe'. The sampling parameters remain at 10000 ms and 62.5Hz. A red box highlights the 'Label' field.

Label the Sample

Step 19: Verify both the phone and PC Screen

Collect data

Device ⓘ
phone_lewxkon7

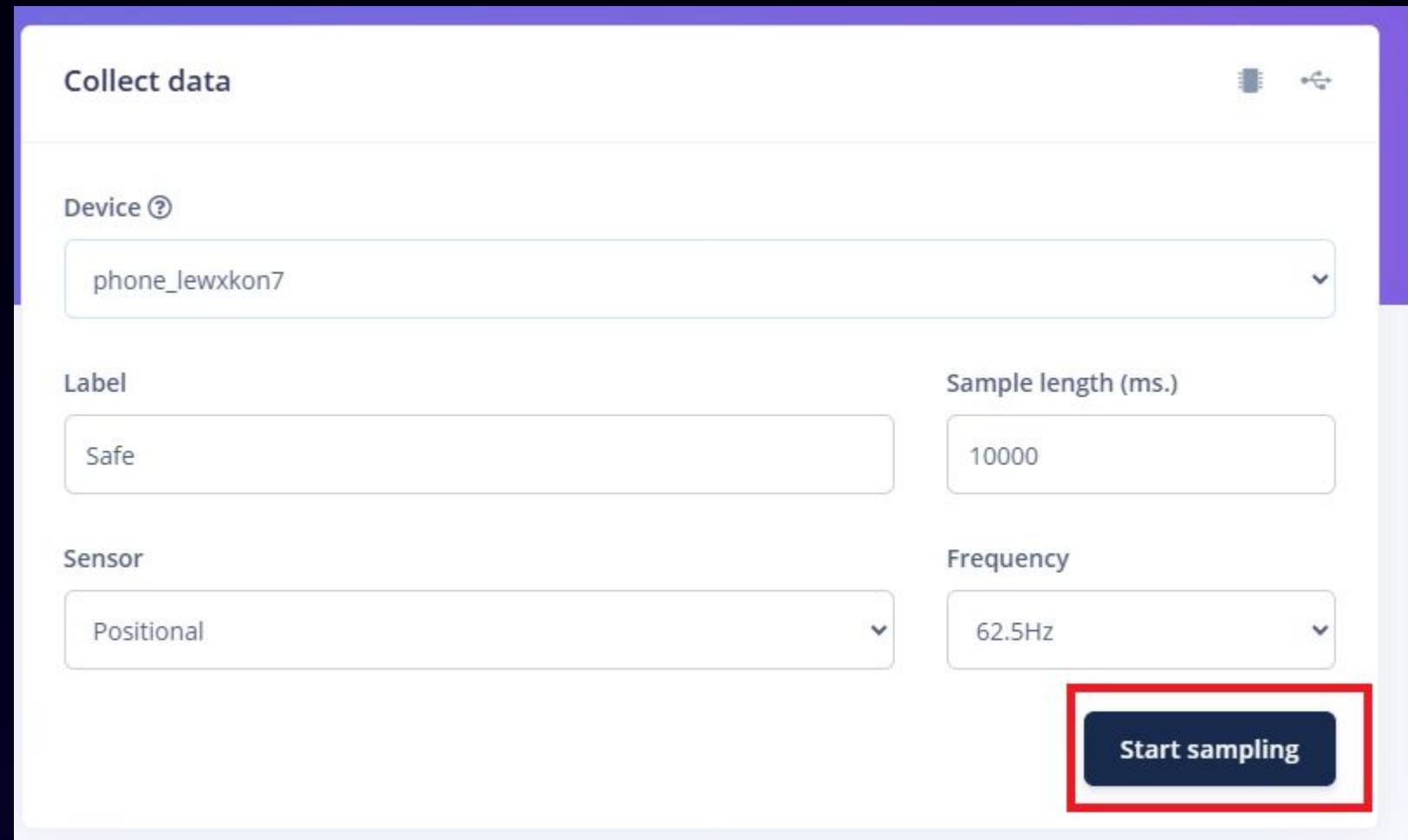
Label
Safe

Sample length (ms.)
10000

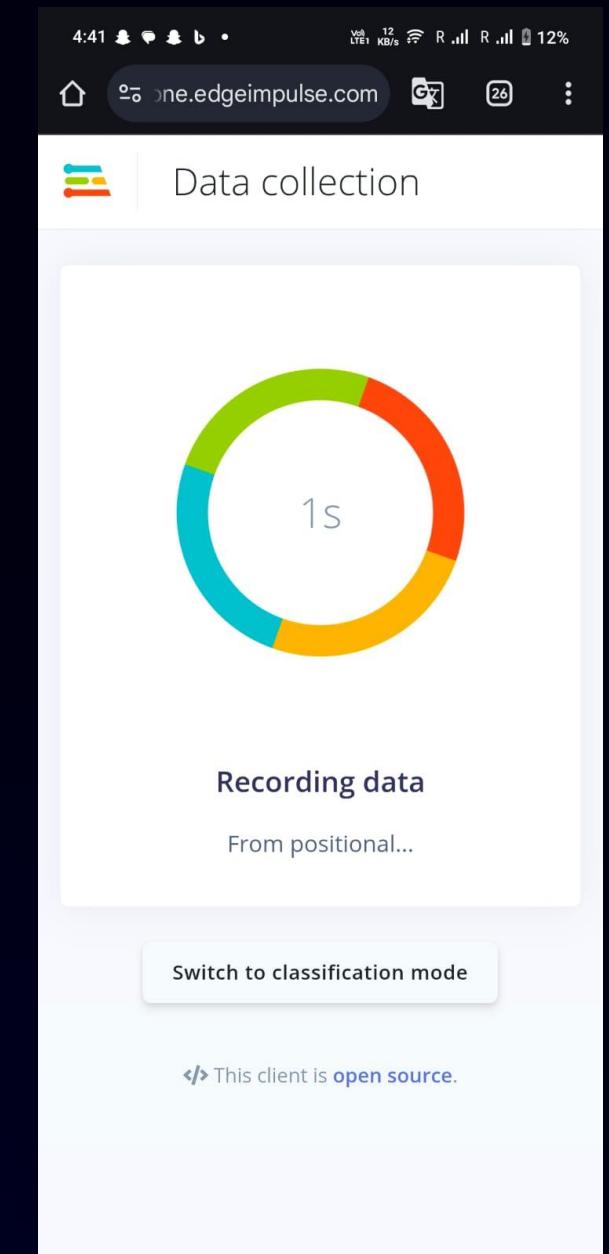
Sensor
Positional

Frequency
62.5Hz

Start sampling

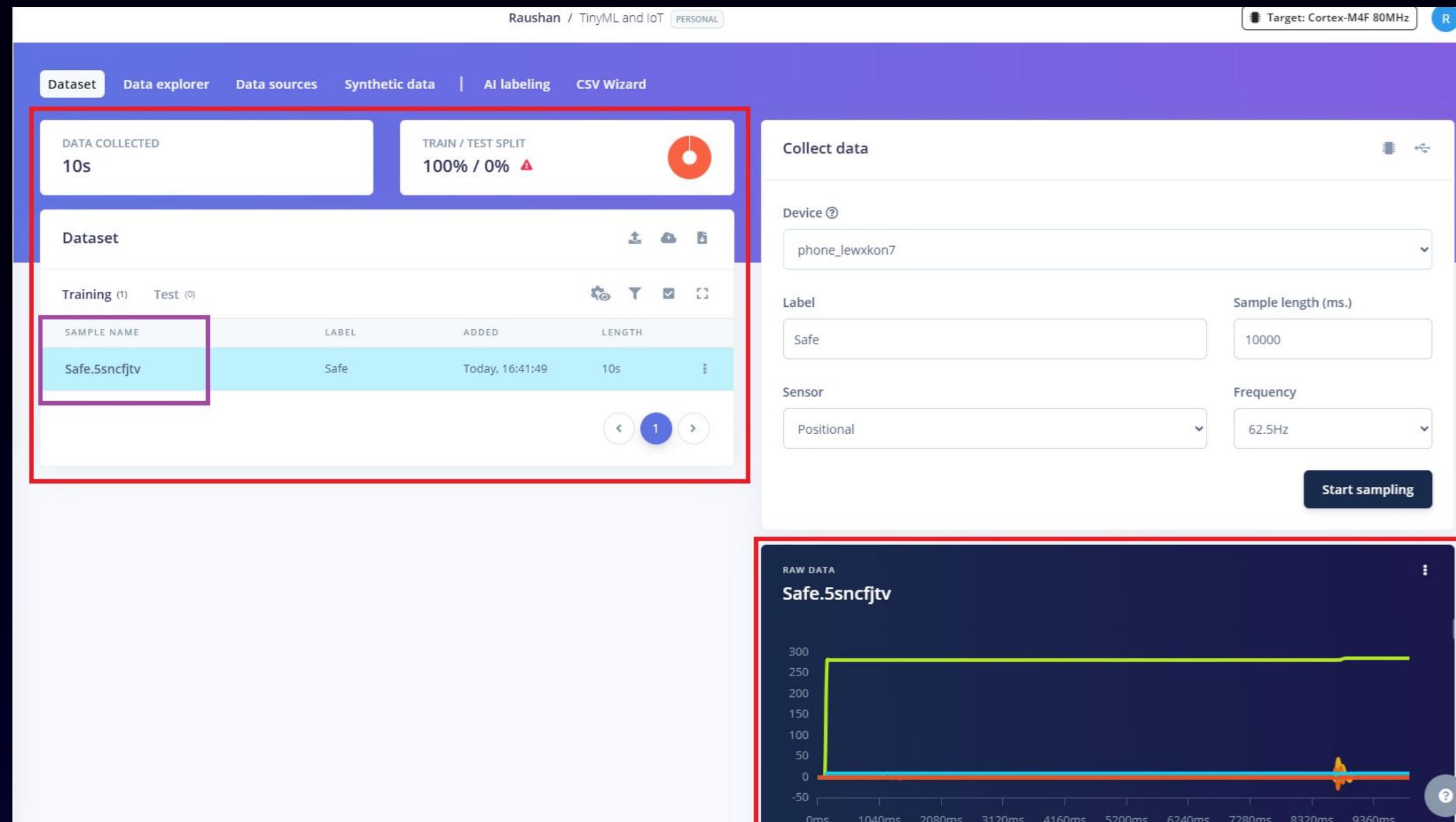


PC: Start Sampling

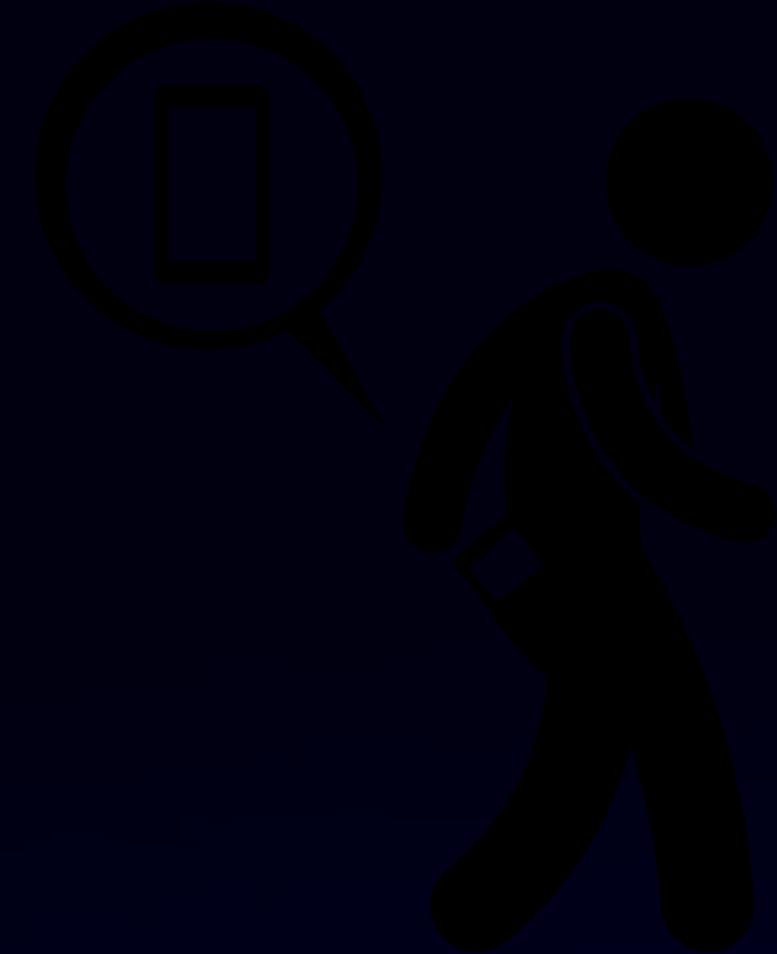


Phone: Record data

Step 20: Verify both the phone and PC Screen

