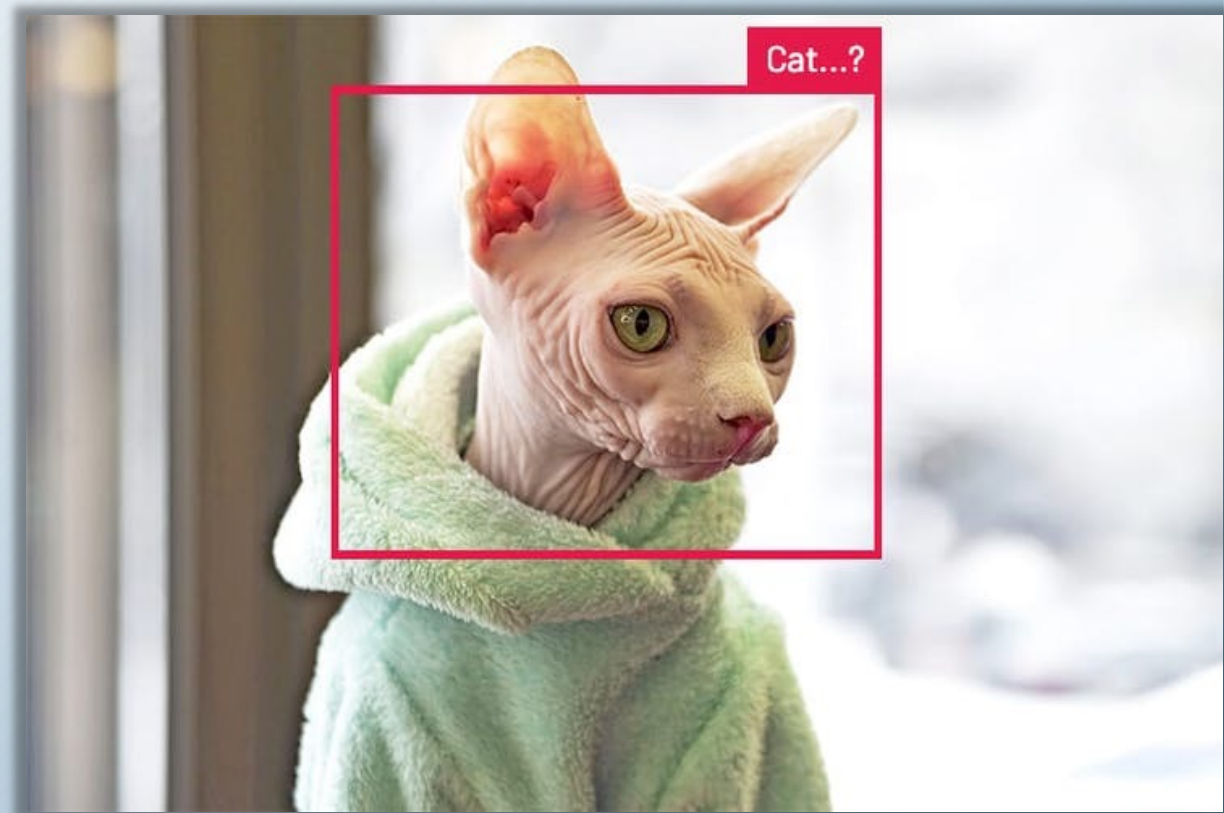
Accelerating TinyML with Edge Impulse

Rob Lauer
Director of Developer Relations

EDGE IMPULSE




60% of the time, it works every time.


cup: 53%
wine glass: 51%
cell phone: 51%

“

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.

“

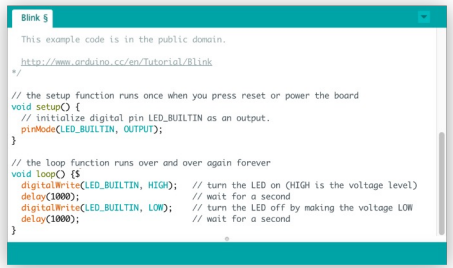
Machine learning lets computers learn from data...versus being explicitly told what to do.

Paradigm Shift

Traditional Programming (Strict)

Rules

Data



Answers

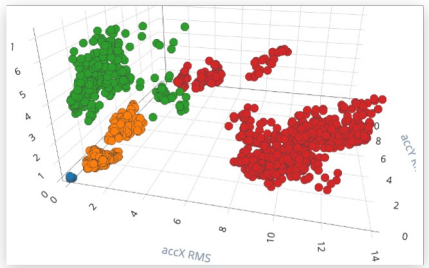
Adapt the rules

VS

Machine Learning (Flexible)

