tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

“Creating individualized solutions for industrial-grade and environmental problems with TinyML”

Kutluhan Aktar – Edge Impulse Ambassador

July 16, 2023

TINY ML

www.tinyML.org

Thank you, **tinyML Strategic Partners**, for committing to take tinyML to the next Level, together

AIZIP
ANALOG DEVICES AHEAD OF WHAT'S POSSIBLE™
ARDUINO
arm AI
brainchip
EDGE IMPULSE
GREENWAVES TECHNOLOGIES
Grovety Inc.
IBM
imagimob
Infineon
innatera
Microsoft
Nota AI
NXP
POLYN TECHNOLOGY
Qeexo
Qualcomm
RENESAS
Schneider Electric
SensiML™
SONY
SILICON LABS
ST life.augmented
synaptics
SYNTIANT

Executive Strategic Partners

Qualcomm
AI research

Advancing AI research to make efficient AI ubiquitous

Power efficiency

Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization

Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning

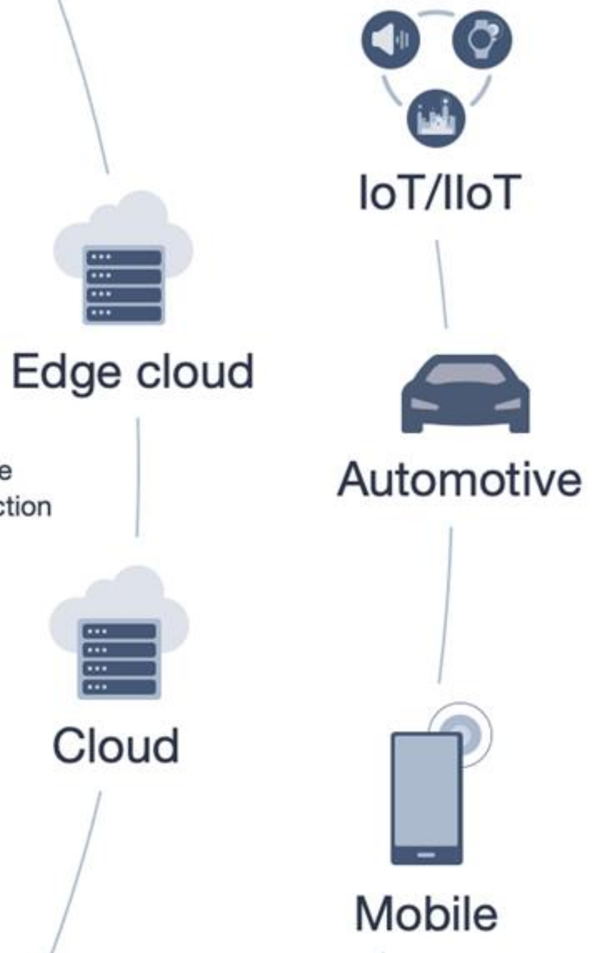
Robust learning through minimal data, unsupervised learning, on-device learning

A platform to scale AI across the industry

Perception
Object detection, speech recognition, contextual fusion

Reasoning
Scene understanding, language understanding, behavior prediction

Action
Reinforcement learning for decision making



Accelerate Your Edge Compute

SYNTIANT

Making Edge AI A Reality

www.syntiant.com

Platinum Strategic Partners

Renesas is enabling the next generation of AI-powered solutions that will revolutionize every industry sector.

renesas.com

DEPLOY VISION AI AT THE EDGE AT SCALE

Gold Strategic Partners

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Where what if becomes what is.

Witness potential made possible at analog.com.

Build the Future of tinyML

on arm

TINY ML TALKS webcast

EDGE IMPULSE

The Leading Development Platform for Edge ML

edgeimpulse.com

Driving decarbonization and digitalization. Together.

