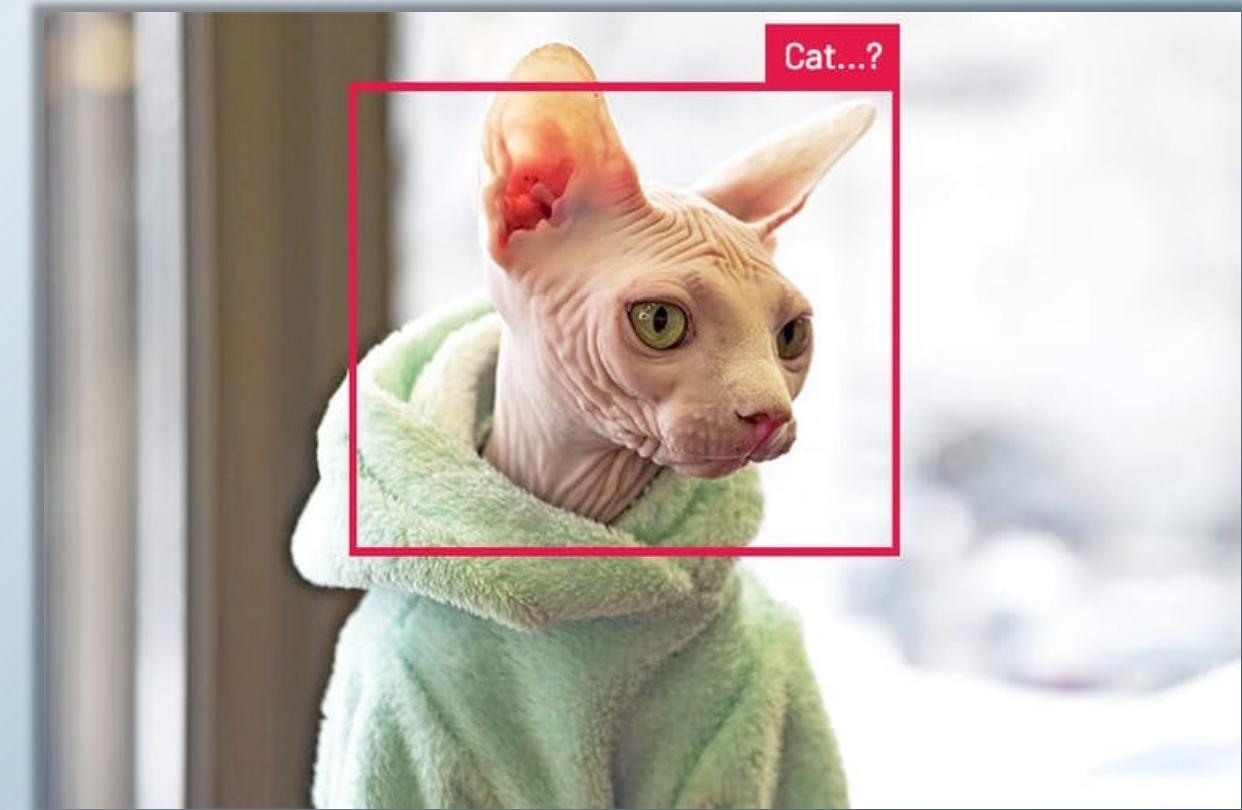


Accelerating TinyML with Edge Impulse

Rob Lauer

Director of Developer Relations





60% of the time; it works every time.



“

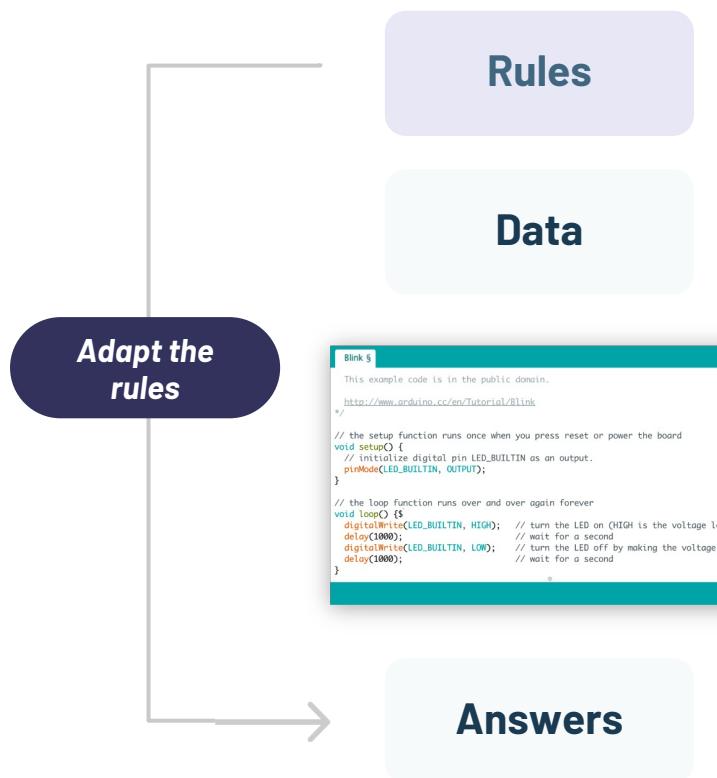
Machine learning is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence.

“

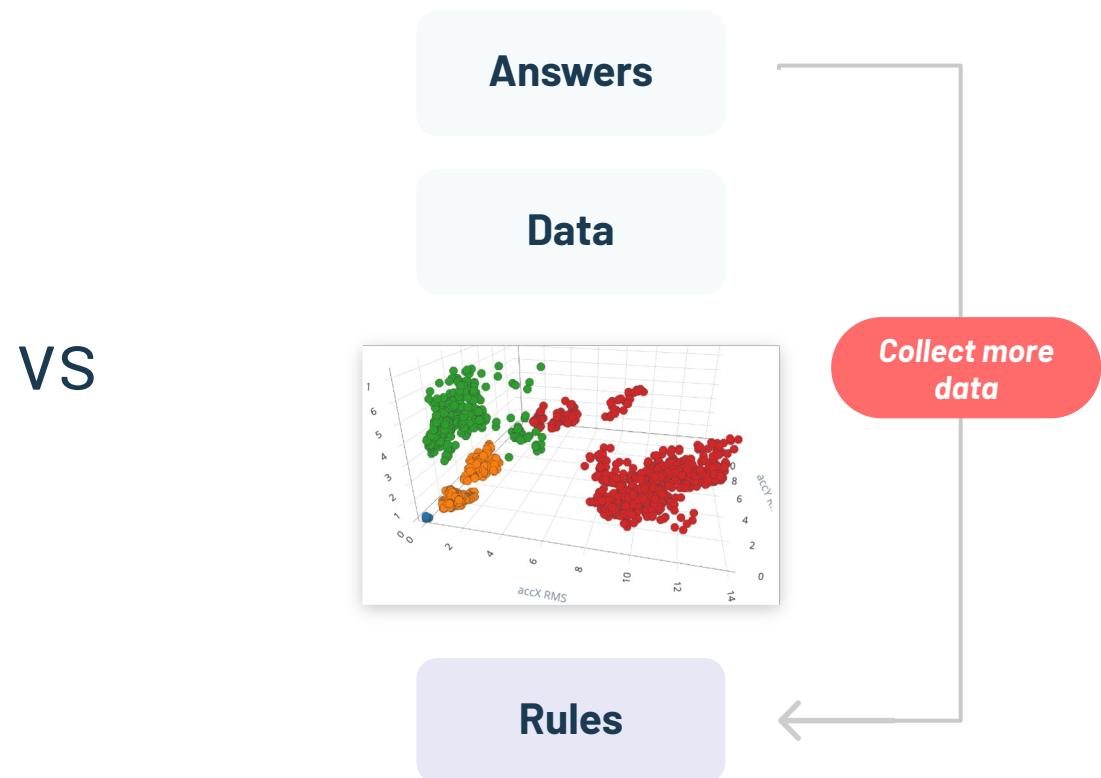
Machine learning lets computers learn from
data...versus being explicitly told what to do.