Answers

Data



Rules

Collect more data

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

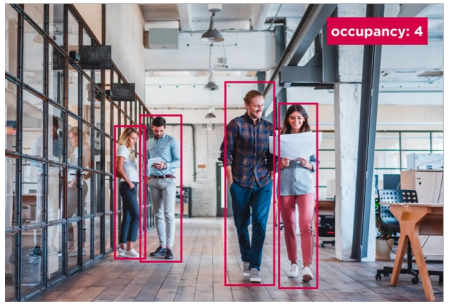
Latency

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

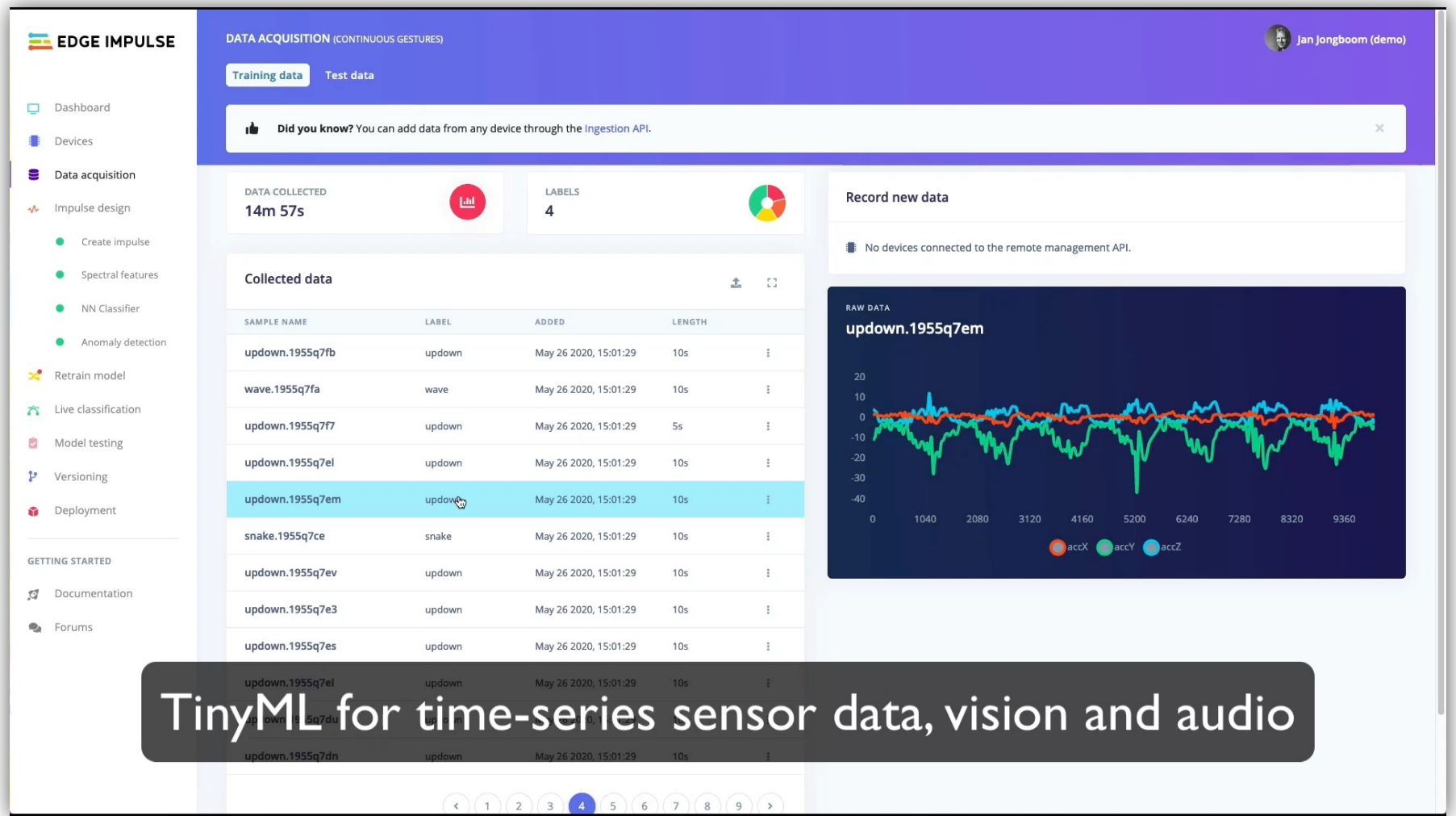
Industry

- 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

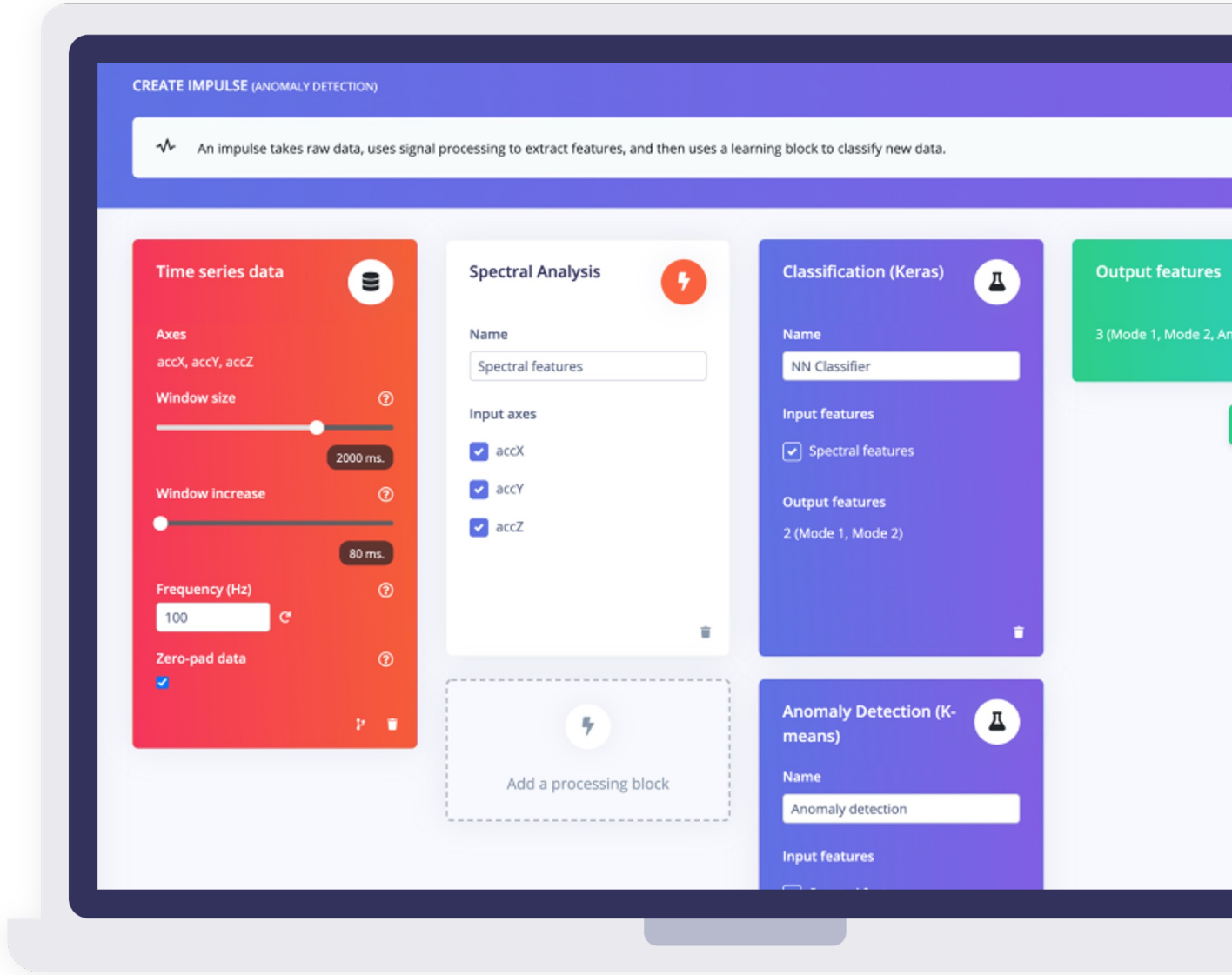


TinyML for time-series sensor data, vision and audio

EDGE IMPULSE

The developer-first edge ML platform

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



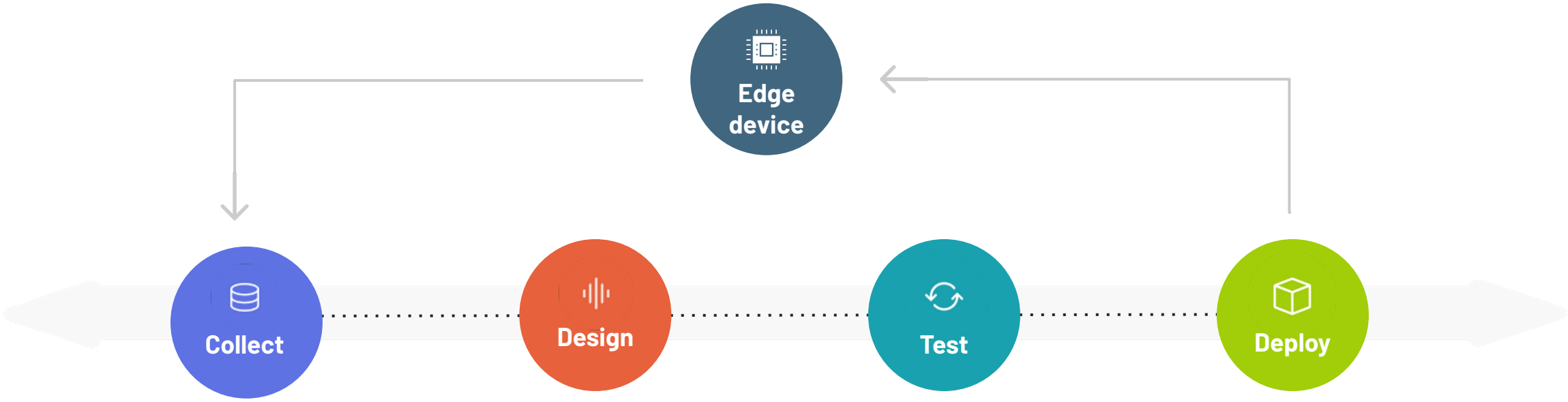
How Edge Impulse Solves Edge ML Challenges

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



Rapidly build custom datasets

Develop models & algorithms

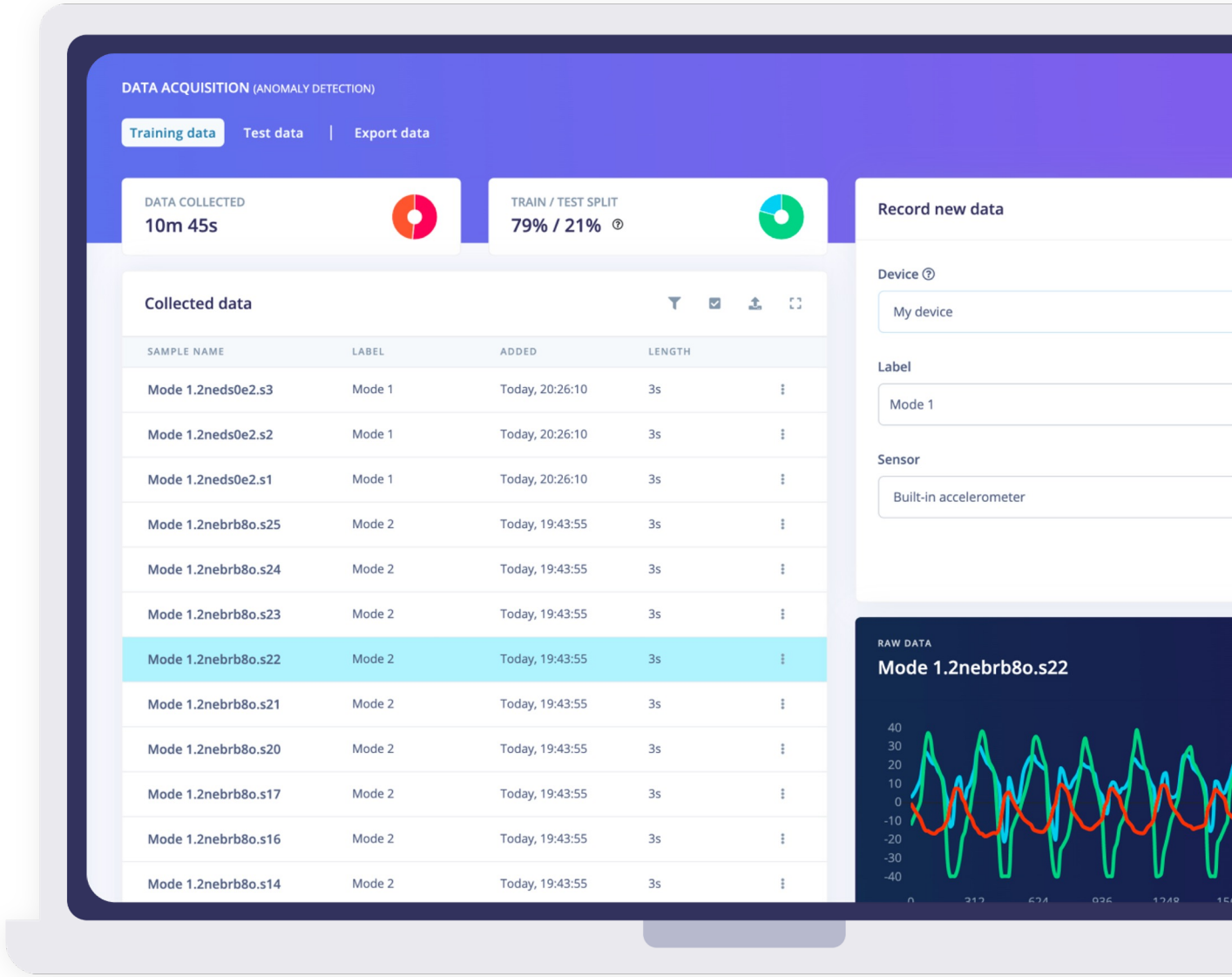
Accurately test model performance

Easily deploy on any edge target

Collect

Build valuable datasets at scale

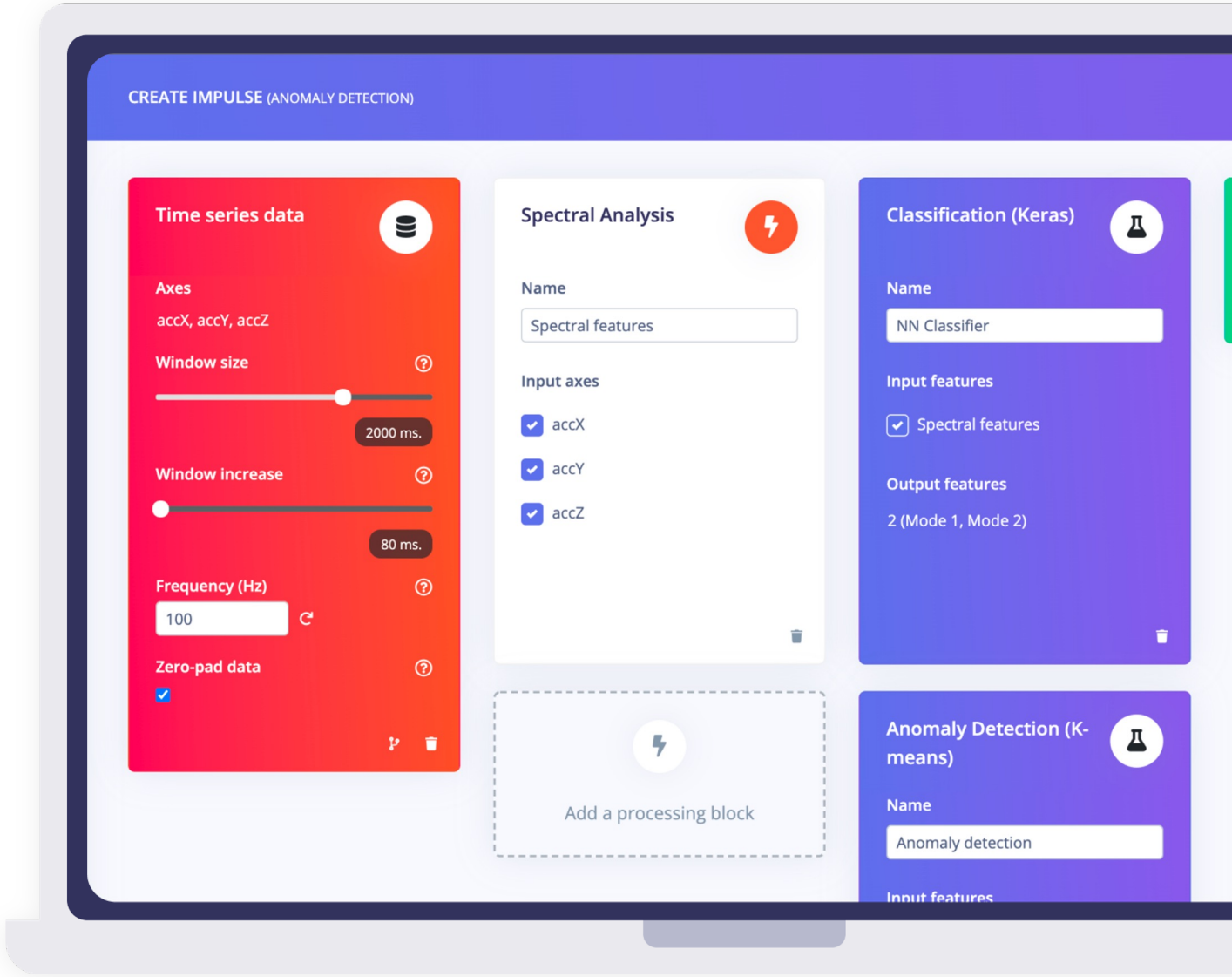
- Data science infrastructure
- Auto-labeling tools
- Integrations with data science tools
- Data traceability and quality control
- Secure data exchange portal