Infineon serving all target markets as
Leader in Power Systems and IoT

www.infineon.com

innatera

TALKS
webcast

NEUROMORPHIC INTELLIGENCE FOR THE SENSOR-EDGE

www.innatera.com

tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

Microsoft

TINY ML

www.tinyML.org

The Right Edge AI Tools Can Make or Break Your Next Smart IoT Product

STMicroelectronics provides extensive solutions to make tiny Machine Learning easy

life.augmented

www.st.com/ai

synaptics

ENGINEERING EXCEPTIONAL EXPERIENCES

We engineer exceptional experiences for consumers in the home, at work, in the car, or on the go.

www.synaptics.com

TINY ML TALKS webcast

Silver Strategic Partners

AIZIP
brainchip
GREENWAVES TECHNOLOGIES
Grovety Inc.

IBM
imagimob
Nota AI
NXP
POLYN TECHNOLOGY

Qeexo
Schneider Electric
SILICON LABS

Join Growing tinyML Communities:

15.5k members in
49 Groups in 41 Countries

tinyML - Enabling ultra-low Power ML at the Edge

https://www.meetup.com/tinyML-Enabling-ultra-low-Power-ML-at-the-Edge/

4k members
&
12.4k followers

The tinyML Community

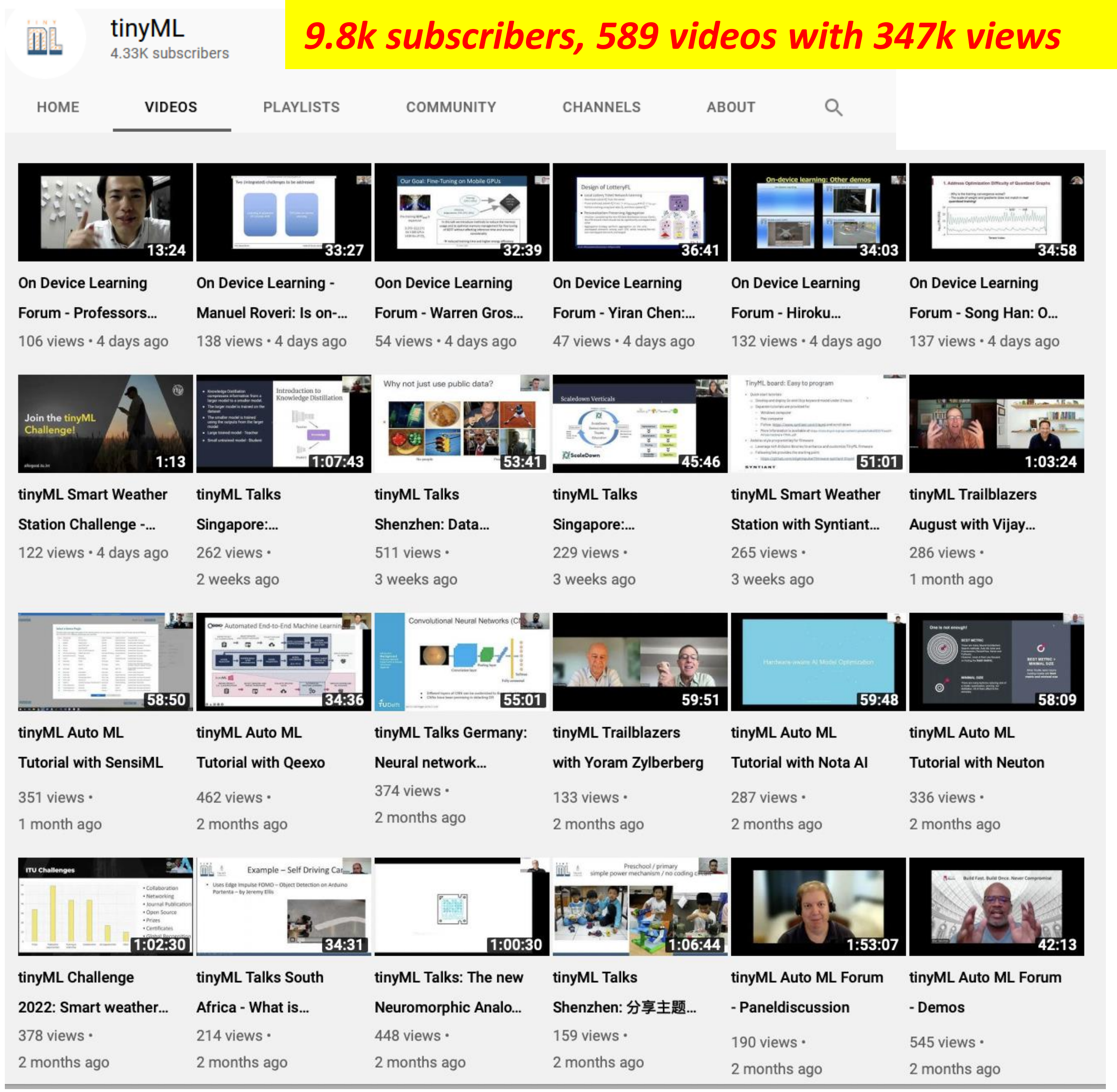
https://www.linkedin.com/groups/13694488/