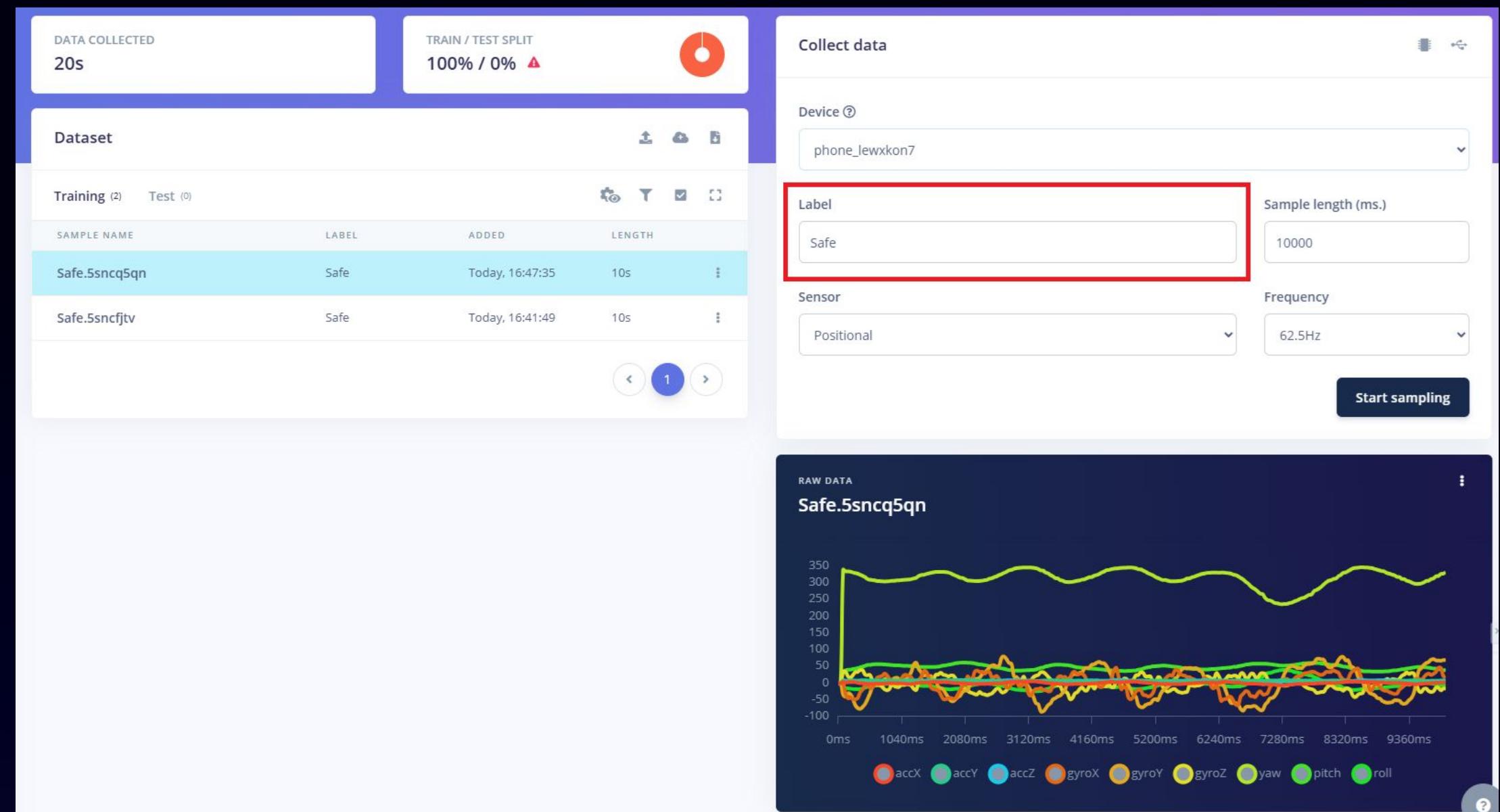


Data collection and Visualization



Labelling: Safe

Step 21: Keep phone in pocket and do normal work



Change Label to “Safe”

Step 22: Keep doing that for around 30-40 samples



Screenshot of a data collection interface showing a dataset of 20 training samples labeled "Safe". The interface includes a "Collect data" panel for setting device, label, sensor, sample length, frequency, and starting sampling. A "RAW DATA" visualization shows a signal over time for one sample.