Design

Advanced algorithm and ML expertise

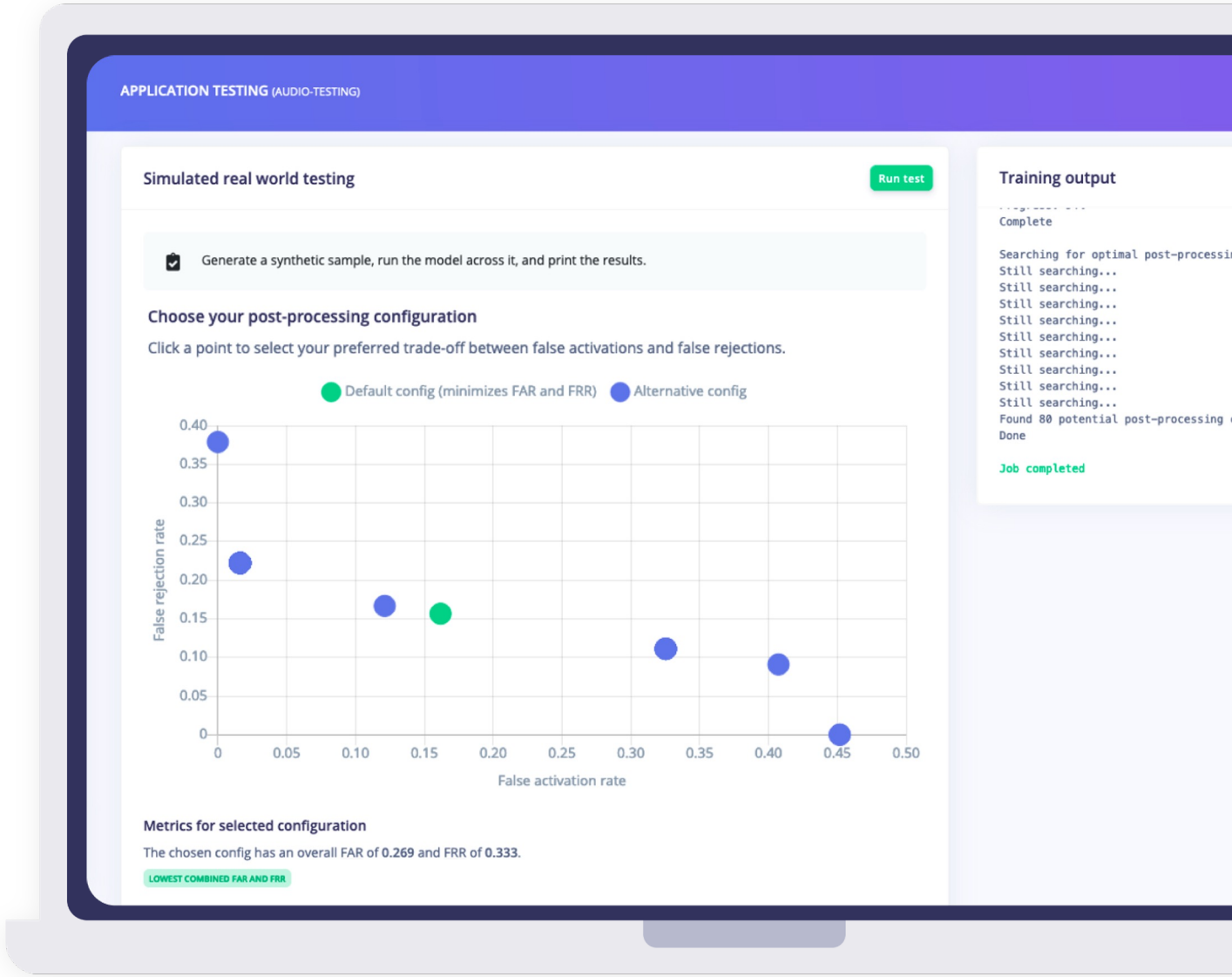
- Advanced algorithm and DSP expertise
- No black boxes
- Explainable AutoML
- Collaboration between teams



Test

Go to market faster, with confidence

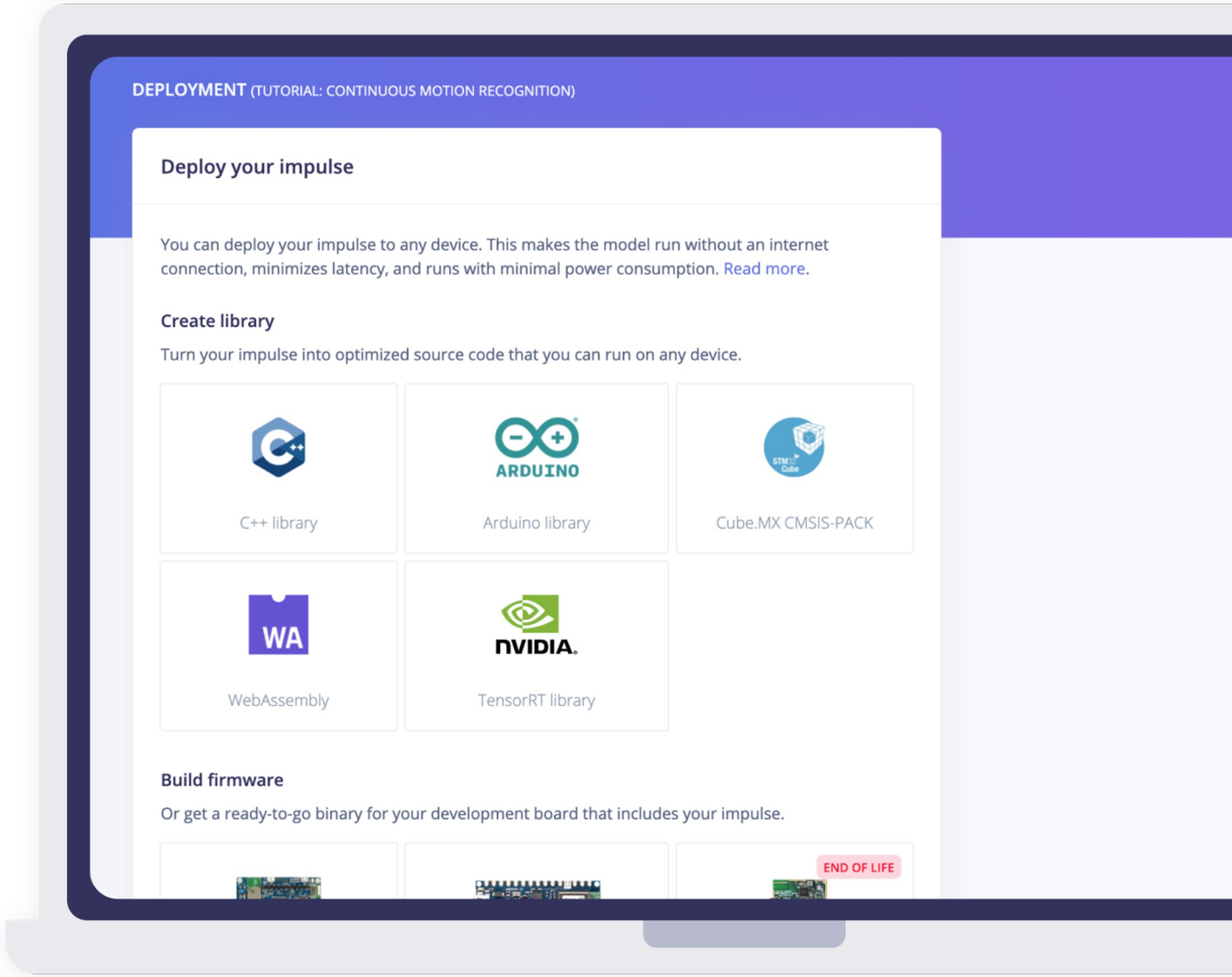
- Hardware-aware development
- Full visibility across the whole ML pipeline
- Test against real world data
- Tune the algorithm to perform optimally