Paradigm Shift

Traditional Programming (Strict)



Machine Learning (Flexible)



VS

```
Blink.ino
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/Blink
/*
  The setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Answers

Rules

Edge Computing



Machine Learning

Benefits of Edge ML



Innovation

Add new differentiating features, become a market leader by standing out from your competition



Privacy

Data stays on the device, gets processed locally and drives remote alerts, notifications, and actions



Power

Stay operational for longer periods of time



Cost

Save on storage and compute costs by not sending raw data constantly to the cloud



Reliability

Be operational in low connectivity environments



Latency

Process data real-time on the edge device, without having to wait for a response back from the cloud

Edge ML Solutions



Wearables

- Voice Commands
- Health Prediction
- Activity Detection
- Gesture Recognition
- Acoustic Event Detection



Spaces

- Voice Commands
- Appliance Smart Sensing
- Fire Detection
- People Counting
- HVAC Maintenance
- Acoustic Event Detection



Industry

- PDM - Motors
- Power Grid Monitoring
- Worker Safety
- Manufacturing Quality
- Analog Meter Reading
- Smart Robotics



Logistics

- Asset Tracking
- Worker Safety
- Smart Cameras
- Battery Monitoring



EDGE IMPULSE

DATA ACQUISITION (CONTINUOUS GESTURES)

Training data Test data

Did you know? You can add data from any device through the Ingestion API.

DATA COLLECTED
14m 57s

LABELS
4

Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
updown.1955q7fb	updown	May 26 2020, 15:01:29	10s
wave.1955q7fa	wave	May 26 2020, 15:01:29	10s
updown.1955q7f7	updown	May 26 2020, 15:01:29	5s
updown.1955q7el	updown	May 26 2020, 15:01:29	10s
updown.1955q7em	updown	May 26 2020, 15:01:29	10s
snake.1955q7ce	snake	May 26 2020, 15:01:29	10s
updown.1955q7ev	updown	May 26 2020, 15:01:29	10s
updown.1955q7e3	updown	May 26 2020, 15:01:29	10s
updown.1955q7es	updown	May 26 2020, 15:01:29	10s

Record new data

No devices connected to the remote management API.

RAW DATA
updown.1955q7em

The chart displays three time-series signals: accX (red), accY (green), and accZ (blue). The x-axis represents time in samples from 0 to 9360, and the y-axis represents the signal value from -40 to 20. The signals show periodic oscillations characteristic of a vertical hand movement.

TinyML for time-series sensor data, vision and audio

updown.1955q7el updown May 26 2020, 15:01:29 10s

updown.1955q7du updown May 26 2020, 15:01:29 10s

updown.1955q7dn updown May 26 2020, 15:01:29 10s

Navigation: < 1 2 3 4 5 6 7 8 9 >

Jan Jongboom (demo)

Dashboard Devices Data acquisition Impulse design Retrain model Live classification Model testing Versioning Deployment GETTING STARTED Documentation Forums