Dataset

DATA COLLECTED
3m 20s

TRAIN / TEST SPLIT
100% / 0% ⚠

Dataset

Training (20) Test (0)

SAMPLE NAME	LABEL	ADDED	LENGTH
Safe.5sndifjm	Safe	Today, 17:00:52	10s
Safe.5sndhvr0	Safe	Today, 17:00:36	10s
Safe.5sndhe01	Safe	Today, 17:00:17	10s
Safe.5sndh06c	Safe	Today, 17:00:03	10s
Safe.5sndgb3a	Safe	Today, 16:59:42	10s
Safe.5ndfm6q	Safe	Today, 16:59:20	10s
Safe.5sndf84t	Safe	Today, 16:59:06	10s
Safe.5sndeque	Safe	Today, 16:58:52	10s
Safe.5snnddu3q	Safe	Today, 16:58:23	10s
Safe.5snndfu6	Safe	Today, 16:58:08	10s
Safe.5sndcoav	Safe	Today, 16:57:44	10s

Collect data

Device phone_lewxkon7

Label Safe

Sample length (ms.) 10000

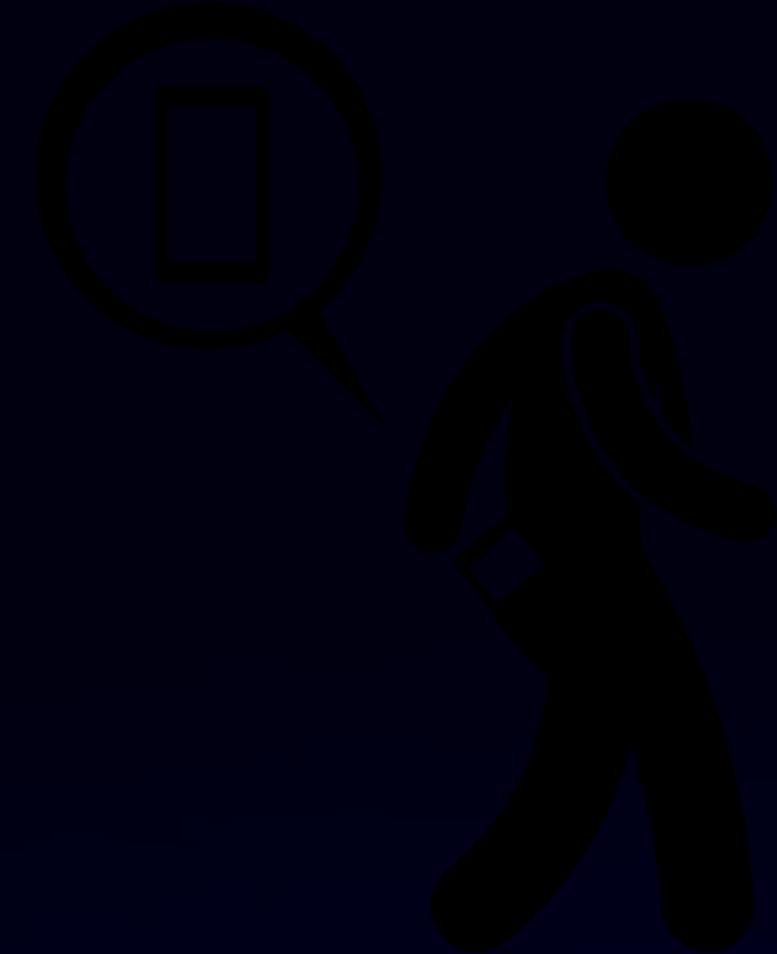
Sensor Positional

Frequency 62.5Hz

Start sampling

RAW DATA
Safe.5sndifjm

Data collection and Visualization



Labelling: Fall

Step 23: Keep collecting fall data.



The screenshot shows a dataset management interface with a purple header. The top bar includes tabs for Dataset, Data explorer, Data sources, Synthetic data, AI labeling, and CSV Wizard. Below the header, there's a summary section showing 'DATA COLLECTED' (3m 30s) and 'TRAIN / TEST SPLIT' (100% / 0%).

The main area displays a table of collected samples under the 'Dataset' tab:

SAMPLE NAME	LABEL	ADDED	LENGTH
Fall.5sne3fin	Fall	Today, 17:10:09	10s
Safe.5sndifjm	Safe	Today, 17:00:52	10s
Safe.5sndhvr0	Safe	Today, 17:00:36	10s
Safe.5sndhe01	Safe	Today, 17:00:17	10s
Safe.5sndh06c	Safe	Today, 17:00:03	10s
Safe.5sndgb3a	Safe	Today, 16:59:42	10s
Safe.5sndfm6q	Safe	Today, 16:59:20	10s
Safe.5sndf84t	Safe	Today, 16:59:06	10s
Safe.5sndeqfe	Safe	Today, 16:58:52	10s
Safe.5snddu3q	Safe	Today, 16:58:23	10s
Safe.5snddfu6	Safe	Today, 16:58:08	10s

To the right, a 'Collect data' panel is open, showing settings for a device named 'phone_lewxkon7'. It includes fields for 'Label' (set to 'Fall'), 'Sample length (ms.)' (set to 10000), 'Sensor' (set to 'Positional'), 'Frequency' (set to 62.5Hz), and a prominent 'Start sampling' button.

Below the collect panel is a 'RAW DATA' visualization for the sample 'Fall.5sne3fin'. The plot shows multiple colored lines representing sensor data over time, with a sharp vertical drop around 4160ms.

Change Label to “Fall”

Step 24: Keep doing that for around 30-40 samples



Screenshot of a dataset management interface showing a list of recorded samples and a real-time data visualization.

Dataset Overview:

- DATA COLLECTED: 6m 40s
- TRAIN / TEST SPLIT: 100% / 0% (with a warning icon)

Dataset Table:

SAMPLE NAME	LABEL	ADDED	LENGTH
Fall.5snegusa	Fall	Today, 17:17:30	10s
Fall.5sneggmq	Fall	Today, 17:17:16	10s
Fall.5sneg25d	Fall	Today, 17:17:01	10s
Fall.5snefjg5	Fall	Today, 17:16:46	10s
Fall.5snef33i	Fall	Today, 17:16:29	10s
Fall.5snedp3v	Fall	Today, 17:15:46	10s
Fall.5sned9ps	Fall	Today, 17:15:30	10s
Fall.5snecp0u	Fall	Today, 17:15:13	10s
Fall.5snec96s	Fall	Today, 17:14:57	10s
Fall.5snebnim	Fall	Today, 17:14:39	10s
Fall.5sneb8lu	Fall	Today, 17:14:24	10s

Collect data:

- Device: No devices connected
- Label: Fall
- Sample length (ms.): 10000
- Sensor: (dropdown menu)
- Frequency: (dropdown menu)

RAW DATA: Fall.5snegusa

A line graph titled "RAW DATA" for the sample "Fall.5snegusa". The x-axis represents time in milliseconds, ranging from 0ms to 9360ms with major ticks every 1040ms. The y-axis ranges from -800 to 600 with major ticks every 200 units. The graph displays multiple overlapping colored lines (red, orange, yellow, green, blue) representing different sensor signals. A sharp vertical spike in the orange signal occurs at approximately 5200ms, indicating a fall event. The background of the plot area is dark blue.

Change Label to “Fall”

Step 25: Select the Model deployment tool

The screenshot shows the TinyML and IoT platform interface. At the top, it displays "Raushan / TinyML and IoT PERSONAL" and "Target: Cortex-M4F 80MHz". The main dashboard includes sections for "DATA COLLECTED" (6m 40s), "TRAIN / TEST SPLIT" (100% / 0%), and "Collect data" (with a note to connect a device). Below this is a "Dataset" section showing "Training (40)" and "Test (0)". A modal window titled "Configure your target device and application budget" is open. It has two tabs: "Target device" and "Application budget". The "Target device" tab is active, showing a dropdown menu where "Cortex-M4F 80MHz" is selected and highlighted with a red box. The "Processor family" dropdown is set to "Cortex-M". The "Clock rate" input field is set to "80 MHz". The "Custom device name (optional)" input field is empty. The "Application budget" tab shows inputs for "RAM" (128 KB), "ROM" (1 MB), and "Latency" (100 ms). At the bottom of the modal are "Reset to default settings" and "Save" buttons. To the right of the main window, a zoomed-in view of the "Target device" dropdown is shown, displaying a list of options. The option "Seed SenseCAP A1101 (HX6537-A ARC DSP 400MHz)" is highlighted with a blue box.

Target: Cortex-M4F 80MHz

DATA COLLECTED
6m 40s

TRAIN / TEST SPLIT
100% / 0% ▲

Collect data

Connect a device to start building your dataset.

Dataset

Training (40) Test (0)

RAW DATA
Click on a sample to load...

Configure your target device and application budget

Target device

Processor family

Clock rate (MHz)

Custom device name (optional)

Application budget

RAM (KB)

ROM (MB)

Latency (ms)

Reset to default settings Save

Target device

Processor family

Clock rate (MHz)

Custom device name (optional)

Application budget

RAM (MB)

ROM (MB)

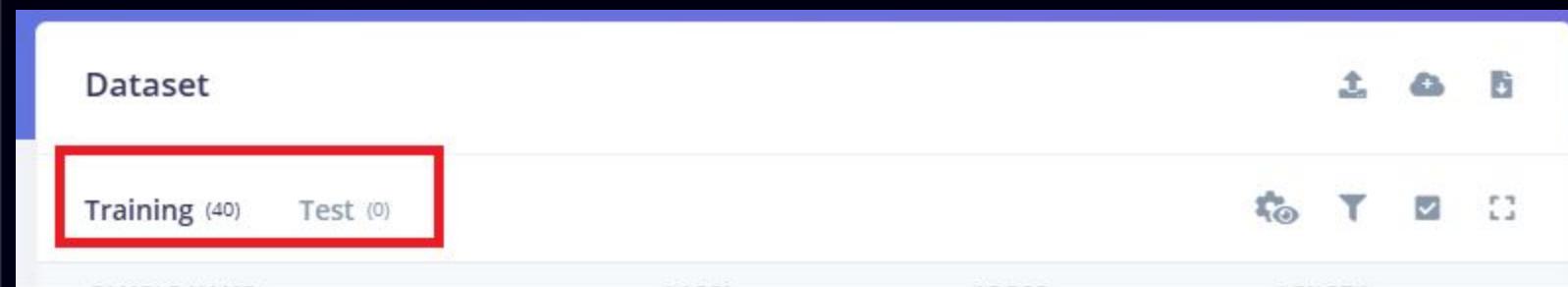
Latency (ms)

Reset to default settings Save

Seed SenseCAP A1101 (HX6537-A ARC DSP 400MHz)

© 2025, Raushan Kr. Singh. Unauthorized use prohibited.