Deploy

Deploy to any edge device with ease

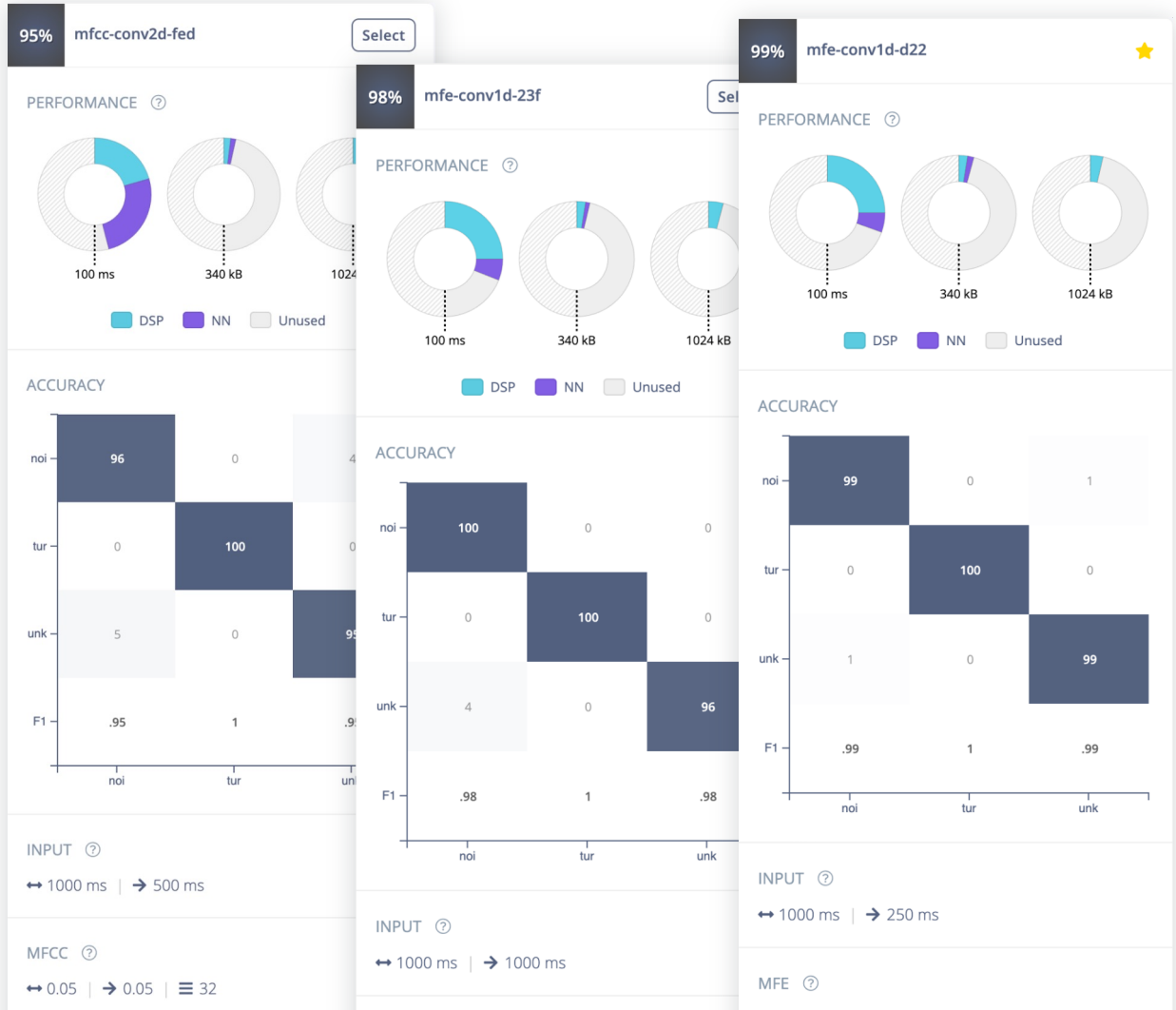
- The largest silicon ecosystem
- Award-winning compiler
- Get access to full source code
- Full firmware integration



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

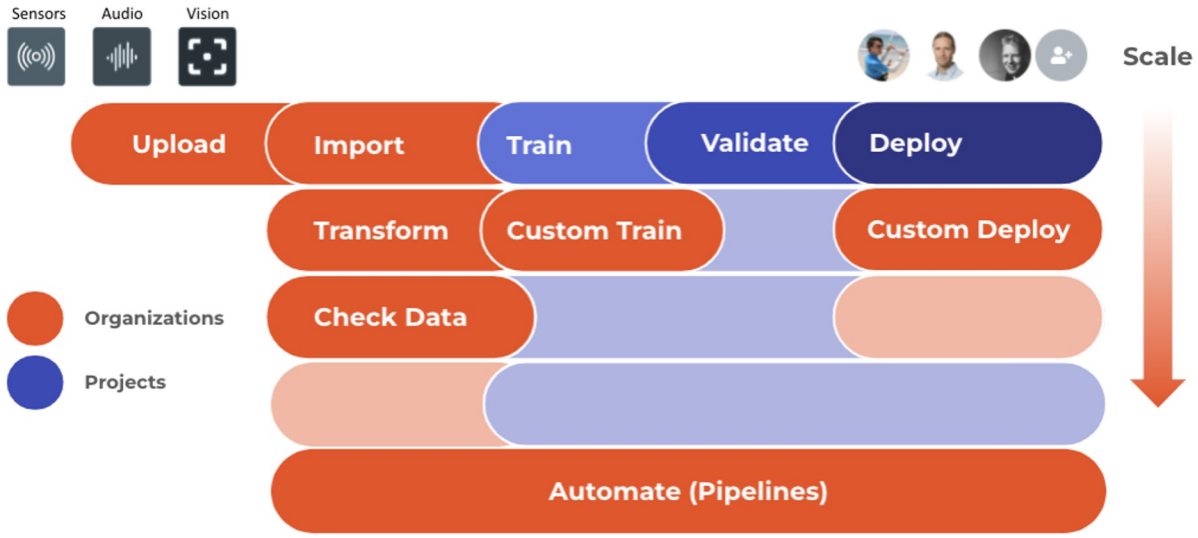


FOMO: Faster objects, more objects

- 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

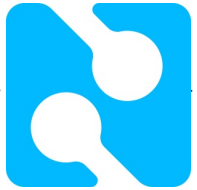
Remote Birding Workflow

PIR Motion Sensor
Detect motion from a bird at the bird feeder with infrared

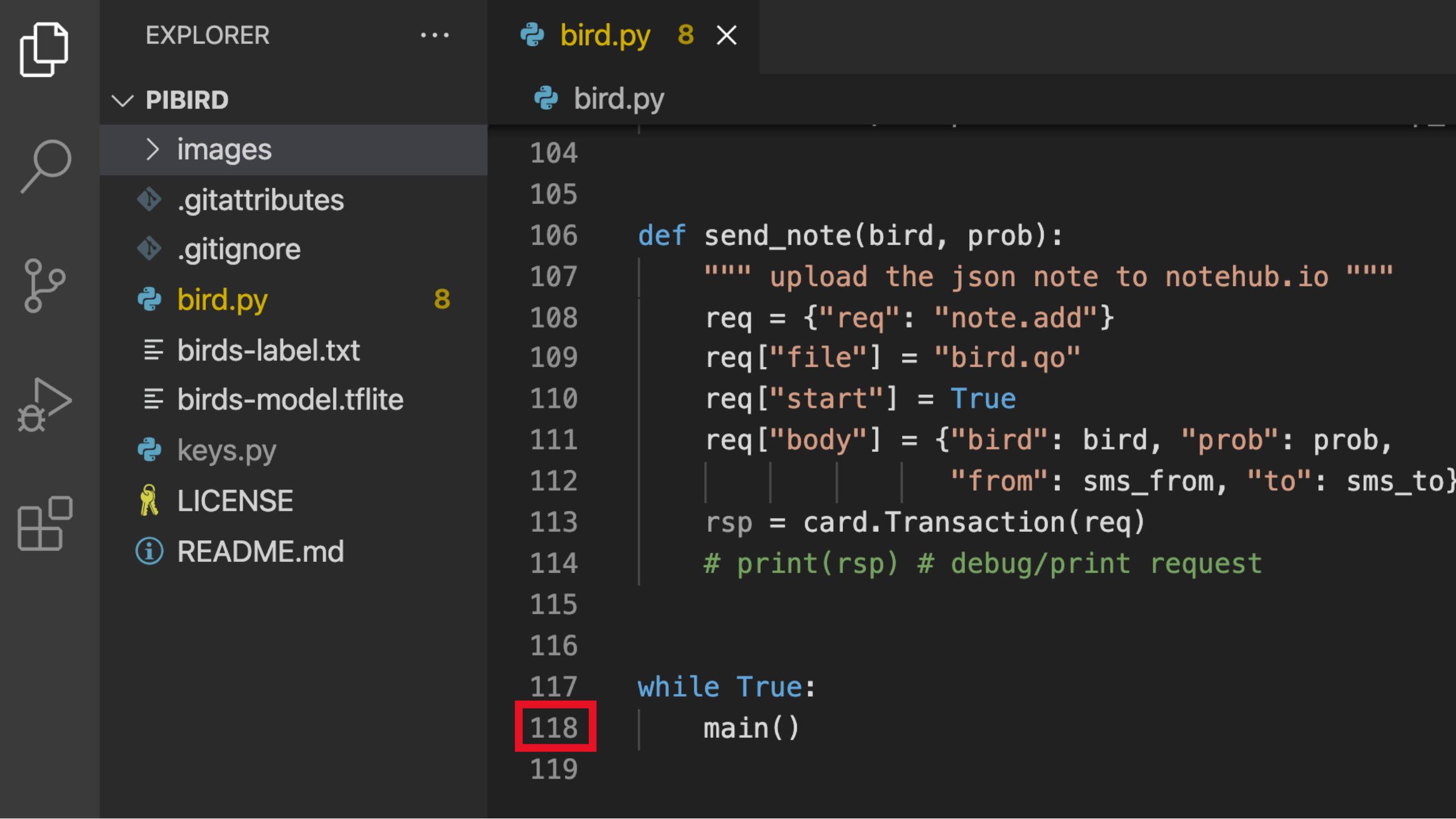
Pi Camera
Snap a quick picture of the bird

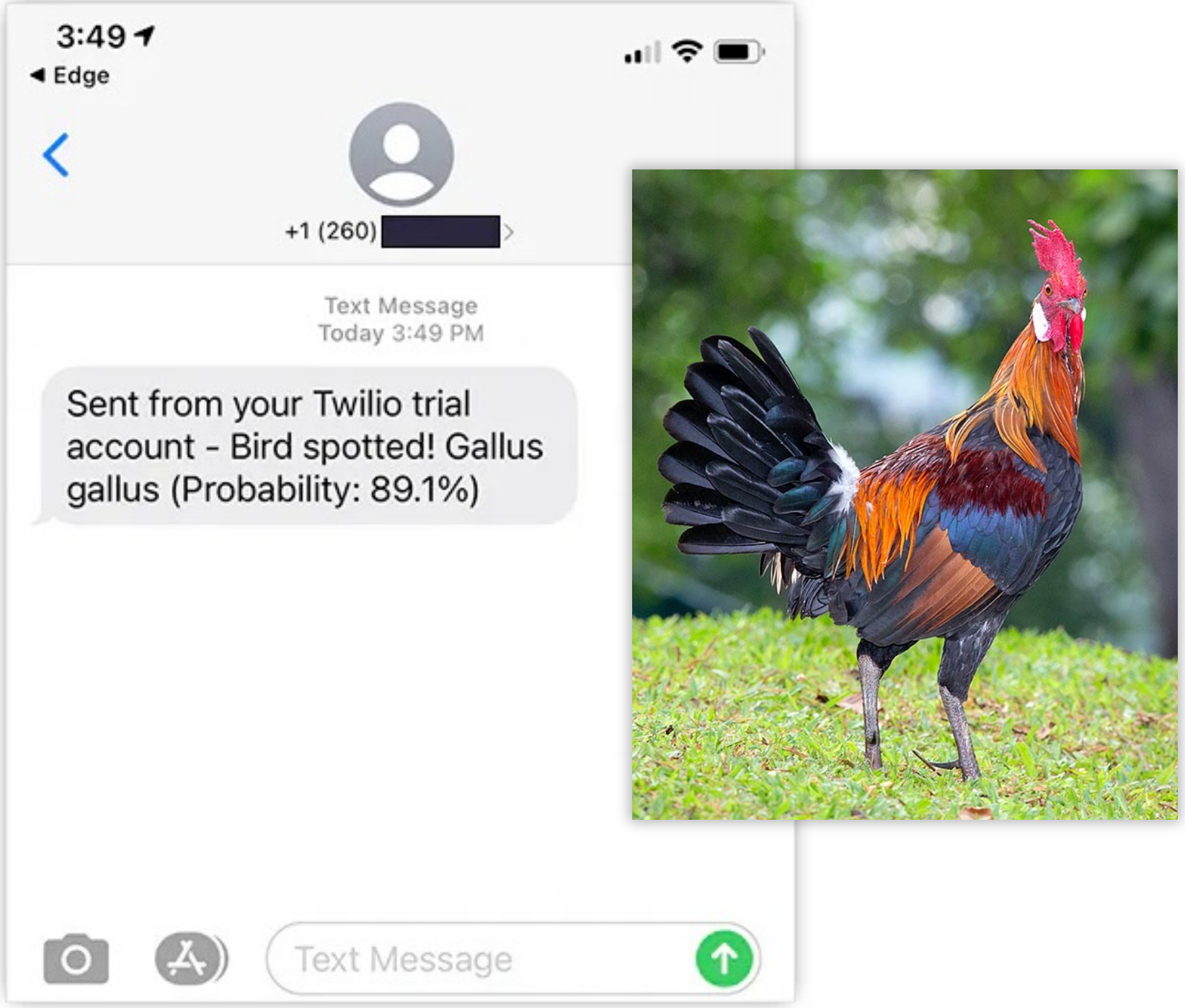
TFLite
Use an ML model for birds around the world