TINY ML TALKS webcast

Subscribe to
tinyML YouTube Channel
for updates and notifications
(including this video)

www.youtube.com/tinyML

YouTube



Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums

youtube.com/tinyml

Please use the Q&A window for your questions

Kutluhan Aktar

I am a self-taught developer and maker who enjoys contemplating proof-of-concept AIoT projects in various fields. I was an aspiring physics major, but I decided to drop out of university in order to follow my vocation to be an independent researcher and build original projects from scratch. With the help of lots of innovative companies, I have been able to keep devising inspiring projects and realize my ideas in recent years as an occupation.

Creating **individualized solutions** for industrial-grade and environmental problems with **TinyML**

Kutluhan Aktar
Self-Taught Developer
Edge Impulse Ambassador
Maker
Independent Researcher

TINY ML TALKS webcast

Challenges with machine learning and large-scale problems

- Defining all possible use case scenarios
- Grasping the extent of the targeted problem
- Interminable workloads due to the lack of talent
- Exorbitant costs due to hardware requirements
- Constructing miscellaneous data sets
- Inaccessible and inaccurate data
- Adapting to sporadic conditions
- Building complex algorithms
- Cloud computing and storage space
- Meagre funds and limited market opportunity

Advantages of creating individualized solutions for large-scale problems with TinyML

- Focusing on a single use case scenario of a large-scale problem
- Pinpointing and refining solution methods
- Manageable objectives
- Considerably low budgets and affordable hardware
- Constricted data sets
- Efficient data collection
- Consistent and stable instances
- Optimized algorithms for simple tasks
- Minimum latency and power consumption
- No need for profit from continuing operations

Assets for tailoring solutions with TinyML

- Specialized development boards for edge AI
- Wi-Fi, Bluetooth, and LoRaWAN-enabled edge devices
- Pre-configured sensors and cameras with supported libraries
- Single-board computers (SBCs)
- Edge AI platforms
- Cloud computing services
- Optimized algorithms and model architectures
- HTTP and REST APIs
- SDKs and development tools
- Open-source projects and tutorials

My approach to creating individualized solutions for large-scale problems

- Perusing research papers to fathom the targeted industrial-grade, environmental, or health-related problem
- Choosing the most suitable transmission method
- Devising a budget-friendly and accessible device capable of collecting data and running inferences
- Finding a local area or designing a controlled environment for data collection
- Creating a distinctive data set with notable validity
- Building and training a neural network model with Edge Impulse
- Conducting experiments with the deployed model in the field
- Developing an application or interface to inform the user of the model detection results

IoT AI-driven Tree Disease Identifier w/ Edge Impulse & MMS

Issue

Due to climate change and excessive deforestation, trees and plants are becoming more susceptible to contagious diseases.

Most tree diseases are fungal and instigated mainly by drought, high carbon dioxide levels, overcrowding, and damage to stem or roots.

Since trees are crucial for pollination, spreading tree diseases can cause crop yield loss, animal deaths, widespread infectious epidemics, and even land degradation due to soil erosion.

Various environmental factors can cause trees to be stressed and catch a highly contagious disease.

Tree diseases can engender unrecoverable damage to forests, farms, and arable lands.

Solution

Detect tree diseases before permeating forests to avoid their detrimental effects.

Provide prescient warnings regarding potential tree diseases and environmental factors to avoid hazardous outcomes related to forest degradation and crop yield loss.