Step 26: Divide Training and Test sample to 100/20 %



Dataset			
Training (40)	Test (0)	Filter	Check
SAMPLE NAME	LABEL	ADDED	LENGTH
Fall.5snegusa	Fall	Today, 17:17:30	10s
Fall.5sneggmq	Fall	Today, 17:17:16	10s
Fall.5sneg25d	Fall	Today, 17:17:01	10s
Fall.5snefjg5	Fall	Today, 17:16:46	10s
Fall.5snef33i	Fall	Today, 17:16:29	10s
Fall.5snedp3v	Fall	Today, 17:15:46	10s
Fall.5sned9ps	Fall	Today, 17:15:30	10s
Fall.5snecp0u	Fall	Today, 17:15:13	10s
Fall.5snec96s	Fall	Today, 17:14:57	10s
Fall.5snebnim	Fall	Today, 17:14:39	10s
Fall.5sneb8lu	Fall	Today, 17:14:24	10s

Step 27: Move 4 samples from safe & fall each to test

Dataset Data explorer Data sources Synthetic data | AI labeling CSV Wizard

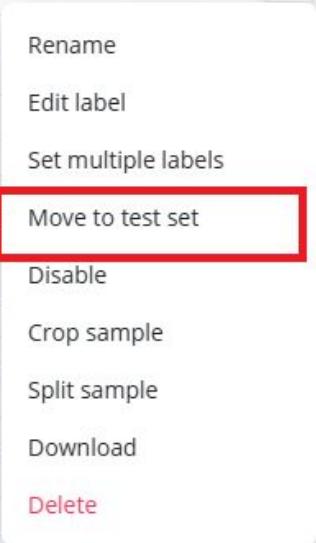
DATA COLLECTED 6m 40s 

TRAIN / TEST SPLIT 100% / 0% 

Dataset

Training (40) Test (0) 

SAMPLE NAME	LABEL	ADDED	LENGTH	
Fall.5snegusa	Fall	Today, 17:17:30	10s	
Fall.5sneggmq	Fall	Today, 17:17:16		
Fall.5sneg25d	Fall	Today, 17:17:01		
Fall.5snefjg5	Fall	Today, 17:16:46		
Fall.5snef33i	Fall	Today, 17:16:29		
Fall.5snedp3v	Fall	Today, 17:15:46		
Fall.5sned9ps	Fall	Today, 17:15:30		
Fall.5snecp0u	Fall	Today, 17:15:13		
Fall.5s nec96s	Fall	Today, 17:14:57	10s	
Fall.5snebnim	Fall	Today, 17:14:39	10s	
Fall.5sneb8lu	Fall	Today, 17:14:24	10s	


Rename
Edit label
Set multiple labels
Move to test set
Disable
Crop sample
Split sample
Download
Delete

DATA COLLECTED		TRAIN / TEST SPLIT	
6m 40s		80% / 20% ?	
			
Dataset		  	
Training (32) Test (8)			
SAMPLE NAME	LABEL	ADDED	LENGTH
Fall.5sne7ib3	Fall	Today, 17:12:23	10s
Fall.5sne732u	Fall	Today, 17:12:07	10s
Fall.5sne5vj9	Fall	Today, 17:11:31	10s
Fall.5sne3fin	Fall	Today, 17:10:09	10s
Safe.5sndifjm	Safe	Today, 17:00:52	10s
Safe.5sndhe01	Safe	Today, 17:00:17	10s
Safe.5sndf84t	Safe	Today, 16:59:06	10s
Safe.5sndeqfe	Safe	Today, 16:58:52	10s
 1 2 3 			

Step 28: Create impulse

The screenshot shows the EON Tuner interface. At the top, there is a navigation bar with the user name "Raushan / TinyML and IoT" and a "PERSONAL" button. To the right of the navigation bar is a status bar with the text "Target: Seeed SenseCAP ..." and a blue circular icon with the letter "R". Below the navigation bar, there are two tabs: "Experiments" (which is selected) and "EON Tuner". The main content area has a purple header bar with the text "Experiments (0 / 10 ⓘ)". On the right side of this bar are three icons: a funnel, a gear, and a download arrow, followed by a "Create new impulse" button with a plus sign and a blue border. The main body of the page displays the message "You have no impulses yet" and features a blue button with a white icon and the text "Create a new impulse". This "Create a new impulse" button is highlighted with a thick red rectangular border.

Step 29: Create impulse: Add processing and Learning

Impulse #1

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

Time series data



Input axes (9)

accX, accY, accZ, gyroX, gyroY, gyroZ, yaw, pitch, roll

Window size



2,000 ms.

Window increase (stride)



200 ms.

Frequency (Hz)



62.5



Zero-pad data



Add a processing block



Add a learning block

Output features



Save Impulse



Step 30: Create impulse: Add processing and Learning

Impulse #1

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

Time series data

Input axes (9)
accX, accY, accZ, gyroX, gyroY, gyroZ, yaw, pitch, roll

Window size
2,000 ms.

Window increase (stride)
200 ms.

Frequency (Hz)
62.5

Zero-pad data

Spectral Analysis

Name
Spectral features

Input axes (9)

- accX
- accY
- accZ
- gyroX
- gyroY
- gyroZ
- yaw
- pitch
- roll

Classification

Name
Classifier

Input features

- Spectral features

Output features
2 (Fall, Safe)

Output features

2 (Fall, Safe)

Save Impulse

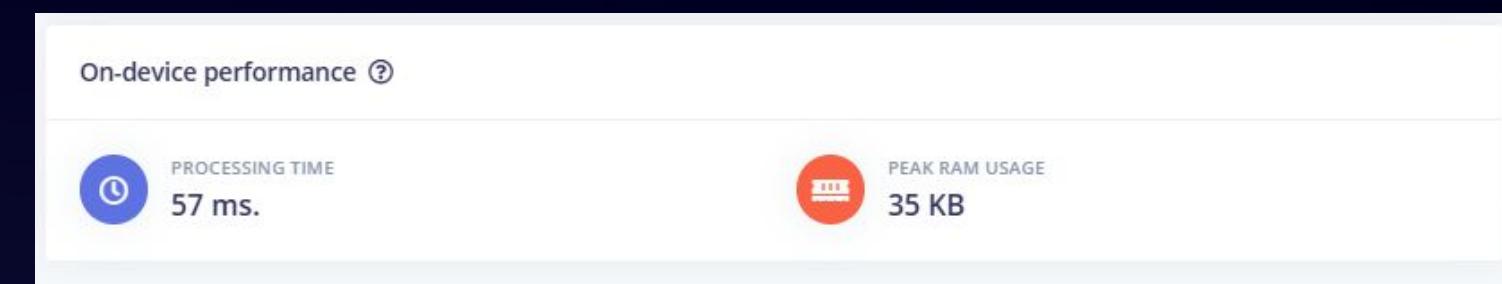
Add a learning block

Step 31: Spectral Features: Autotune and Save

Parameters

The screenshot shows the Spectral Features interface with the following components:

- Raw data:** A line graph showing multiple signals over time (0ms to 9728ms). The signals include accX, accY, accZ, gyroX, gyroY, gyroZ, yaw, pitch, and roll.
- Raw features:** A table showing raw feature values for the "Fall" label. The values are all 0.0000, except for the last column which is 279.0.
- Parameters:** Configuration settings for the analysis:
 - Scale axes: 1
 - Input decimation ratio: 1
 - Type: none
 - Analysis:
 - Type: FFT
 - FFT length: 16
 - Take log of spectrum?: checked
 - Overlap FFT frames?: checked
 - Improve low frequency resolution?: unchecked
- Autotune output:** Shows the progress of an autotuning job.
 - Creating job... OK (ID: 34000728)
 - Job scheduled at 03 Jun 2025 16:15:26
 - Job started at 03 Jun 2025 16:15:34
- DSP result:** A line graph showing spectral power (log) over sample numbers (0.00 to 1872.00). The graph shows several colored lines representing different frequency components.
- Save parameters:** A button highlighted with a red box.



Step 32: Generate Feature

The screenshot shows a user interface for generating features from a training set. The top navigation bar has tabs for 'Parameters' and 'Generate features', with 'Generate features' being active. The left panel, titled 'Training set', displays the following data:

Data in training set	5m 20s
Classes	2 (Fall, Safe)
Training windows	1,344
Calculate feature importance	<input type="checkbox"/>

Below this is a large blue button labeled 'Generate features' with a red rectangular border around it. The right panel, titled 'Feature explorer', shows the message: 'No features generated yet.'