To the Cloud
Send data about the bird to Notehub.io

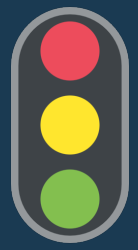


Twilio SMS
Notehub.io securely routes the data to Twilio for an SMS


EXPLORER
PIBIRD
images
.gitattributes
.gitignore
bird.py 8
birds-label.txt
birds-model.tflite
keys.py
LICENSE
README.md
bird.py 8
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 "from": sms_from, "to": sms_to}
113 rsp = card.Transaction(req)
114 # print(rsp) # debug/print request
115
116
117 while True:
118 main()
119



blues wireless



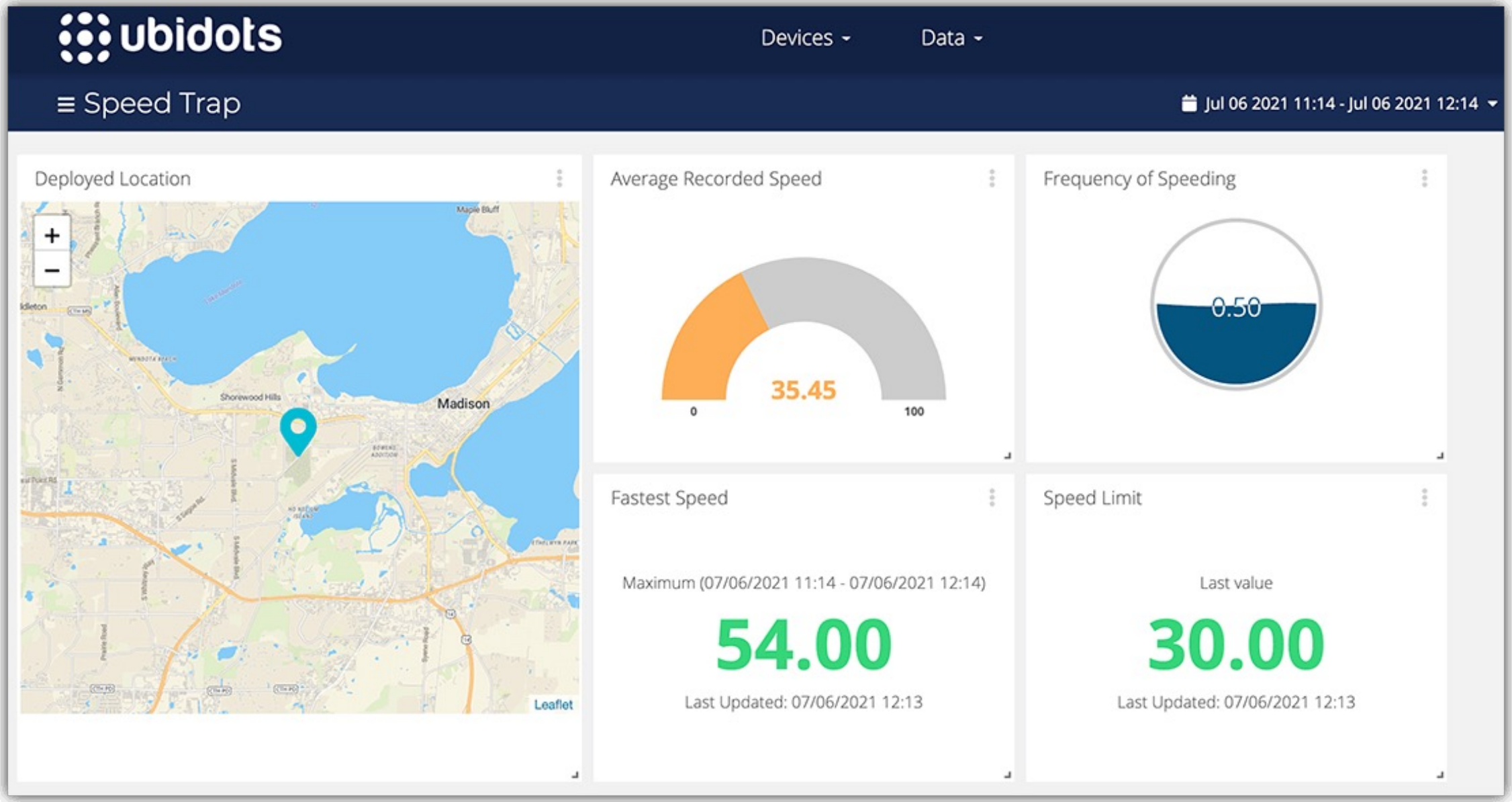
Build a Portable Speed Trap

Portable "Speed Trap"

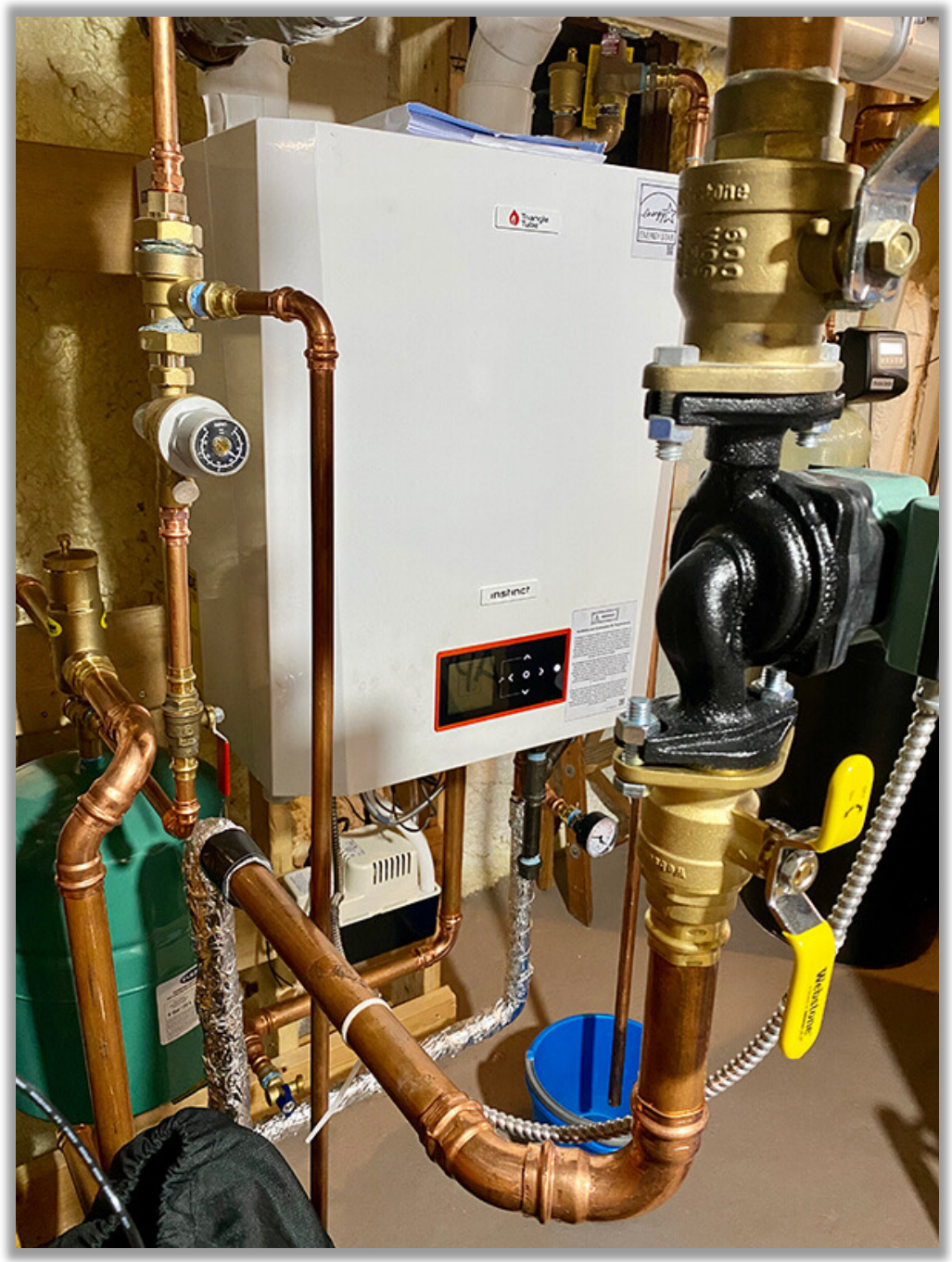
- Uses **ML** to identify vehicles on the road
- Uses a **doppler radar sensor** to gather speed
- Uses a **Notecard** to relay aggregated data to cloud
- Uses a **cloud-based dashboard** for reporting