The developer-first edge ML platform

- No royalty, no impact on BOM
- Your IP, stays your IP
- Total explainability, no black boxes

The screenshot shows the "CREATE IMPULSE (ANOMALY DETECTION)" screen. At the top, a description states: "An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data." Below this, there are four main configuration sections:

- Time series data**: Configures raw data settings. It includes fields for "Axes" (set to "accX, accY, accZ"), "Window size" (set to "2000 ms."), "Window increase" (set to "80 ms."), "Frequency (Hz)" (set to "100"), and "Zero-pad data" (checkbox checked).
- Spectral Analysis**: Configures feature extraction. It includes fields for "Name" ("Spectral features"), "Input axes" (checkboxes checked for "accX", "accY", and "accZ"), and "Output features" (checkbox checked for "Spectral features").
- Classification (Keras)**: Configures machine learning classification. It includes fields for "Name" ("NN Classifier"), "Input features" (checkbox checked for "Spectral features"), and "Output features" (checkbox checked for "2 (Mode 1, Mode 2)").
- Anomaly Detection (K-means)**: Configures anomaly detection. It includes fields for "Name" ("Anomaly detection") and "Input features" (checkbox checked for "Spectral features").

A dashed box highlights the "Add a processing block" area, indicating where additional signal processing steps can be added to the workflow.

How Edge Impulse Solves Edge ML Challenges

99% of sensor & device data is discarded

Accelerate next gen IoT products and solutions turning data from any edge device into actionable insights

Current ML tools are not designed for edge

Enable developers enterprise-wide to rapidly build ML solutions natively for edge devices, amplifying your ML experts

87% of AI/ML projects fail due to infra complexity

MLOps infrastructure from data collection to edge deployment, for enterprise data science and ML teams, with leading integrations

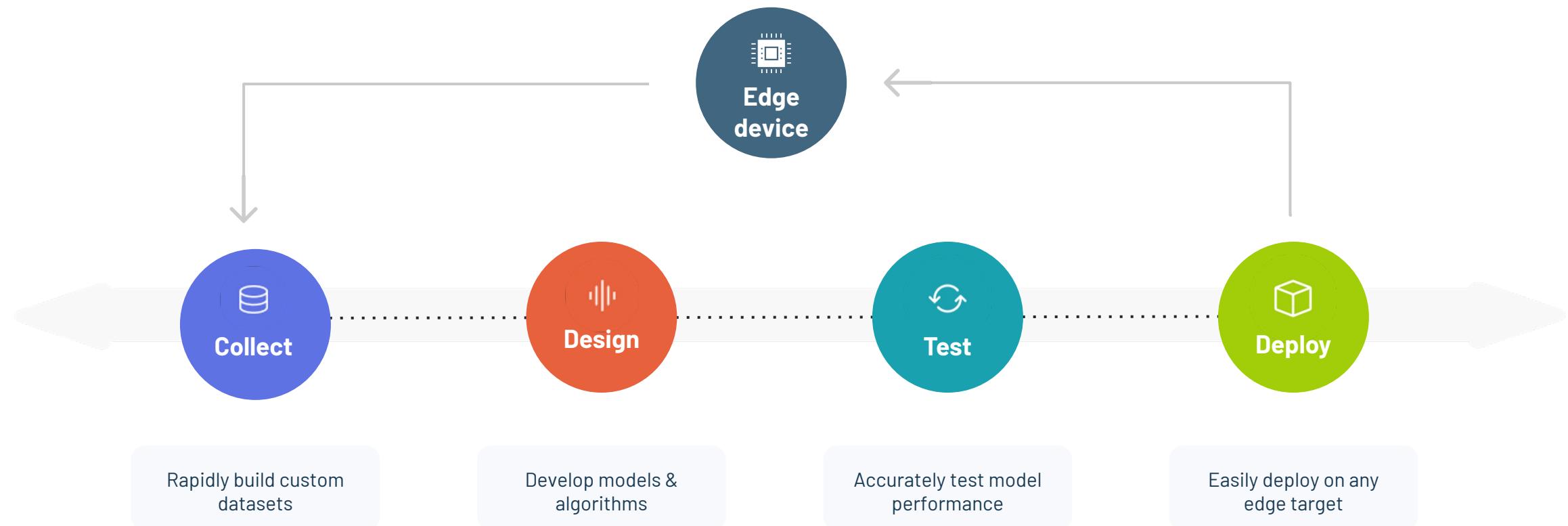
Difficulties in building smart product experiences

Build innovative experiences with ML with ease, leading to growth and competitive differentiation

Any sensor, any data, any use case

	Ultra low power	Low-end MCU	High-end MCU	NPU	MPU	GPU
Memory	Anomaly detection 10kB	Sensor data classification 18kB	Audio classification 50kB	Image classification 256kB	Object detection complex voice processing 1MB+	Video classification 1GB+
Sensor	✓	✓	✓	✓	✓	✓
Audio	✓	✓	✓	✓	✓	✓
Image			✓	✓	✓	✓
Video					✓	✓

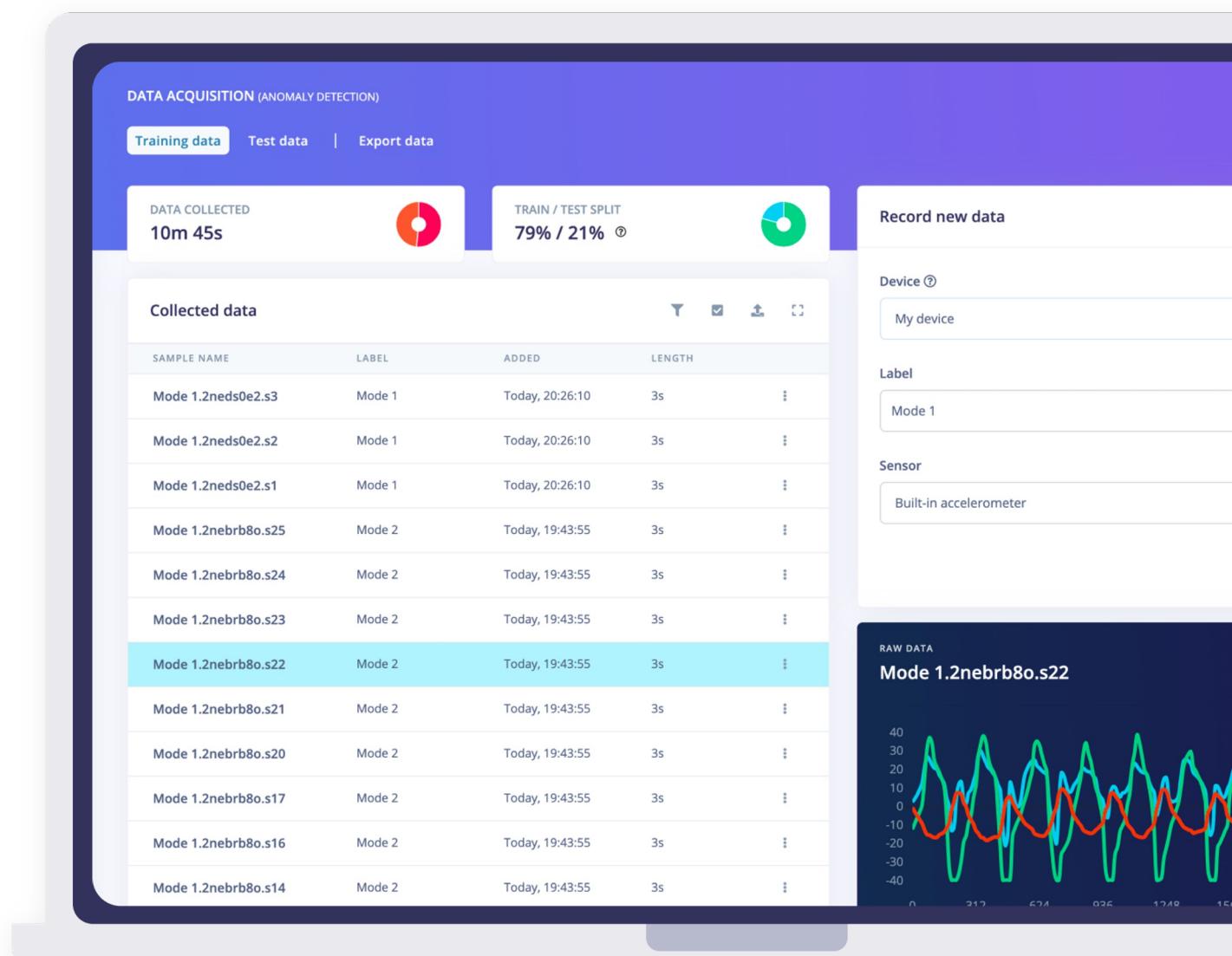
Develop Edge ML Applications with Edge Impulse



Collect

Build valuable datasets at scale

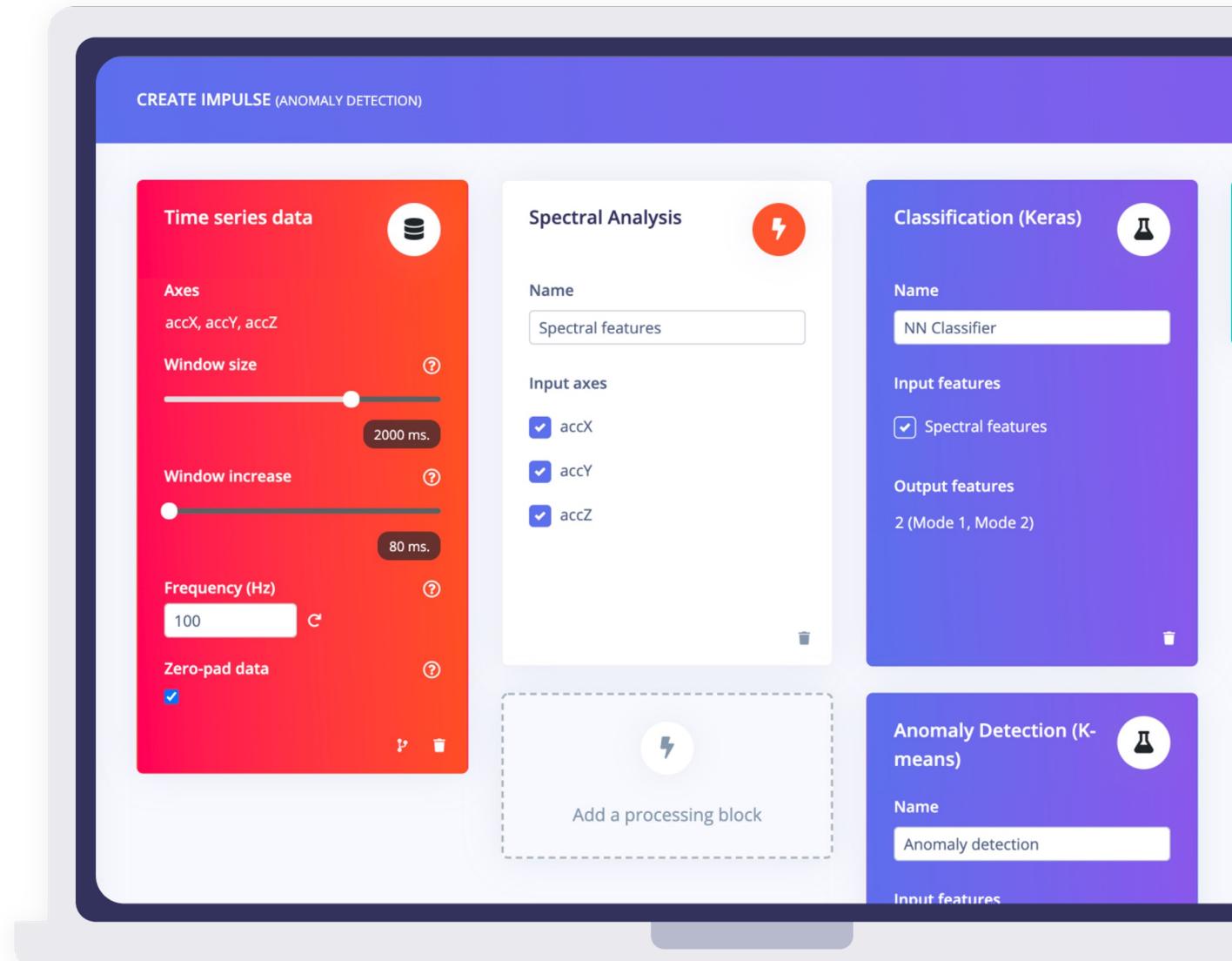
- Data science infrastructure
- Auto-labeling tools
- Integrations with data science tools