At the bottom, there is a section titled 'Feature generation output' with a small icon and '(0)' next to it.

Step 33: Generate Feature

Parameters **Generate features**

Training set

Data in training set	5m 20s
Classes	2 (Fall, Safe)
Training windows	1,344
Calculate feature importance	<input checked="" type="checkbox"/>

Feature explorer

Legend: ● Fall ● Safe

Feature generation output

Generating features... (0)

[785/1344] Creating features...
[842/1344] Creating features...
[898/1344] Creating features...
[955/1344] Creating features...
[1011/1344] Creating features...
[1068/1344] Creating features...
[1124/1344] Creating features...
[1181/1344] Creating features...
[1237/1344] Creating features...
[1294/1344] Creating features...
[1344/1344] Creating features...
Created features

Job completed (success)

On-device performance ⓘ

PROCESSING TIME
33 ms.

PEAK RAM USAGE
5 KB

Step 34: Classifier

Neural Network settings

Training settings

Number of training cycles ② 30

Use learned optimizer ②

Learning rate ② 0.0005

Training processor ② CPU

Advanced training settings

Neural network architecture

Input layer (4,653 features)

Dense layer (20 neurons)

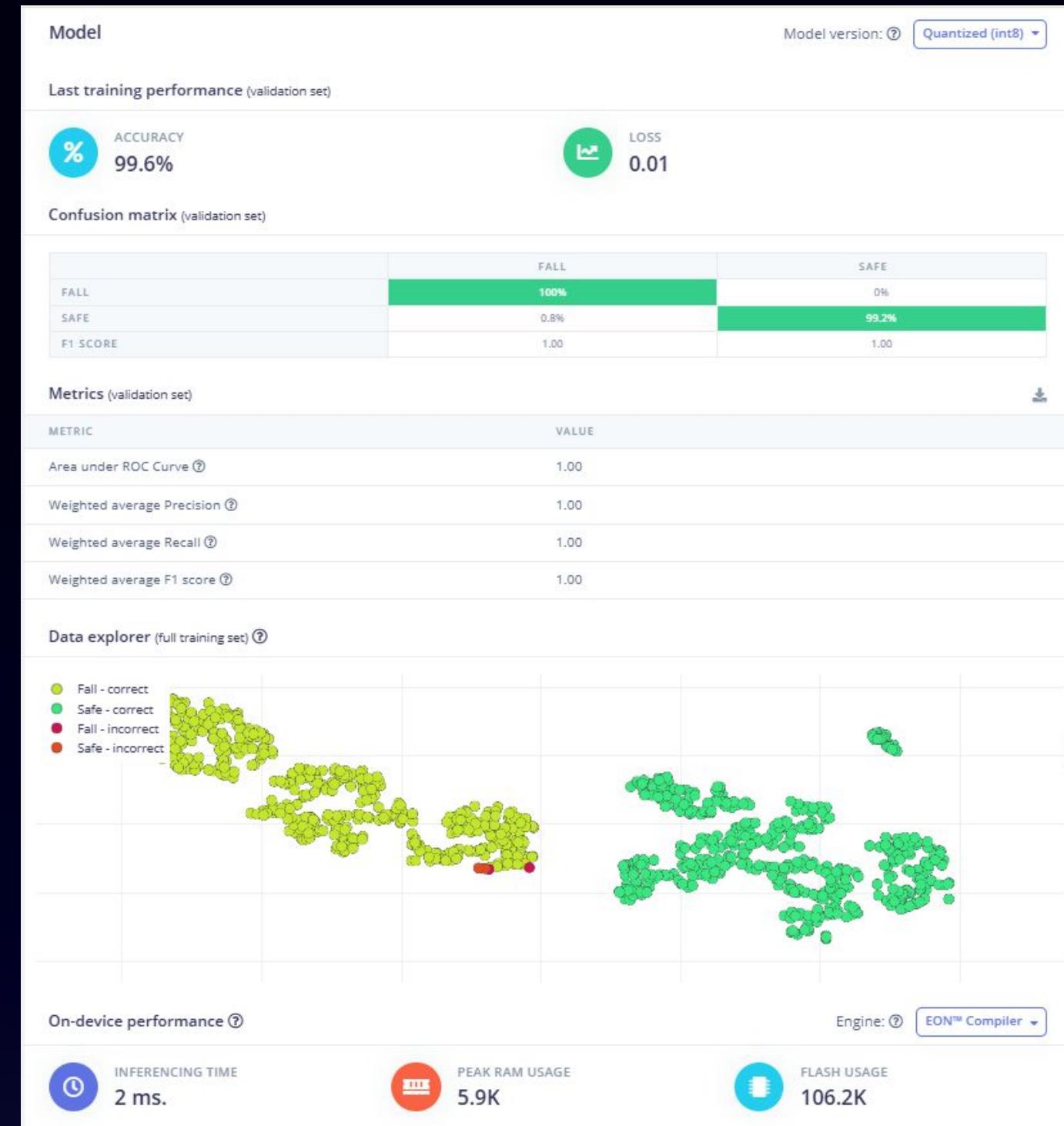
Dense layer (10 neurons)

Add an extra layer

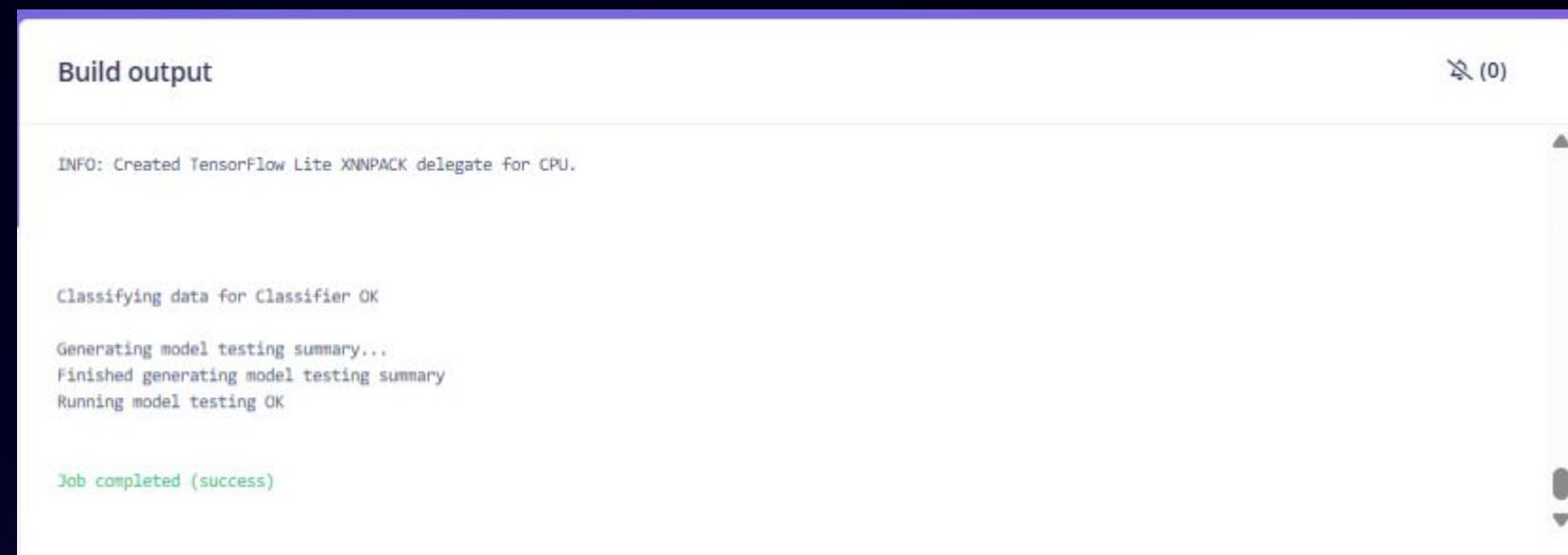
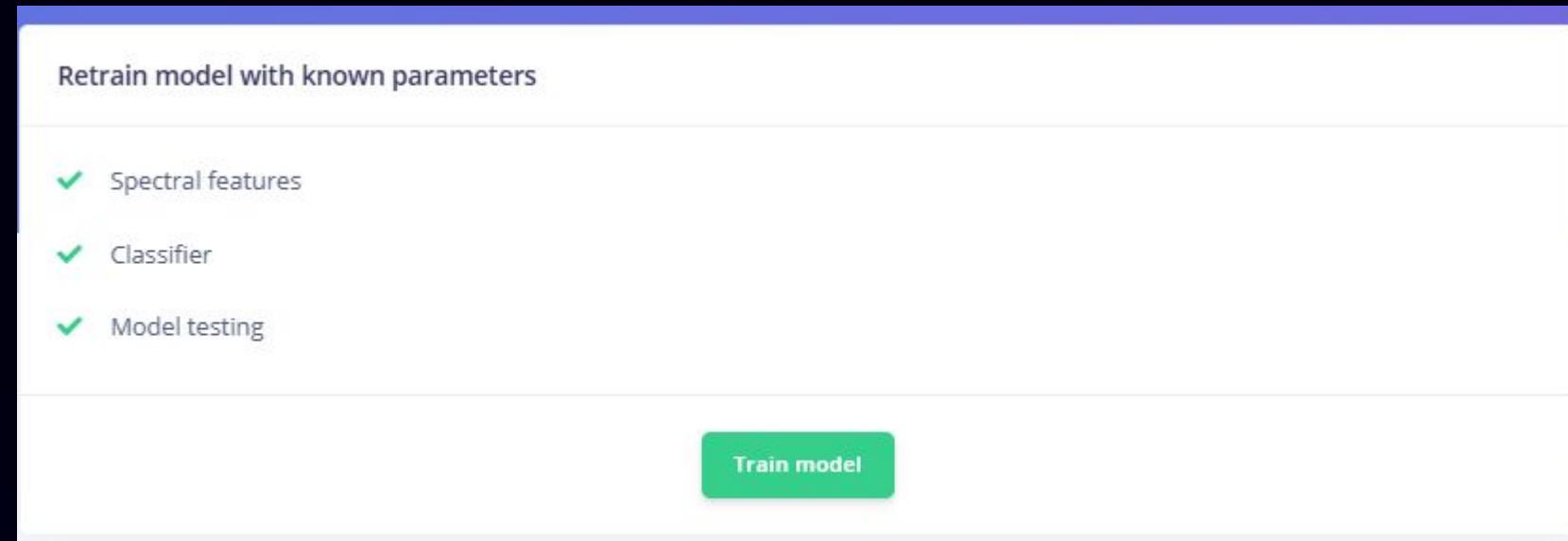
Output layer (2 classes)