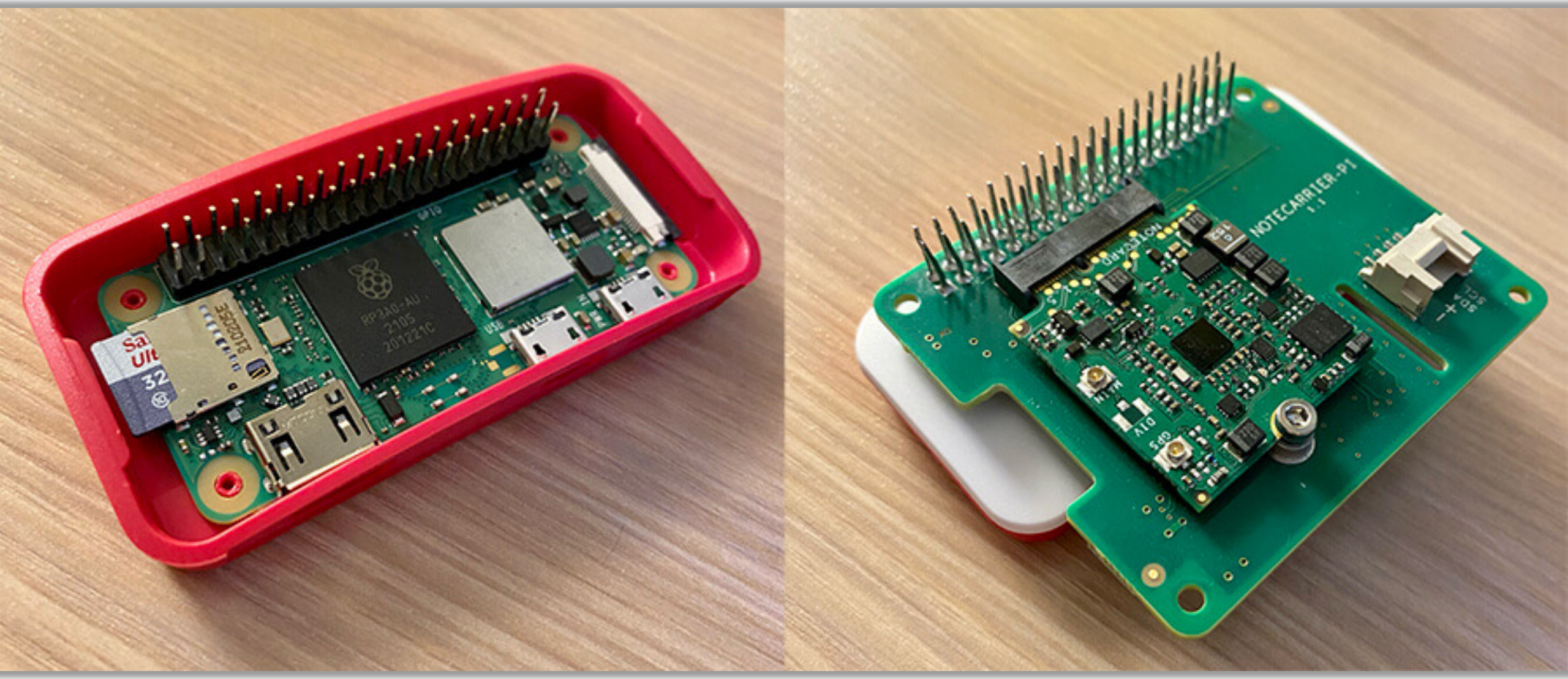

car
SPEED LIMIT 30
NO STOPPING STANDING
MADISON
POLICE
POLICE
E8634

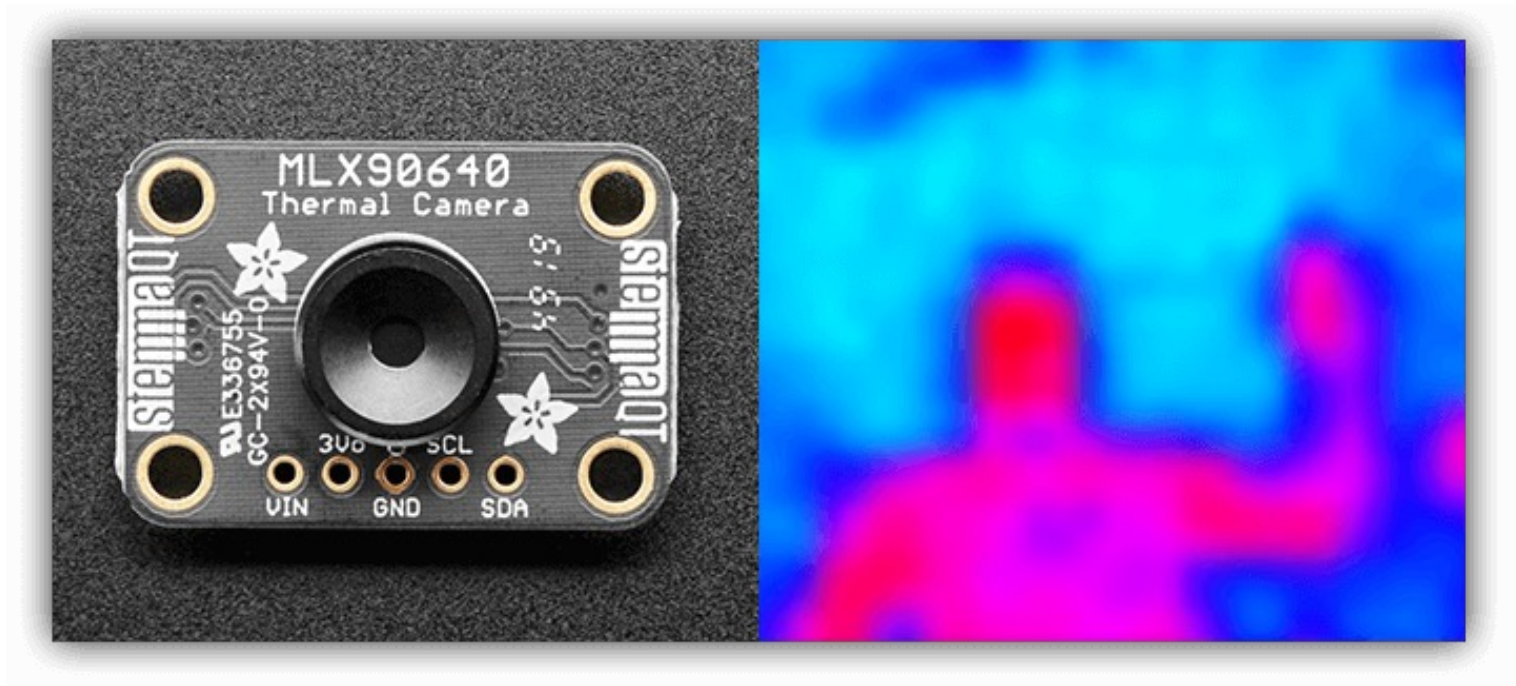

OmniPreSense
OPS243
RX
TX
NOTECARRIER-P1

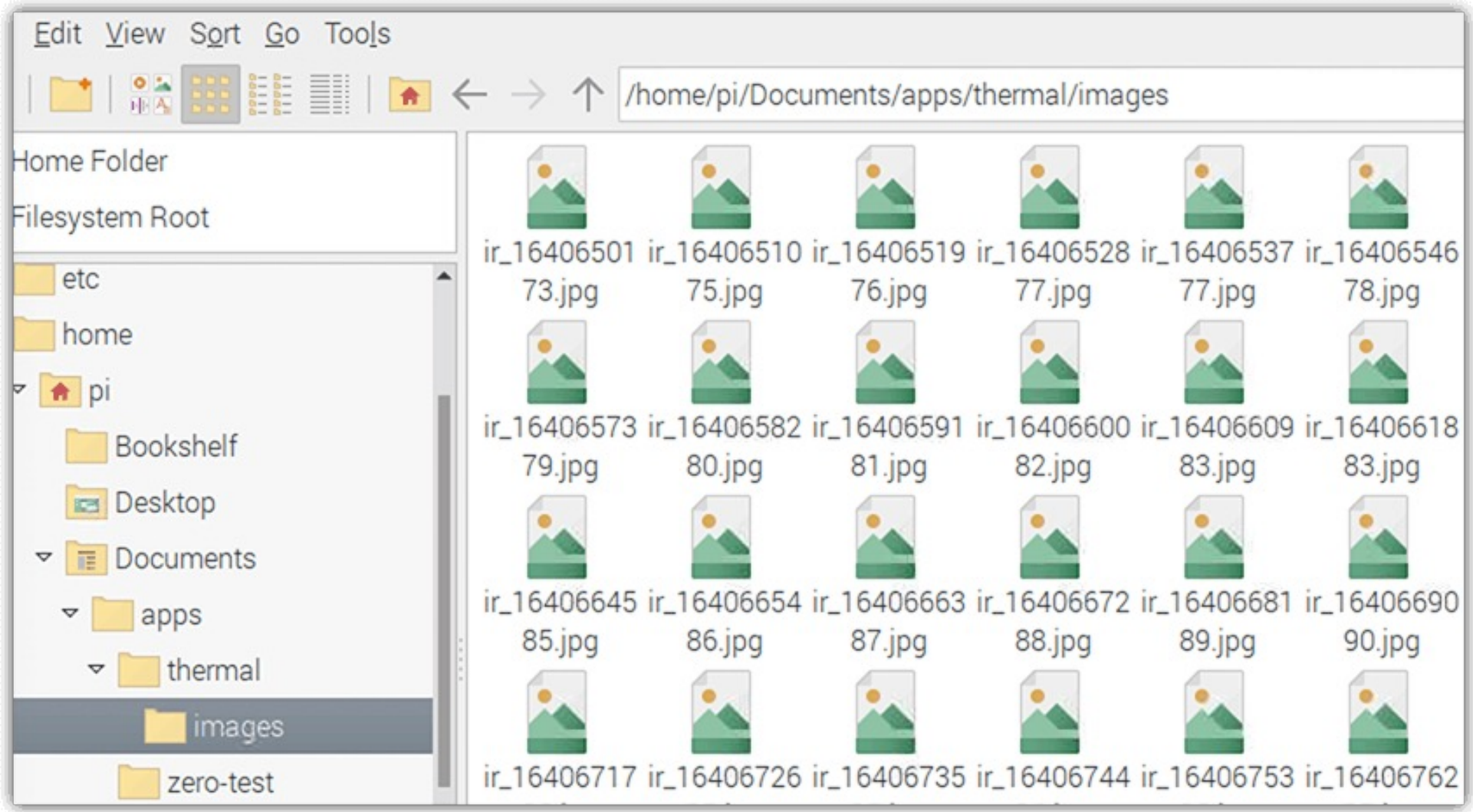

ubidots
Devices
Data
Speed Trap
Jul 06 2021 11:14 - Jul 06 2021 12:14
Deployed Location
Shorewood Hills
Madison
Leaflet
Average Recorded Speed
35.45
0
100
Frequency of Speeding
0.50
Fastest Speed
Maximum (07/06/2021 11:14 - 07/06/2021 12:14)
54.00
Last Updated: 07/06/2021 12:13
Speed Limit
Last value
30.00
Last Updated: 07/06/2021 12:13