- Data traceability and quality control
- Secure data exchange portal



Design

Advanced algorithm and ML expertise

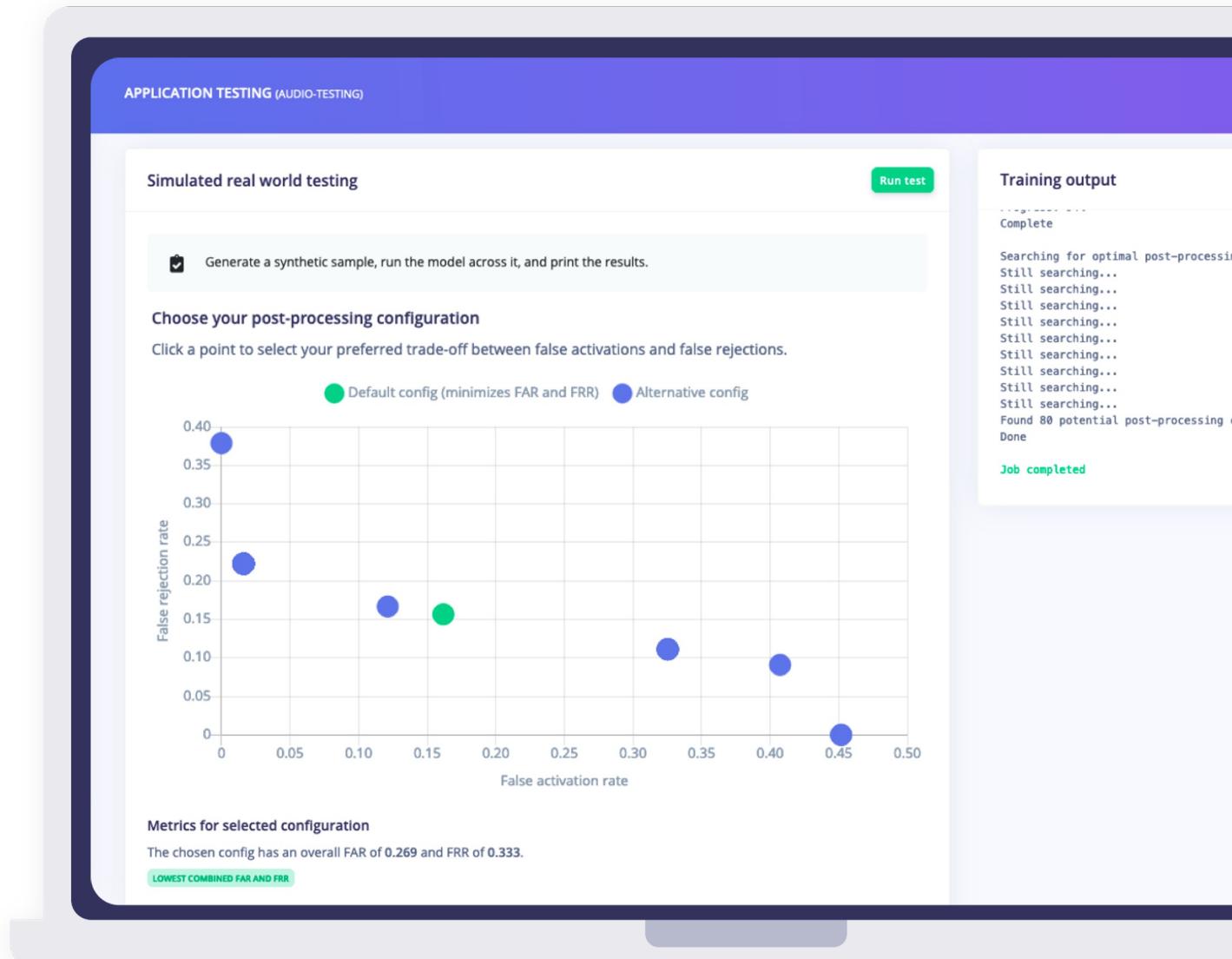
- Advanced algorithm and DSP expertise
- No black boxes
- Explainable AutoML
- Collaboration between teams



Test

Go to market faster, with confidence

- Hardware-aware development
- Full visibility across the whole ML pipeline
- Test against real world data
- Tune the algorithm to perform optimally



Deploy

Deploy to any edge device with ease

- The largest silicon ecosystem
- Award-winning compiler
- Get access to full source code
- Full firmware integration

DEPLOYMENT (TUTORIAL: CONTINUOUS MOTION RECOGNITION)

Deploy your impulse

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.](#)

Create library

Turn your impulse into optimized source code that you can run on any device.

 C++ library	 Arduino library	 Cube.MX CMSIS-PACK
 WebAssembly	 TensorRT library	

Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.

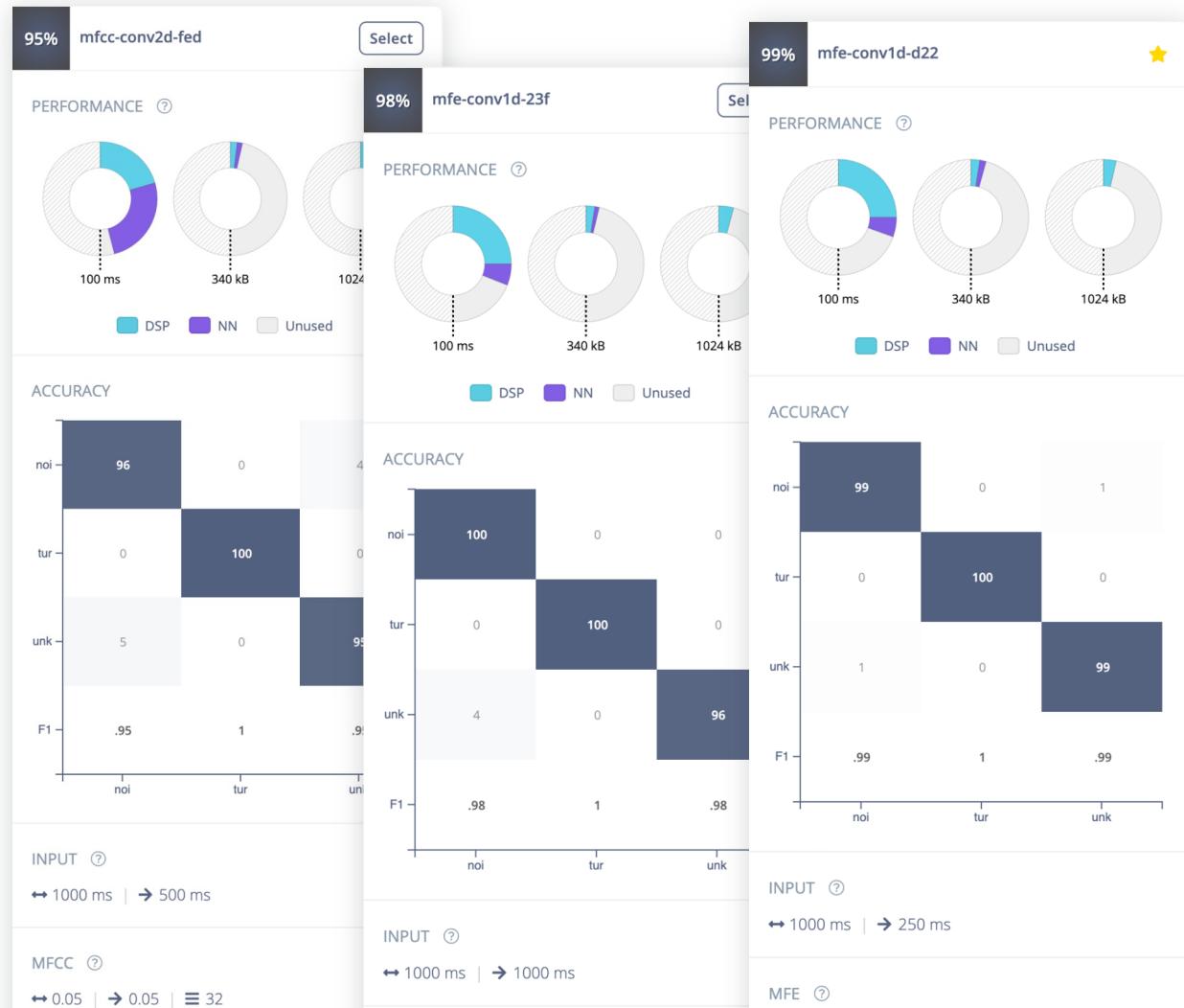


Advanced Edge Impulse Features



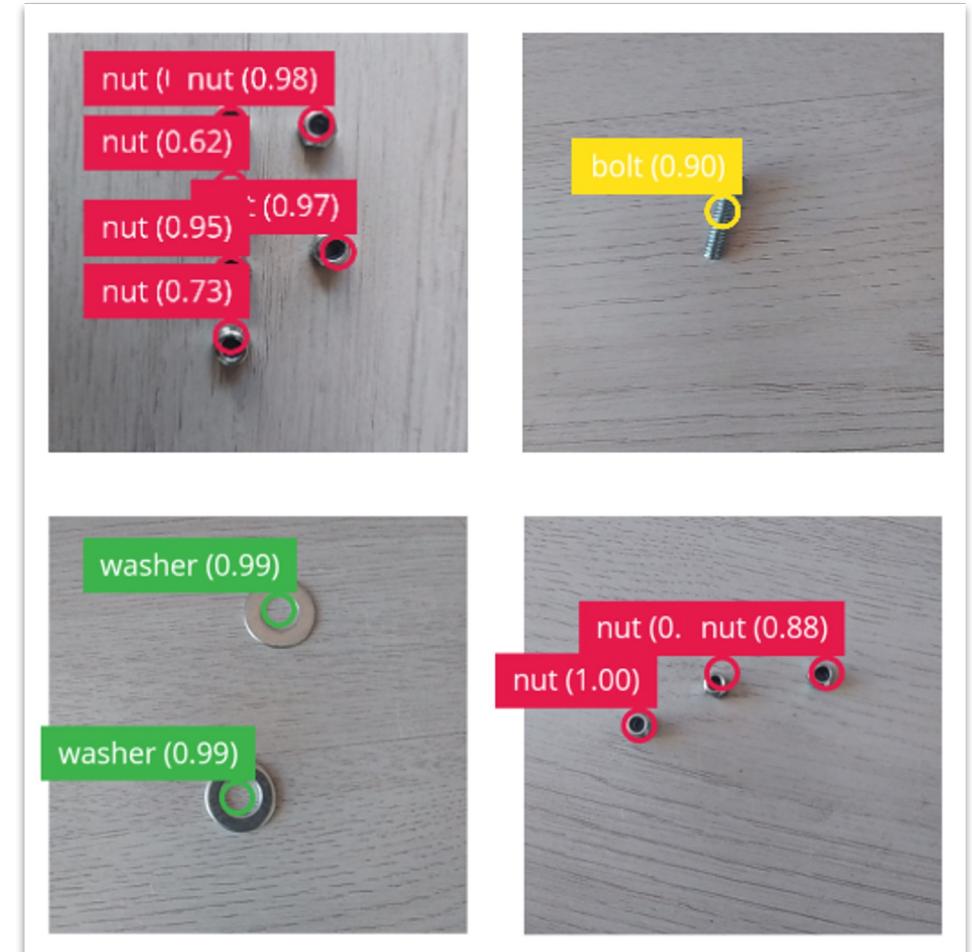
Optimize ML solutions with EON Tuner

- Make the most of limited and heterogeneous compute
- Find the perfect balance between feature extraction and model architecture
- Recommendations based on real performance metrics
- Built for constrained use cases to high end HW and complex use cases (e.g.: high-end CV)
- Suggestions on optimal hardware target for use case



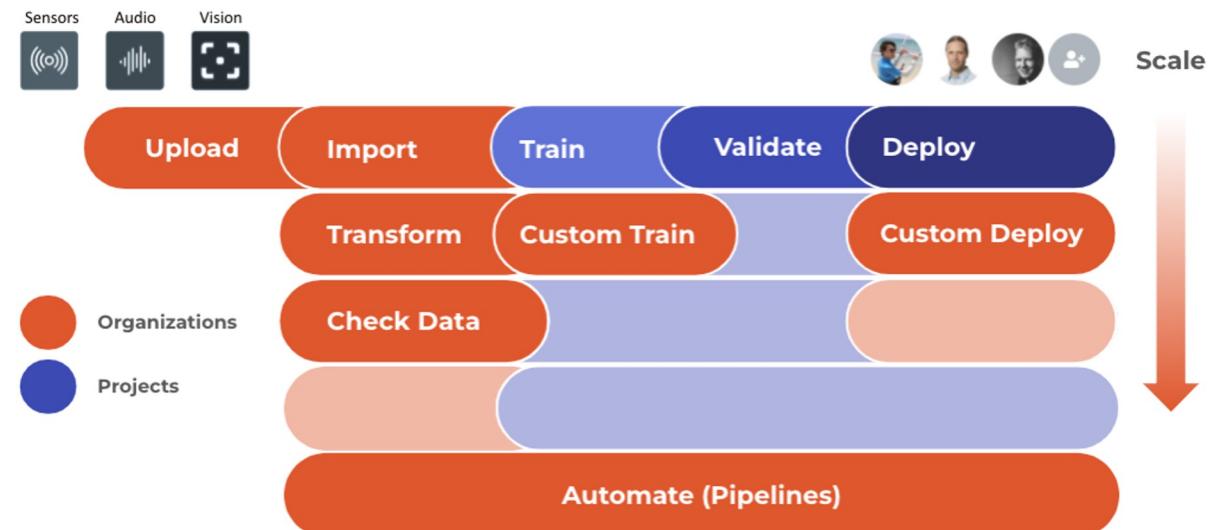
FOMO: Faster objects, more objects

- Object detection on MCUs
- 10 fps on Arm Cortex-M7 MCU
- Ultra fast on embedded Linux - 60 fps on RPi class
- Better at detecting smaller and more numerous objects
- Capable of segmentation and counting objects

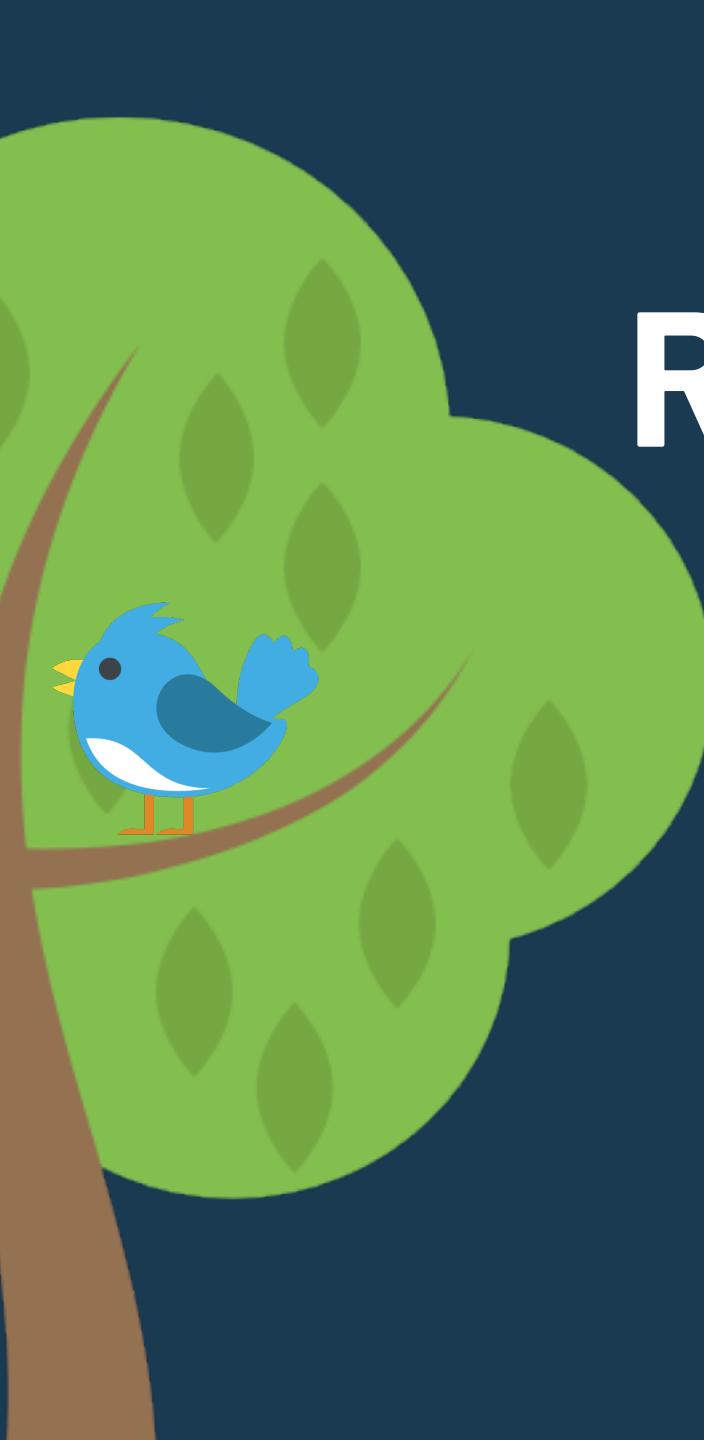


Advanced enterprise infrastructure

- SSO for all of your enterprise's users
- Store, view and manage all of your enterprise's data, projects and users
- Unlock full team collaboration
- Turn raw data into an ML ready dataset
- Build advanced queryable datasets, with easy quality control
- Save time and cost with cloud storage bucket integrations
- Create upload portals to allow 3rd parties to upload data securely



So what can you
do with it? 🤔



Remote Birding with the Raspberry Pi



Remote Birding Workflow

PIR Motion Sensor

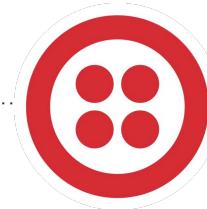
Detect motion from a bird at the bird feeder with infrared



Pi Camera
Snap a quick picture of the bird

TFLite

Use an ML model for birds around the world



Twilio SMS

Notehub.io securely routes the data to Twilio for an SMS



 EXPLORER

...

 bird.py 8 ×

PIBIRD

> images

.gitattributes

.gitignore

 bird.py 8

birds-label.txt

birds-model.tflite

 keys.py LICENSE README.md bird.py

104

105