Save & train ▼

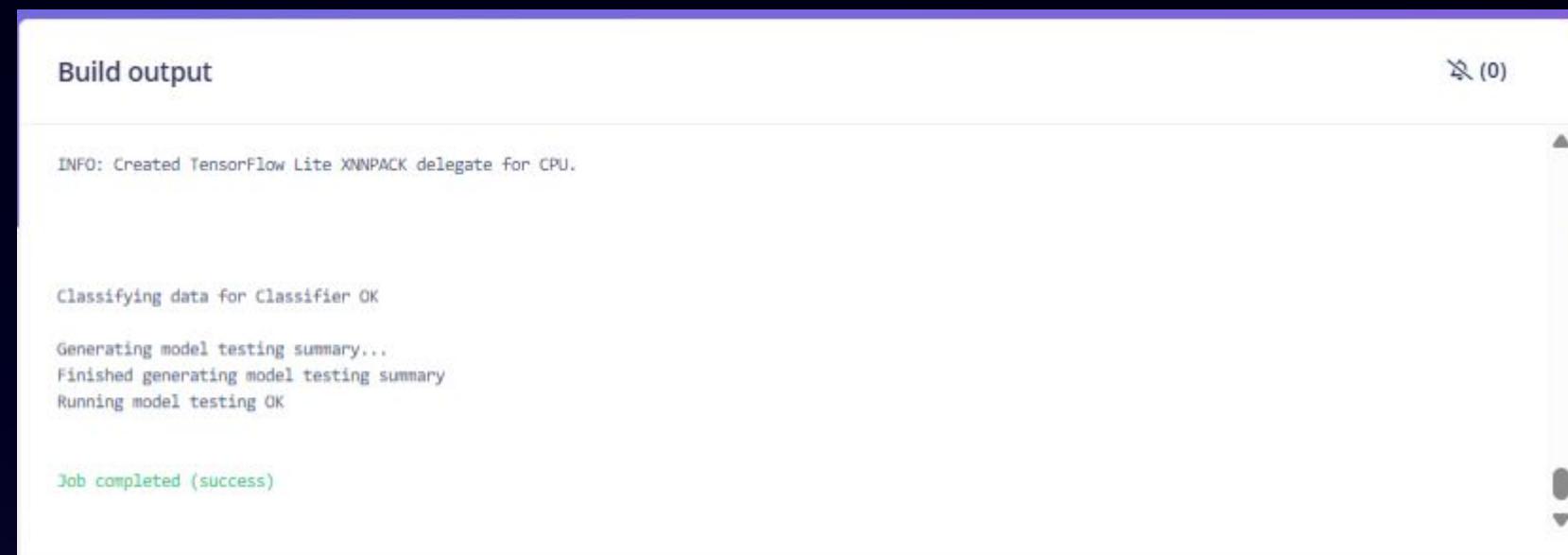
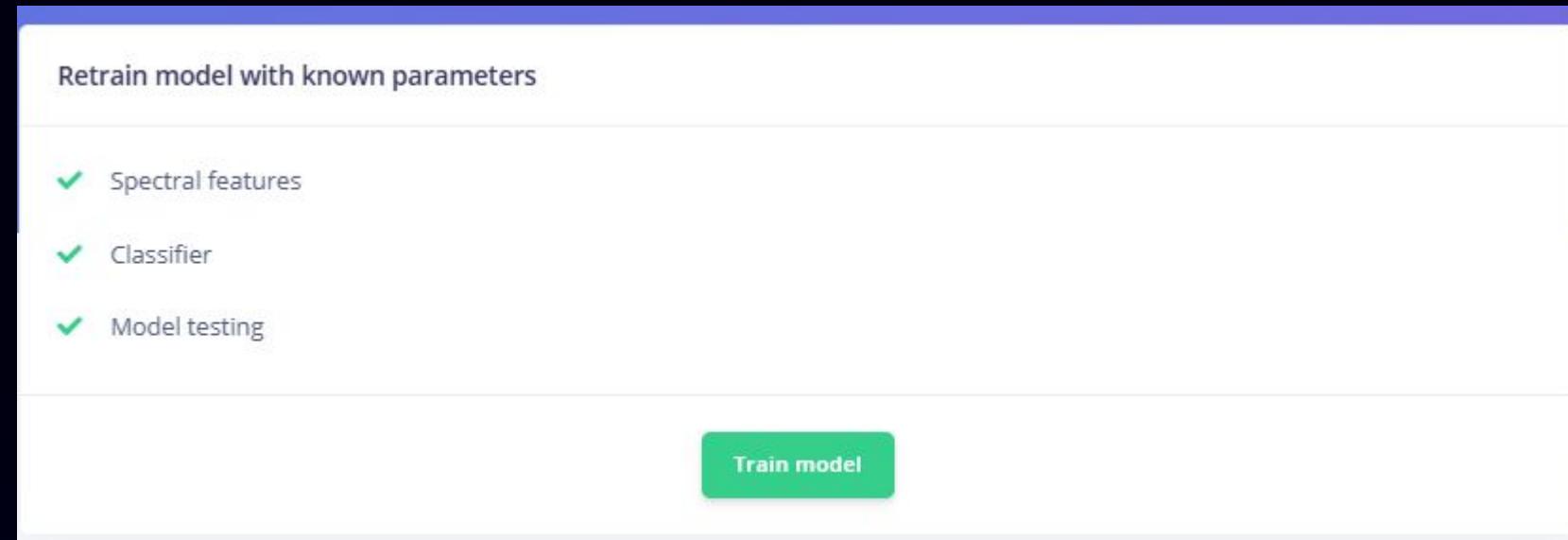
Step 35: Model



Step 36: Retrain Model with known parameters



Step 37: Retrain Model with known parameters



Step 38: Live Classification

Classify new data

Device

Sensor

Sample length (ms.)

Frequency

Start sampling



Classify existing test sample

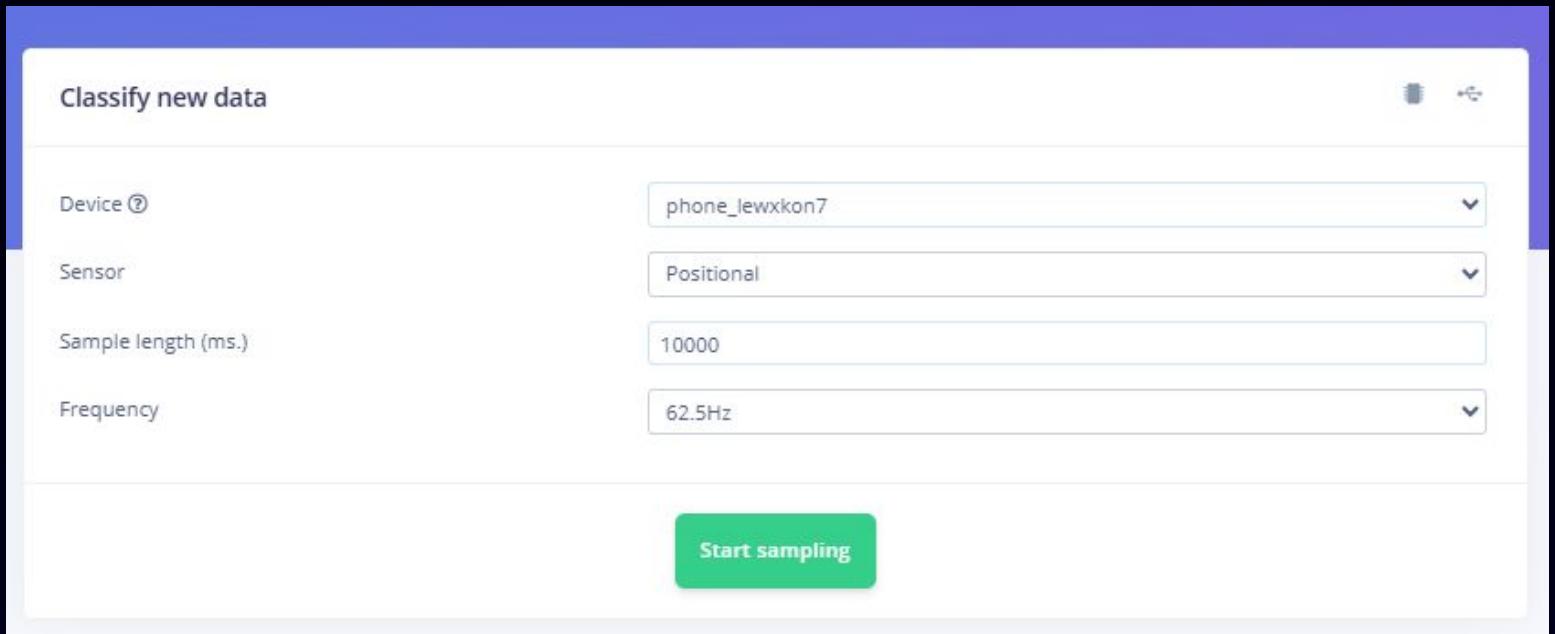
- testing.5sv9s7jl (testing)
- testing.5sv9s7jl (testing)**
- testing.5sv9ql0e (testing)
- Fall.5snegusa (Fall)
- Fall.5sneggmq (Fall)
- Fall.5sneg25d (Fall)
- Fall.5snefjg5 (Fall)
- Safe.5sndhvr0 (Safe)
- Safe.5sndh06c (Safe)
- Safe.5sndgb3a (Safe)
- Safe.5sndfm6q (Safe)