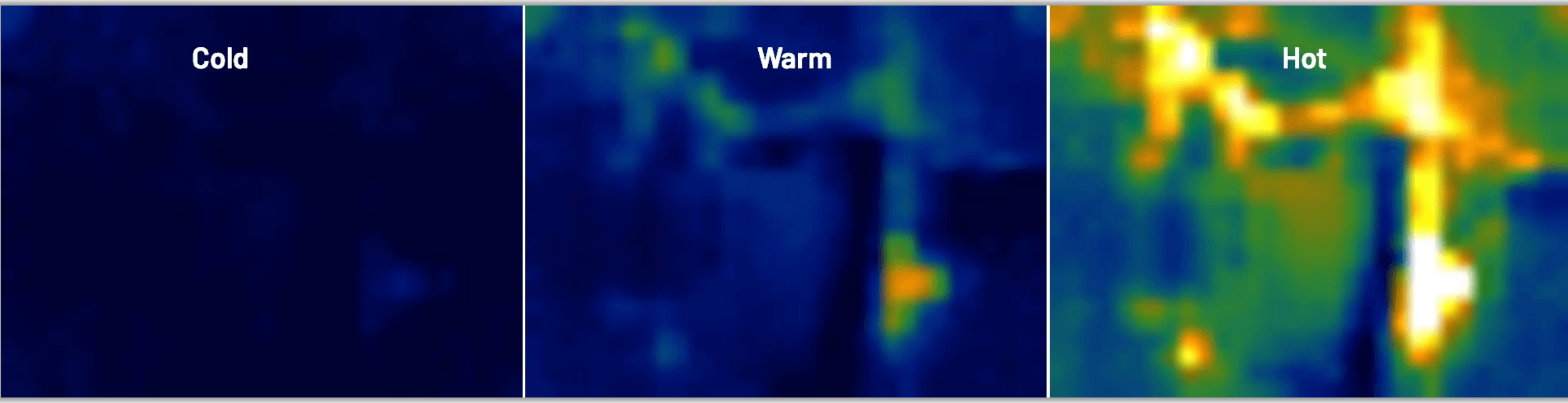
Thermal Image Anomaly Detection

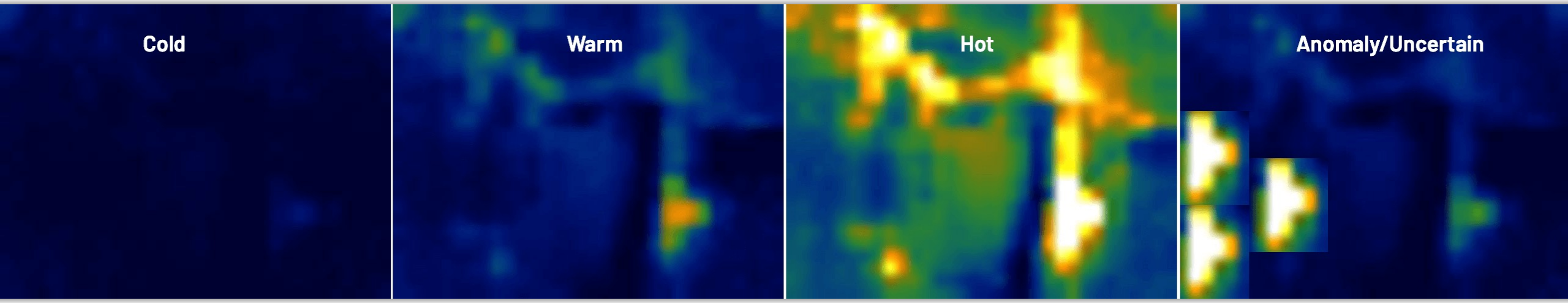

instinct

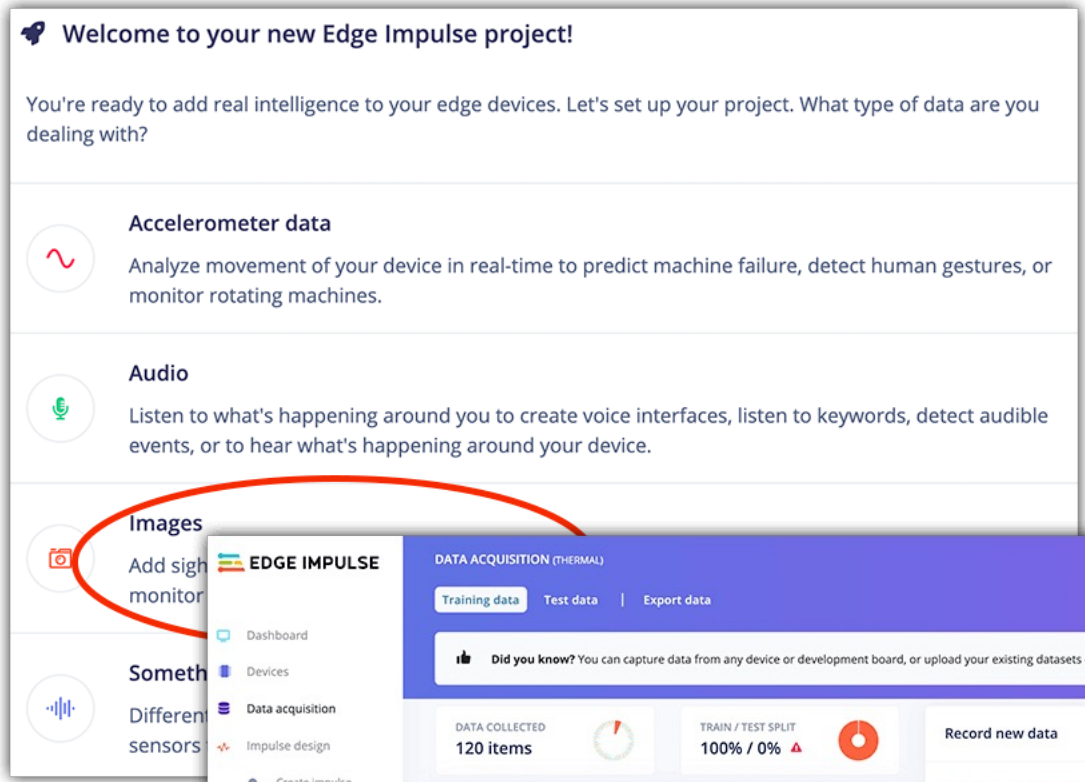

RP3A0-AU
2105
2012210
NOTECARRIER-P1
V1.1
NOTECARD
A10
GPS


MLX90640
Thermal Camera
stemma QT
stemma QT
E336755
GC-2X94V1-0
3V3
SCL
VIN
GND
SDA


Edit View Sort Go Tools
/home/pi/Documents/apps/thermal/images
Home Folder
Filesystem Root
etc
home
pi
Bookshelf
Desktop
Documents
apps
thermal
images
zero-test
ir_1640650173.jpg
ir_1640651075.jpg
ir_1640651976.jpg
ir_1640652877.jpg
ir_1640653777.jpg
ir_1640654678.jpg
ir_1640657379.jpg
ir_1640658280.jpg
ir_1640659181.jpg
ir_1640660082.jpg
ir_1640660983.jpg
ir_1640661883.jpg
ir_1640664585.jpg
ir_1640665486.jpg
ir_1640666387.jpg
ir_1640667288.jpg
ir_1640668189.jpg
ir_1640669090.jpg
ir_16406717
ir_16406726
ir_16406735
ir_16406744
ir_16406753
ir_16406762


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 sigh
monitor
Someth
Differen
sensors