```
106     def send_note(bird, prob):
107         """ upload the json note to notehub.io """
108         req = {"req": "note.add"}
109         req["file"] = "bird.qo"
110         req["start"] = True
111         req["body"] = {"bird": bird, "prob": prob,
112                         |   |   |   |
113                         "from": sms_from, "to": sms_to}
114         rsp = card.Transaction(req)
115
116
117     while True:
118         main()
119
```

118

119

3:49 1
◀ Edge

<  +1 (260) [REDACTED] >

Text Message
Today 3:49 PM

Sent from your Twilio trial account - Bird spotted! Gallus gallus (Probability: 89.1%)

  Text Message 





Build a Portable Speed Trap



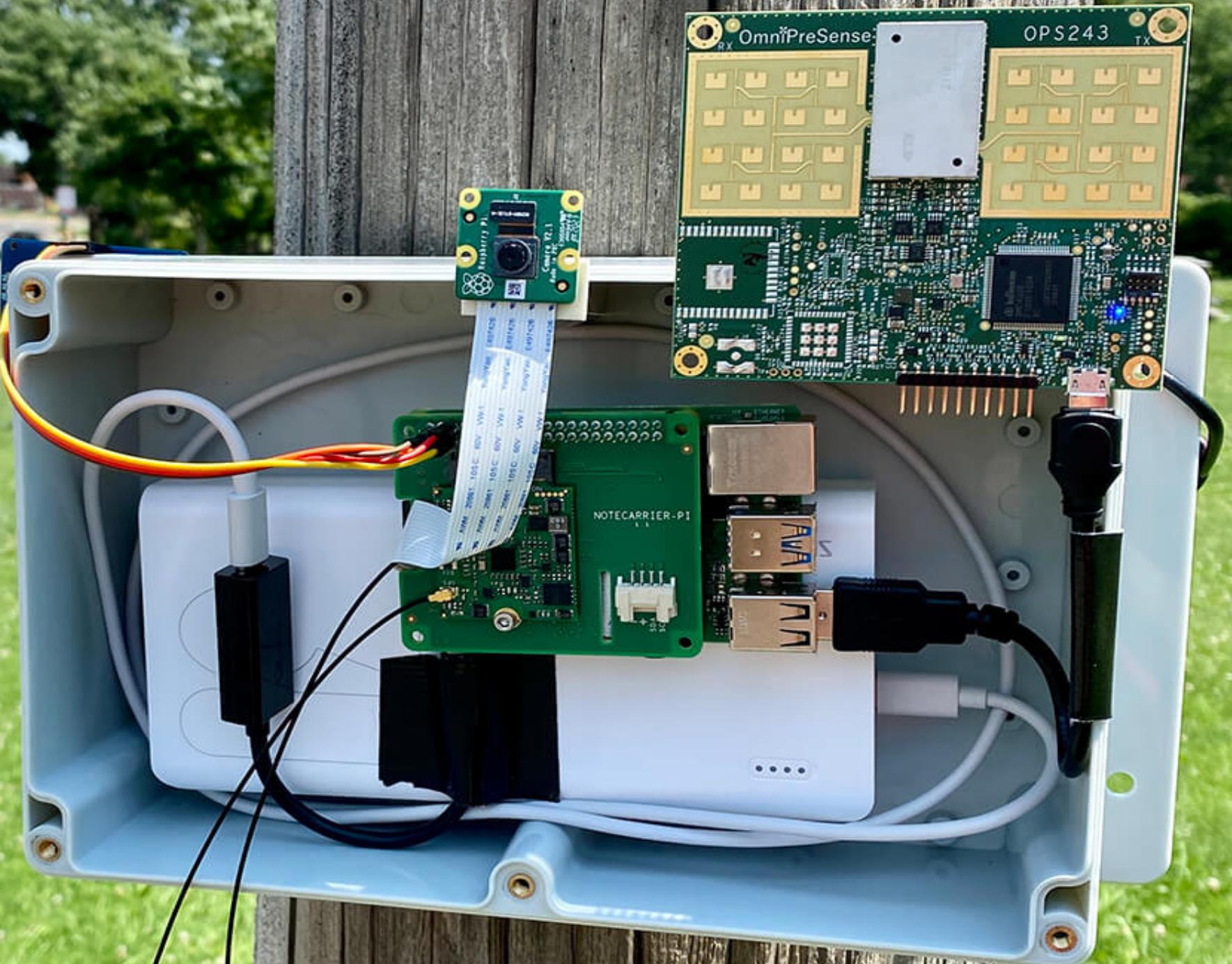
Portable “Speed Trap”

- Uses **ML** to identify vehicles on the road
- Uses a **doppler radar sensor** to gather speed
- Uses a **Notecard** to relay aggregated data to cloud
- Uses a **cloud-based dashboard** for reporting





car



ubidots

Devices ▾ Data ▾

≡ Speed Trap

Jul 06 2021 11:14 - Jul 06 2021 12:14 ▾

Deployed Location

A map of Madison, Wisconsin, showing the location of the speed trap. The map includes labels for Lake Mendota, Maple Bluff, Shorewood Hills, and various streets like University Avenue, State Street, and Monroe Street. A teal location pin is placed near the intersection of University Avenue and State Street.

Average Recorded Speed

A gauge chart indicating the average recorded speed. The scale ranges from 0 to 100, with the needle pointing to 35.45. The value is displayed in large orange digits.