Phone Should be connected

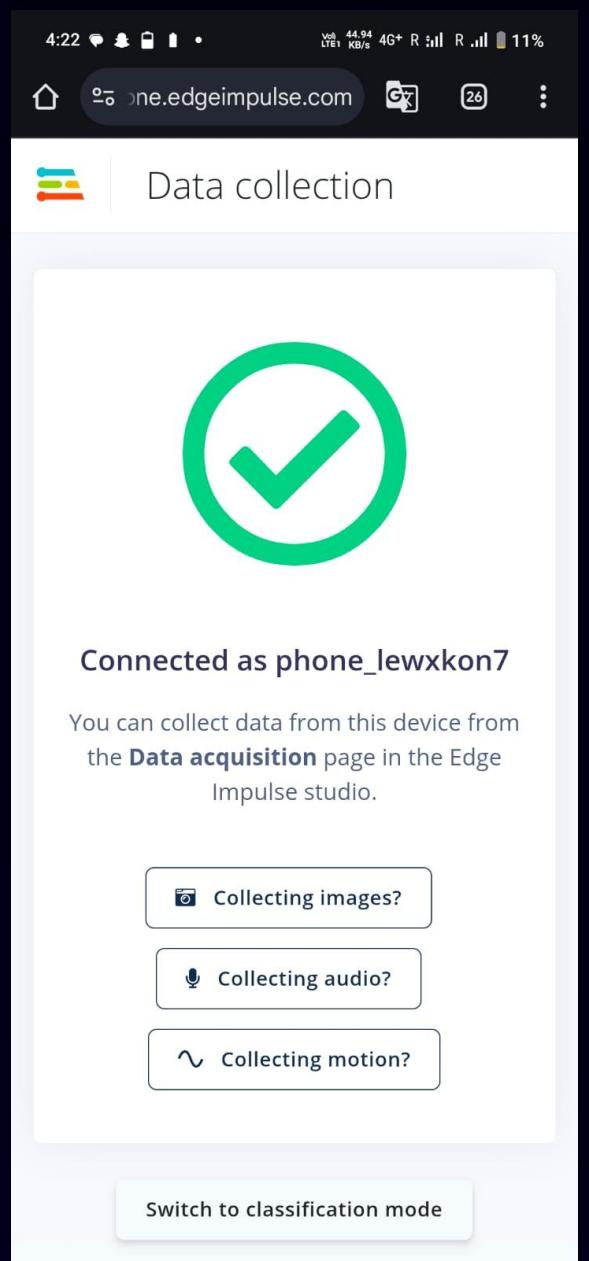
Live from Phone

From existing test samples

Step 39: From Phone



Live from Phone



Phone Screen

Step 40: From Phone

Classify new data

Device [?](#): phone_lewxkon7

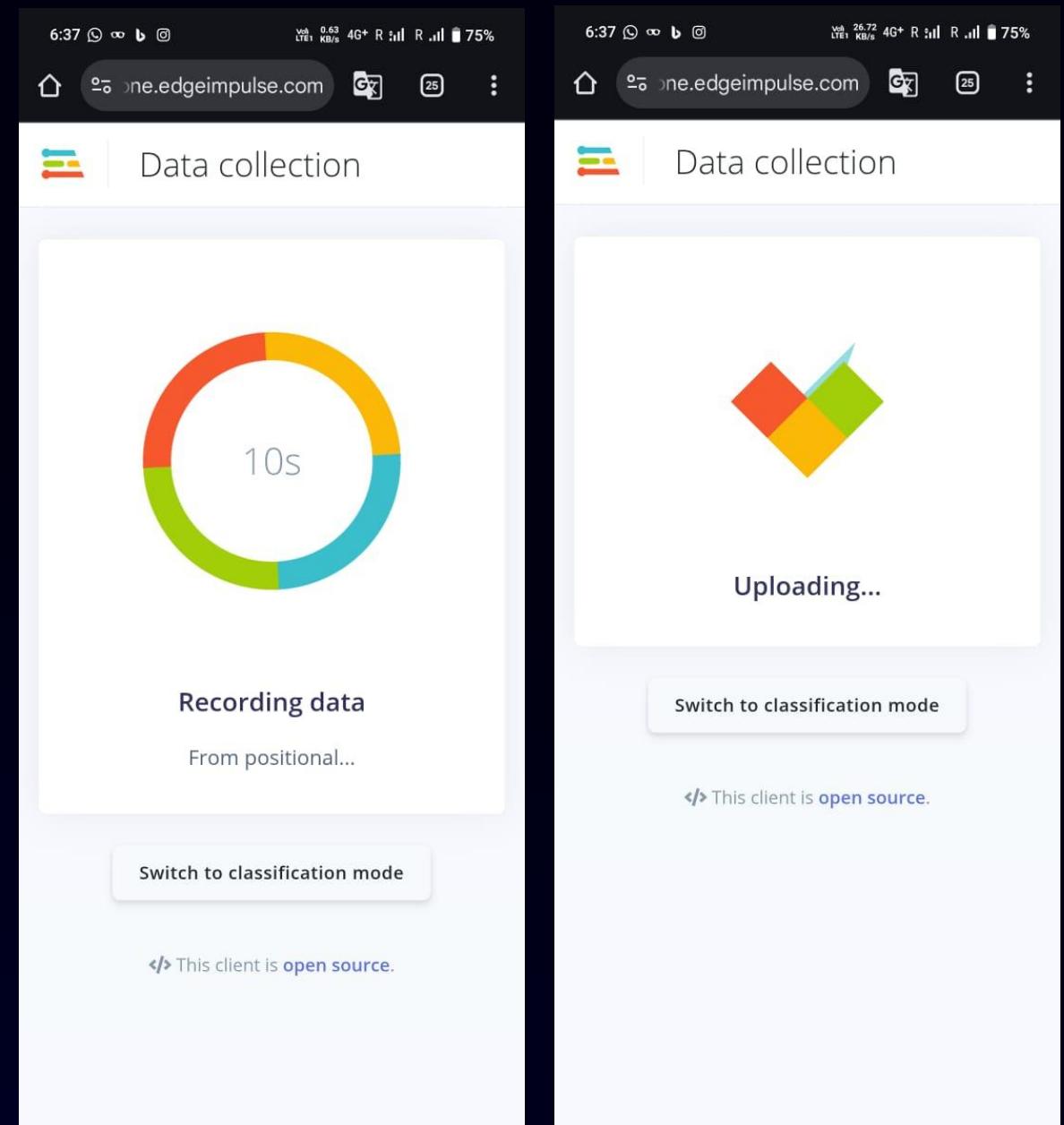
Sensor: Positional

Sample length (ms.): 10000

Frequency: 62.5Hz

Start sampling

Click start sampling



Phone Screen

Step 41: Detailed Result

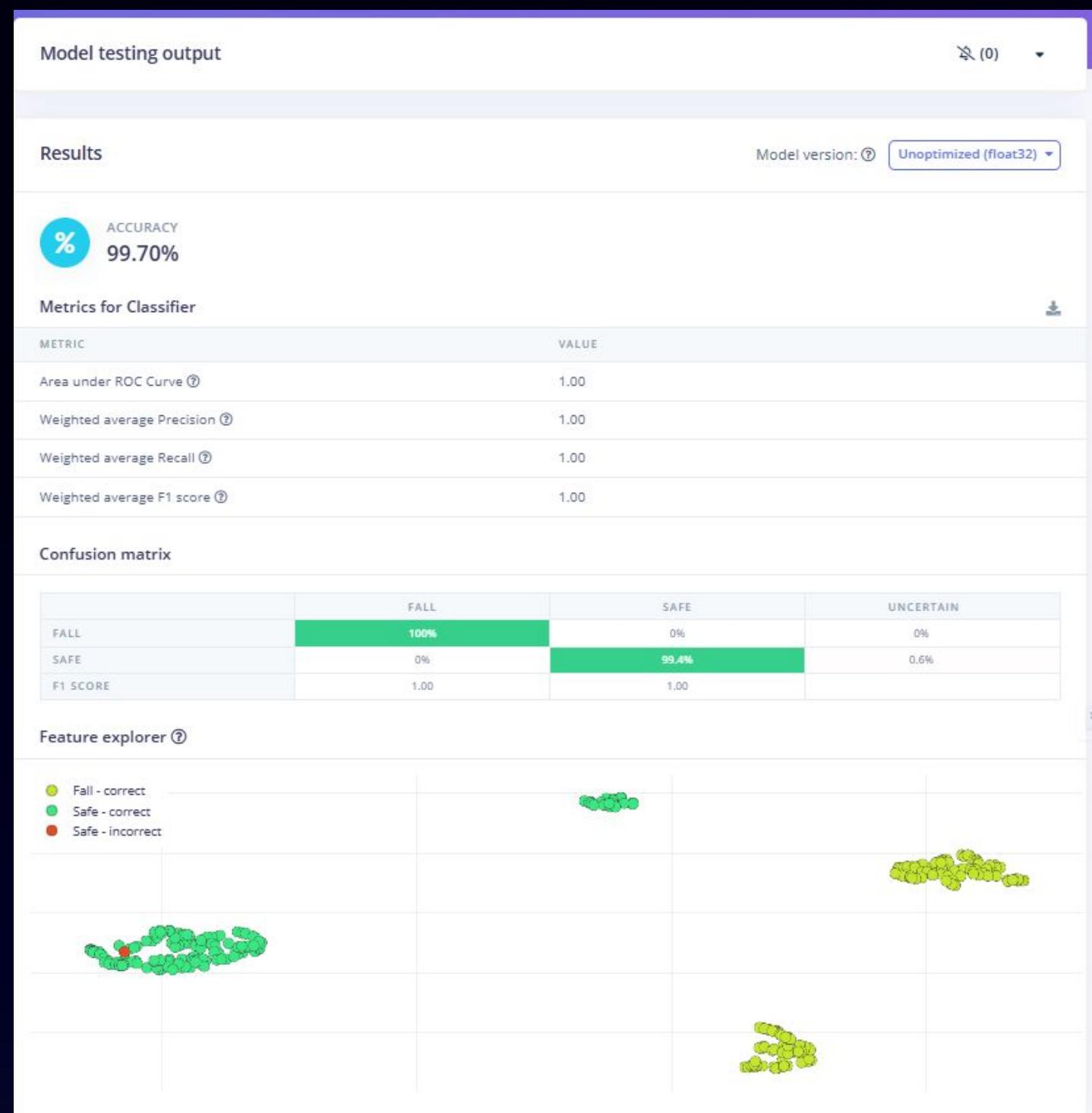
Detailed result			<input type="checkbox"/> Show only unknowns
TIMESTAMP	FALL	SAFE	
0	0	1.00	
192	0	1.00	
384	0	1.00	
576	0.09	0.91	
768	0.14	0.86	
960	0.20	0.80	
1,152	0.12	0.88	
1,344	0.24	0.76	
1,536	0.25	0.75	
1,728	0.24	0.76	
1,920	0.21	0.79	

Step 42: Model Testing

Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT	⋮
testing.5svaabof	testing	10s			⋮
testing.5sv9s7jl	testing	10s			⋮
testing.5sv9ql0e	testing	10s			⋮
Safe.5sndgb3a	Safe	10s	100%	42 Safe	⋮
Safe.5sndfm6q	Safe	10s	100%	42 Safe	⋮
Safe.5sndh06c	Safe	10s	100%	42 Safe	⋮
Safe.5sndhvr0	Safe	10s	98%	41 Safe, 1 uncertain	⋮
Fall.5snefig5	Fall	10s	100%	42 Fall	⋮