Collect data from trees infected with different diseases in order to train an object detection model with notable validity.

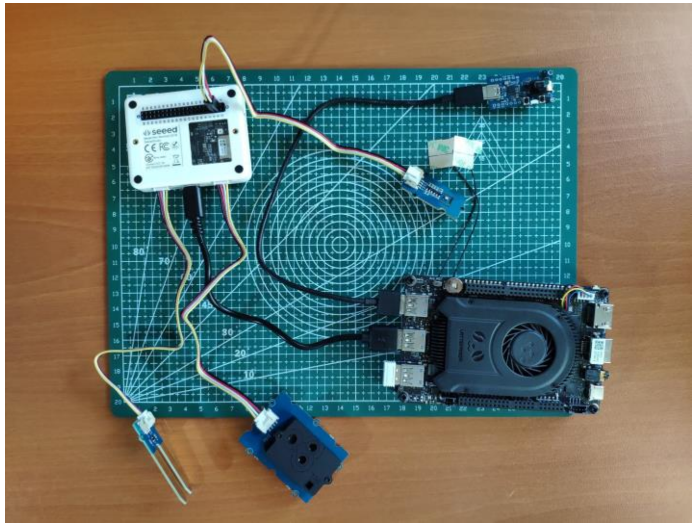
Inform the user of the detection results via MMS remotely after running the object detection model.

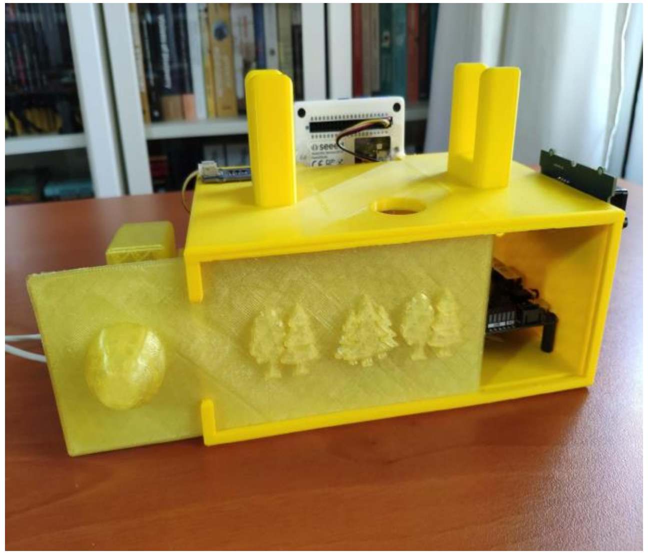
Design a 3D printable case to make the device as robust and sturdy as possible while operating outdoors.

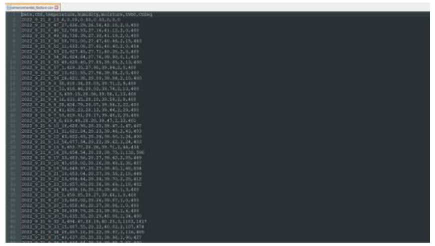
Collecting environmental factors w/ Wio Terminal

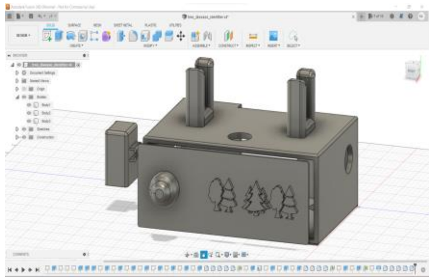
- Configured Wio Terminal on Arduino IDE
- Assembled the forest-themed case
- Programmed Wio Terminal to save the collected environmental factors to a CSV file on the SD card
- Displayed the collected environmental factors as a histogram
- Utilized Wio Terminal to send commands to LattePanda 3 Delta via serial communication