Average Recorded Speed
35.45

Frequency of Speeding

A donut chart showing the frequency of speeding. The chart is divided into two segments: a smaller orange segment and a larger blue segment. The value 0.50 is displayed in the center of the blue segment.

Frequency of Speeding
0.50

Fastest Speed

Maximum (07/06/2021 11:14 - 07/06/2021 12:14)

The maximum speed recorded over the specified time period. The value is displayed in a large green font.

Fastest Speed
54.00

Last Updated: 07/06/2021 12:13

Speed Limit

Last value

The last recorded speed limit. The value is displayed in a large green font.

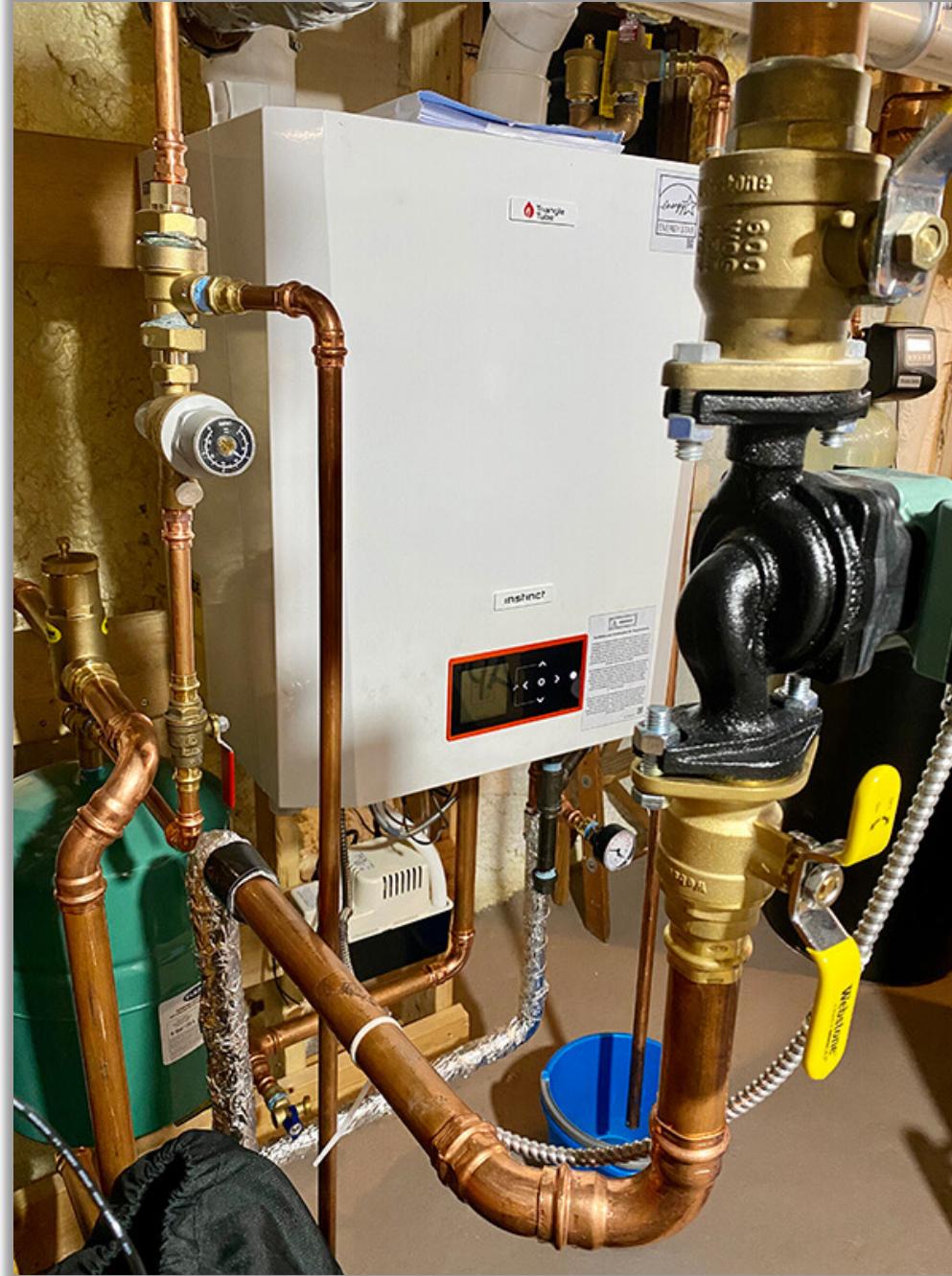
Speed Limit
30.00

Last Updated: 07/06/2021 12:13



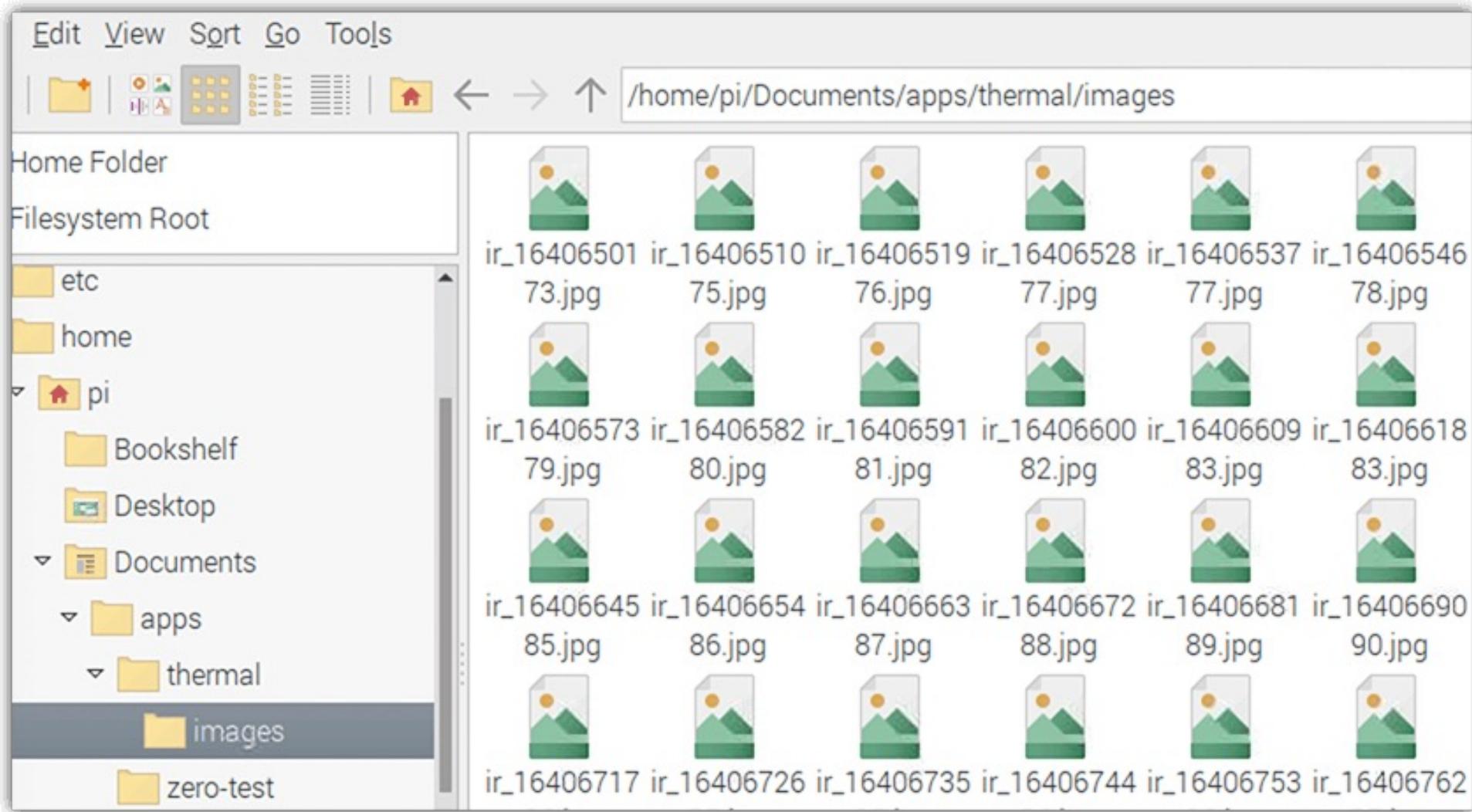
Thermal Image Anomaly Detection







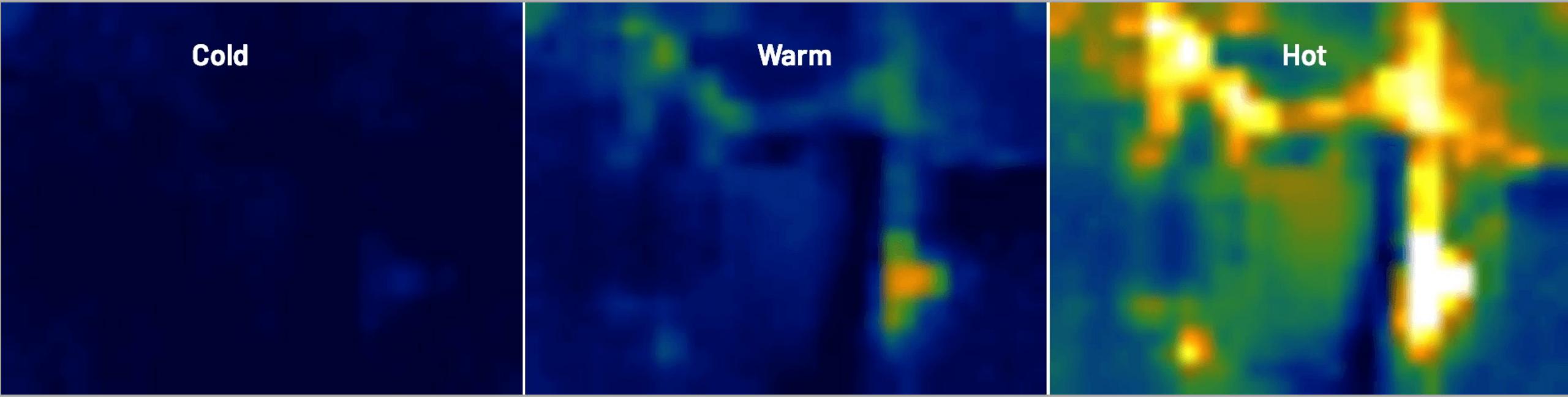




Cold

Warm

Hot

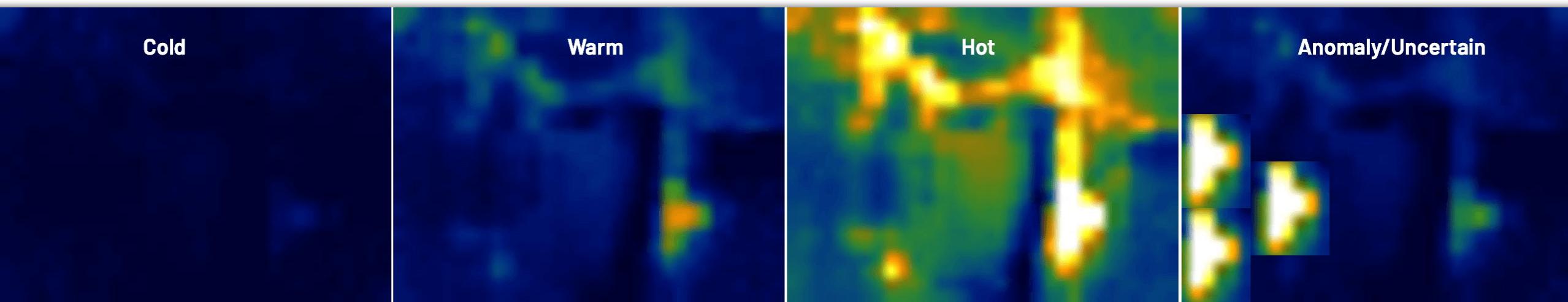


Cold

Warm

Hot

Anomaly/Uncertain



Welcome to your new Edge Impulse project!

You're ready to add real intelligence to your edge devices. Let's set up your project. What type of data are you dealing with?

Accelerometer data



Analyze movement of your device in real-time to predict machine failure, detect human gestures, or monitor rotating machines.

Audio



Listen to what's happening around you to create voice interfaces, listen to keywords, detect audible events, or to hear what's happening around your device.

Images



Add sight, monitor

Someth.