Fall.5sneg25d	Fall	10s	100%	42 Fall	⋮
Fall.5sneggmq	Fall	10s	100%	42 Fall	⋮
Fall.5snegusa	Fall	10s	100%	42 Fall	⋮



Step 43: Deployment

Configure your deployment

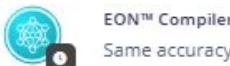
You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

 Search deployment options



MODEL OPTIMIZATIONS

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



Quantized (int8)

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	33 ms.	2 ms.	35 ms.
RAM	5.4K	5.9K	5.9K
FLASH	-	106.2K	-
ACCURACY			-

Unoptimized (float32)

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	33 ms.	3 ms.	36 ms.
RAM	5.4K	19.5K	19.5K
FLASH	-	376.1K	-
ACCURACY			99.70%

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

[Run model testing](#)

Estimate for Seeed SenseCAP A1101 (HX6537-A ARC DSP 400MHz) - [Change target](#)

[Build](#)

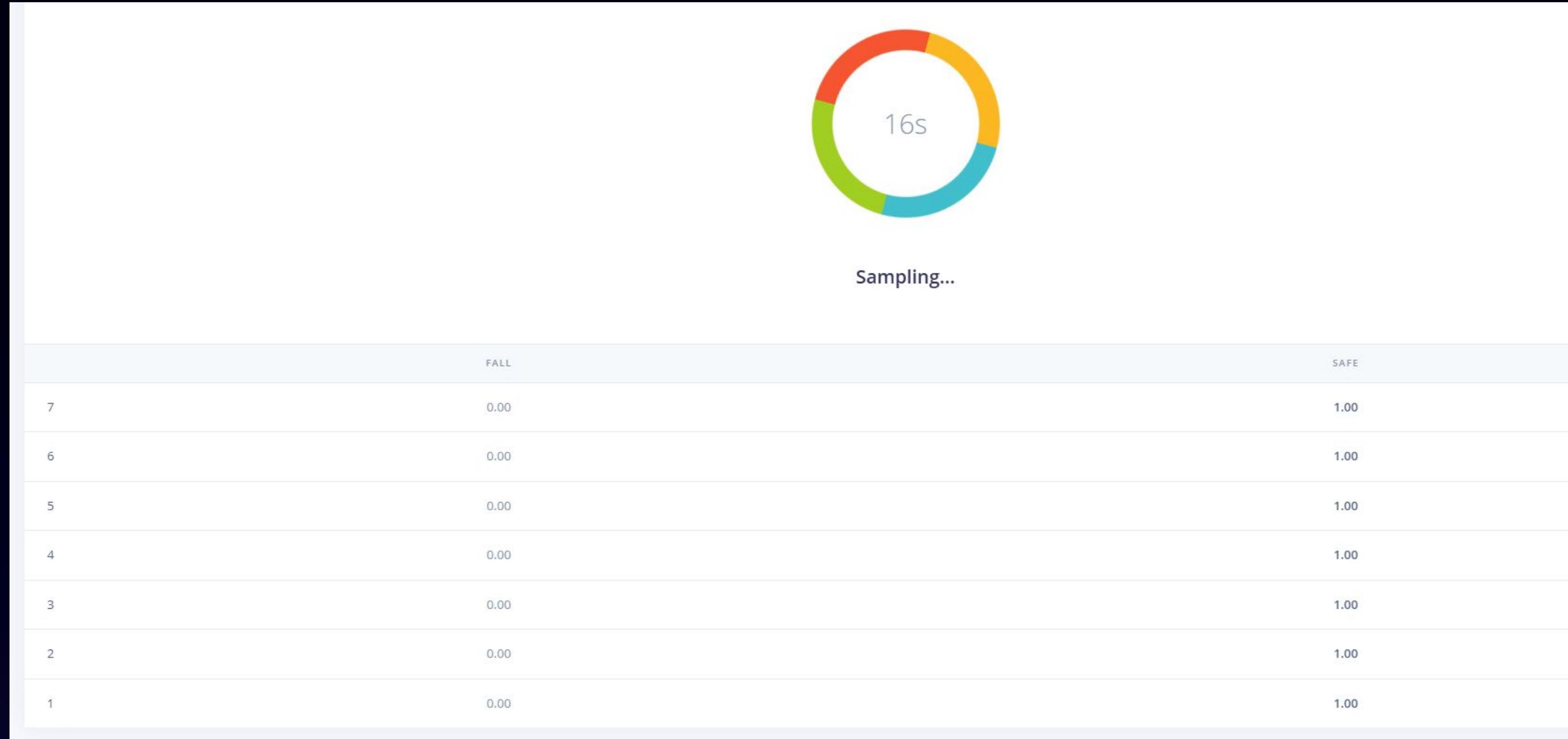
Run this model

Scan QR code or launch in browser to test your prototype



[Launch in browser](#)

Step 44: Deployment



Step 45: Final Library





Small Devices. Big Impact.

Thank You for Your Attention