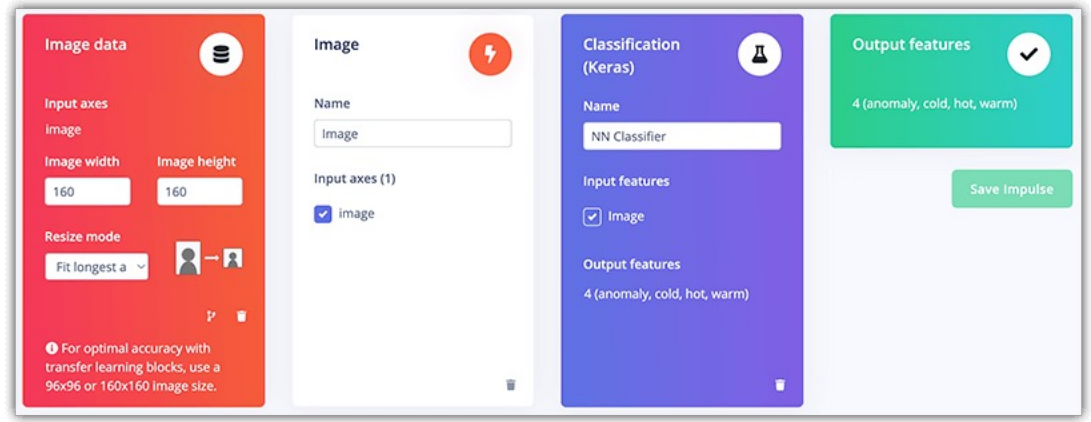
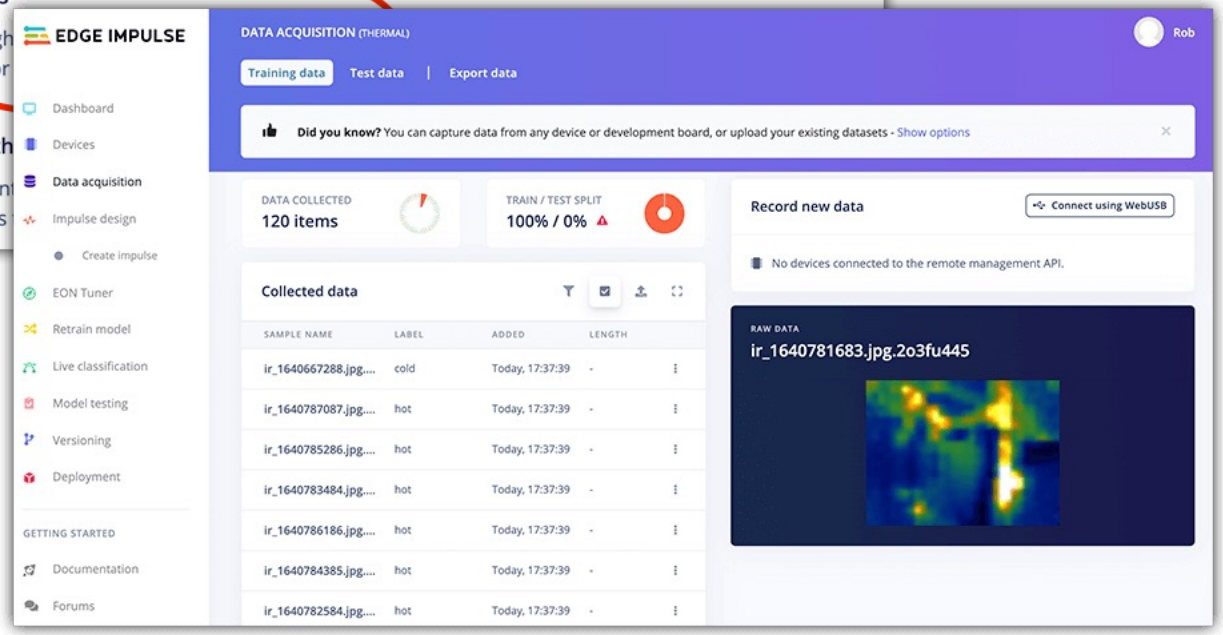
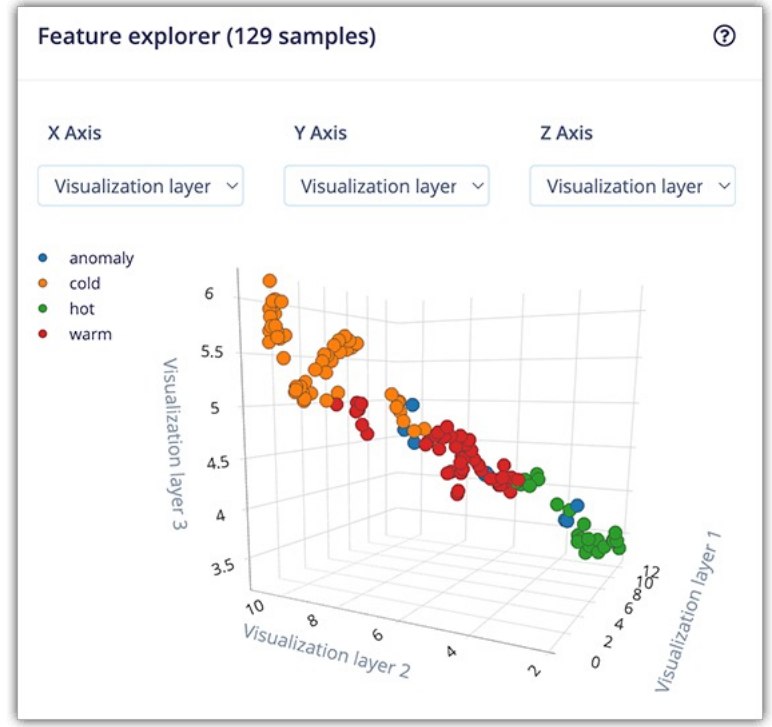
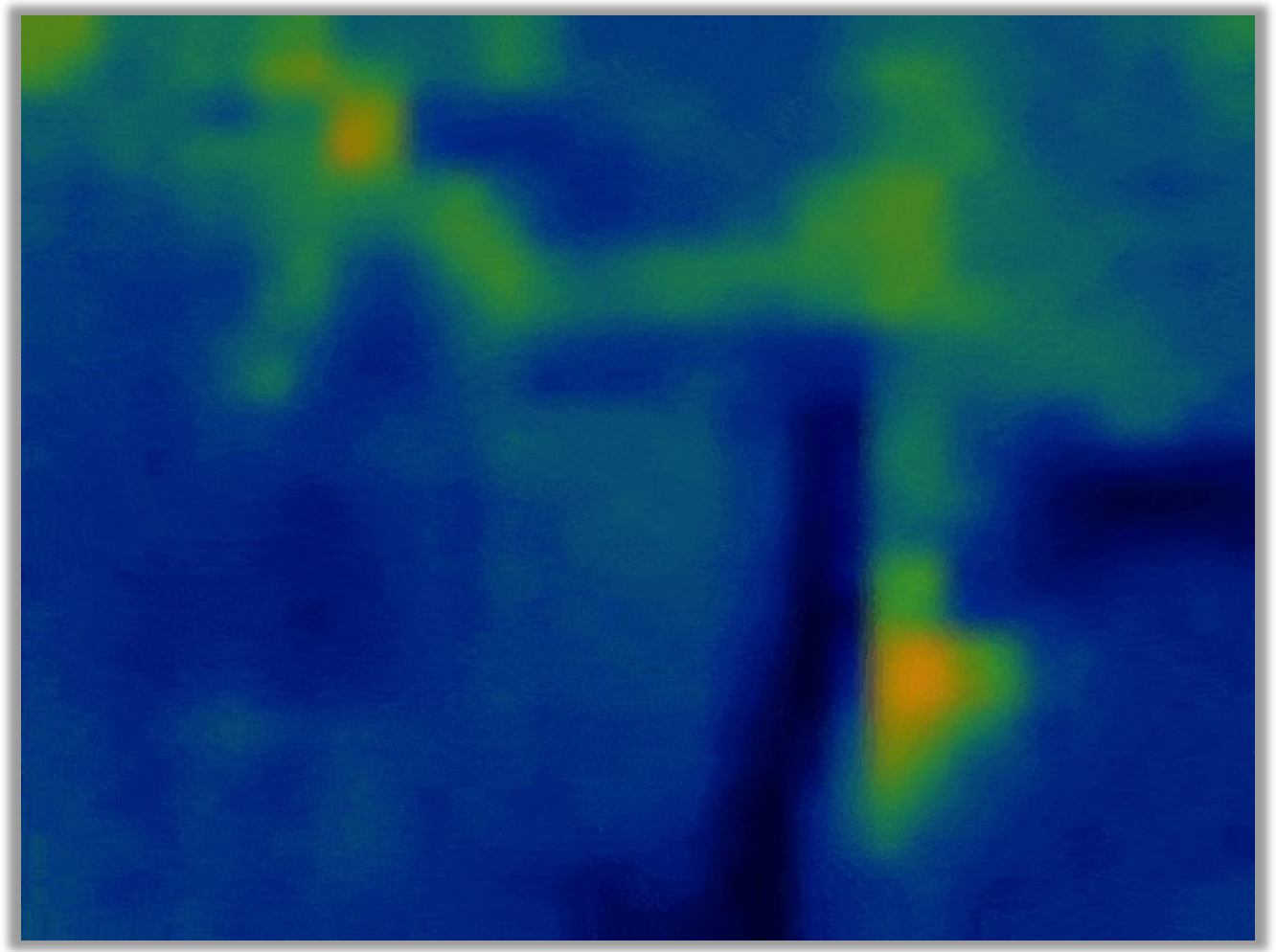


Image data
Input axes
image
Image width
160
Image height
160
Resize mode
Fit longest a
For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.
Image
Name
Image
Input axes (1)
image
Classification (Keras)
Name
NN Classifier
Input features
Image
Output features
4 (anomaly, cold, hot, warm)
Output features
4 (anomaly, cold, hot, warm)
Save Impulse

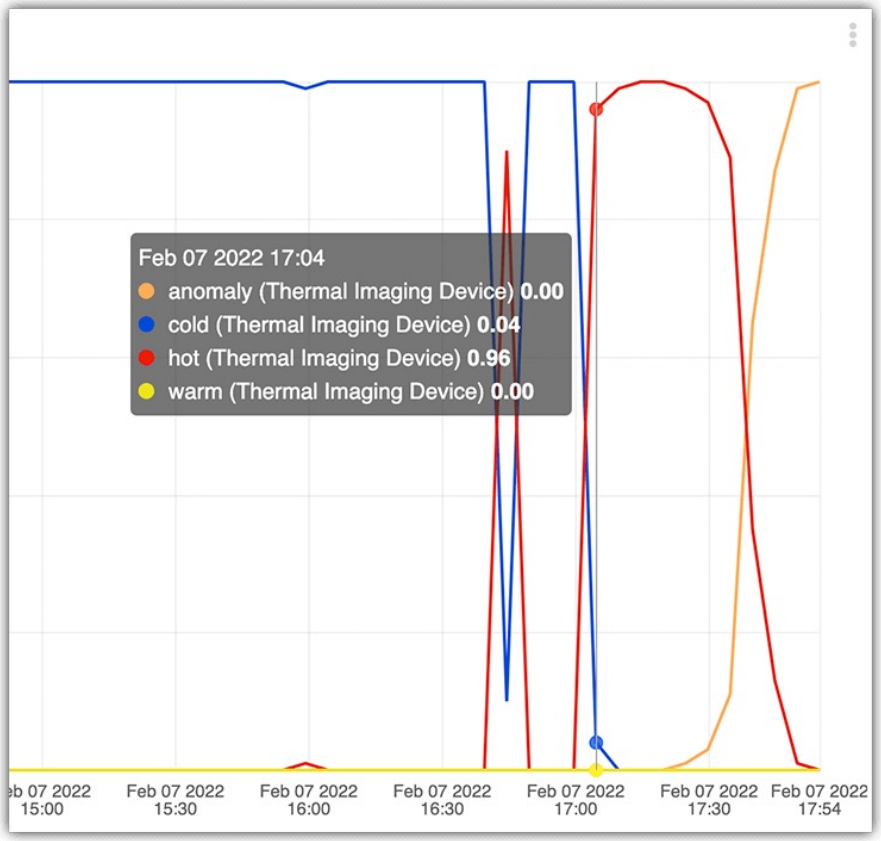

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
DATA ACQUISITION (THERMAL)
Rob
Training data
Test data
Export data
Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options
DATA COLLECTED
120 items
TRAIN / TEST SPLIT
100% / 0%
Record new data
Connect using WebUSB
No devices connected to the remote management API.
Collected data
SAMPLE NAME
LABEL
ADDED
LENGTH
ir_1640667288.jpg....
cold
Today, 17:37:39
ir_1640787087.jpg....
hot
Today, 17:37:39
ir_1640785286.jpg....
hot
Today, 17:37:39
ir_1640783484.jpg....
hot
Today, 17:37:39
ir_1640786186.jpg....
hot
Today, 17:37:39
ir_1640784385.jpg....
hot
Today, 17:37:39
ir_1640782584.jpg....
hot
Today, 17:37:39
RAW DATA
ir_1640781683.jpg.2o3fu445


Feature explorer (129 samples)
X Axis
Visualization layer
Y Axis
Visualization layer
Z Axis
Visualization layer
anomaly
cold
hot
warm
Visualization layer 3
Visualization layer 2
Visualization layer 1



ubidots

twilio



```
# 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
                  model_info['pro
            labels = model_info['

            # grab image from the
            filename = thermal.ta
            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