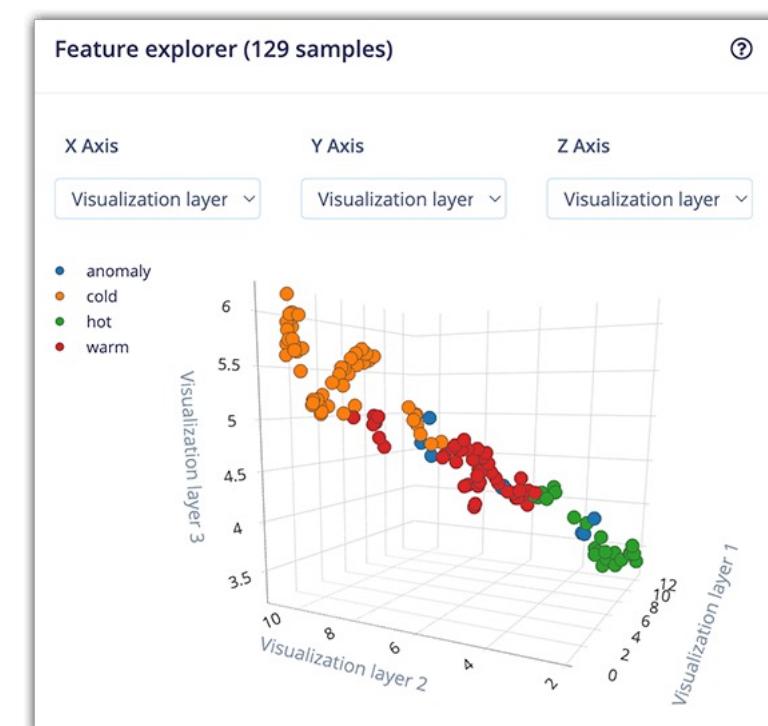


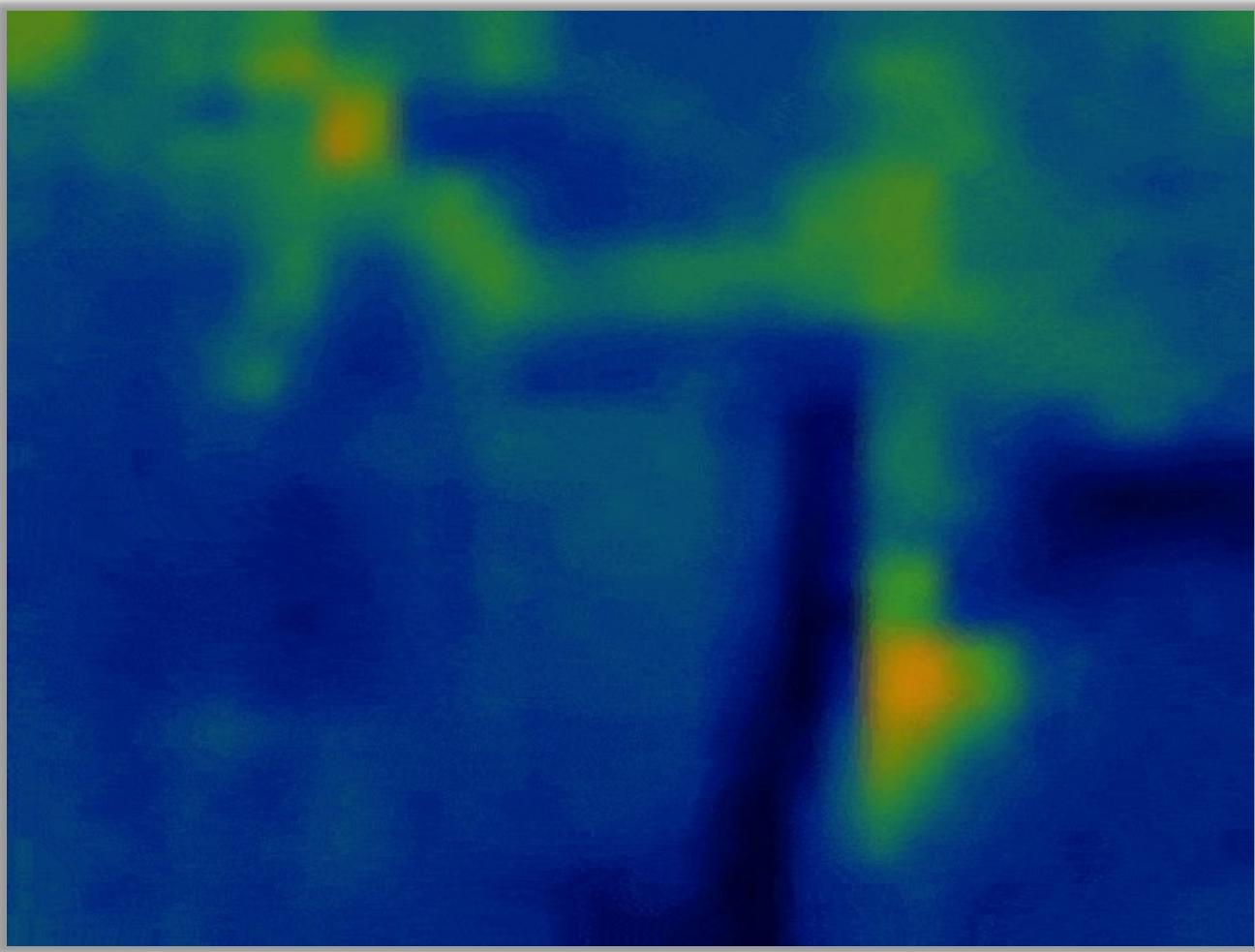
Different sensors

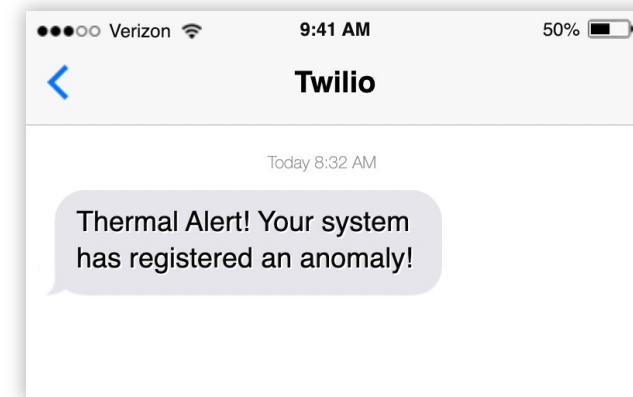
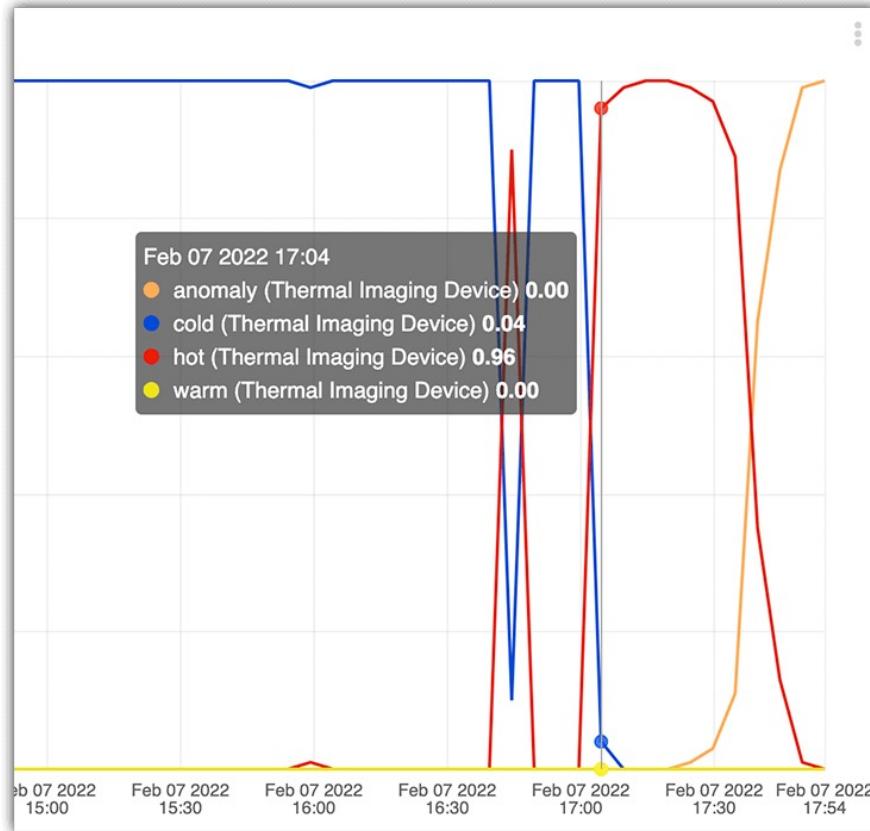
This screenshot shows the initial setup screen for a new Edge Impulse project. It includes sections for Accelerometer data, Audio, and Images. The Images section is highlighted with a red oval and has a red arrow pointing from the 'Images' label above it to the 'Add sight, monitor' text below. The Accelerometer and Audio sections also contain descriptive text and icons.

This screenshot shows the configuration interface for a new Edge Impulse project. It includes four main sections: 'Image data' (red background), 'Image' (white background), 'Classification (Keras)' (purple background), and 'Output features' (green background). The 'Image data' section is for image processing, 'Image' is for naming the dataset, 'Classification (Keras)' is for defining the model architecture, and 'Output features' lists the categories: 'anomaly', 'cold', 'hot', and 'warm'. A 'Save Impulse' button is located at the bottom right.

This screenshot shows the 'Data Acquisition' interface for a thermal dataset. It displays '120 items' collected so far. The 'Record new data' section shows a heatmap titled 'ir_1640781683.jpg.2o3fu445'. The left sidebar contains navigation links for Dashboard, Devices, Data acquisition, EON Tuner, Retrain model, Live classification, Model testing, Versioning, Deployment, Documentation, and Forums.







```

# initialize variables for Edge Impulse
runner = None
dir_path = os.path.dirname(os.path.realpath(__file__))
modelfile = os.path.join(dir_path, "modelfile.eim")

# initialize the Blues Wireless Notecard (blues.io)
productUID = keys.PRODUCT_UID
port = I2C("/dev/i2c-1")
nCard = notecard.OpenI2C(port, 0, 0)

# associate Notecard with a project on Notehub.io
hub.set(nCard, product=productUID, mode="periodic", outbound=30, inbound=720)

def main():

    print('MODEL: ' + modelfile)

    with ImageImpulseRunner(modelfile) as runner:
        try:
            model_info = runner.init()
            print('Loaded runner for "%s"' % model_info['product'])
            labels = model_info['labels']

            # grab image from the thermal camera
            filename = thermal.take()
            img = cv2.imread("images/" + filename)

            features, cropped = runner.get_features_from_image(img)
            res = runner.classify(features)

            if "classification" in res["result"].keys():
                note_body = {}
                for label in labels:
                    score = res['result']['classification'][label]
                    print('%s: %.2f\t' % (label, score), end='')
                    note_body[label] = round(score, 4)

                note.add(nCard,
                         file="thermal.qo",
                         body=note_body)

        while True:
            main()
            time.sleep(60 * 5)

```



Lab!



- Edge Impulse Studio Account
- Node.js
- Edge Impulse CLI
 - `npm install -g edge-impulse-cli`

dev.blues.io/swan