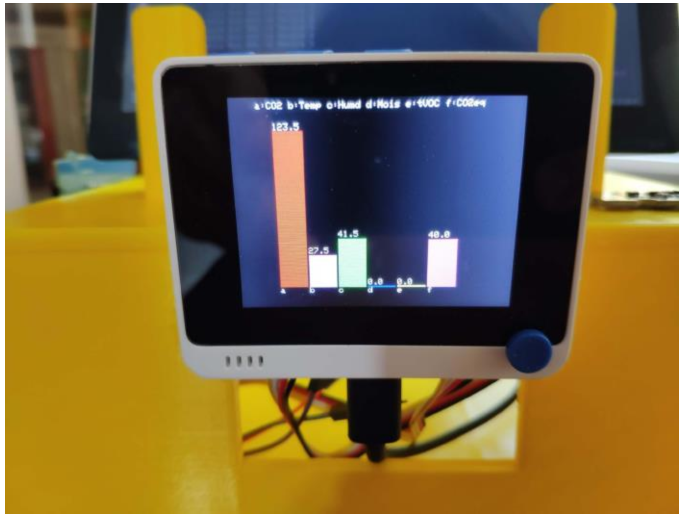
Date
CO2
Temperature
Humidity
Moisture
tVOC (total volatile organic compounds)
CO2eq (carbon dioxide equivalent)









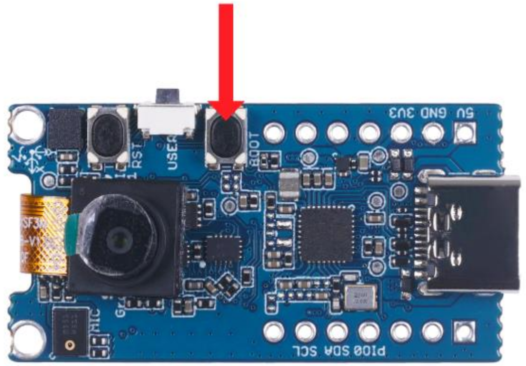


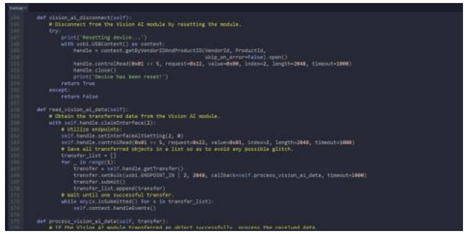
TINY ML TALKS webcast

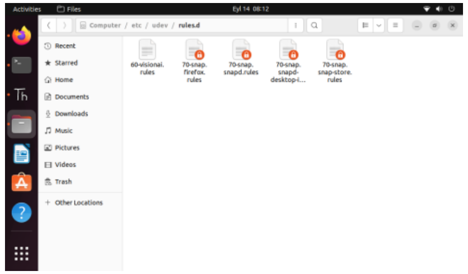
Capturing images of infected trees w/ Grove Vision AI module

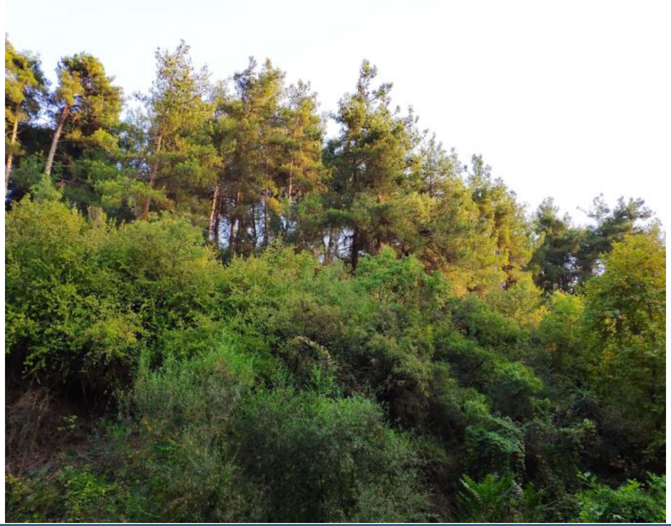
- Updated the Vision AI module firmware to capture images in Python
- Utilized LattePanda 3 Delta (Ubuntu) to store the captured images
- Programmed LattePanda 3 Delta to show a real-time video stream generated by the Vision AI module
- Collected various infected tree images with different foliar and bark tree diseases in a forest near my hometown
- Obtained my data set after capturing images nearly two weeks

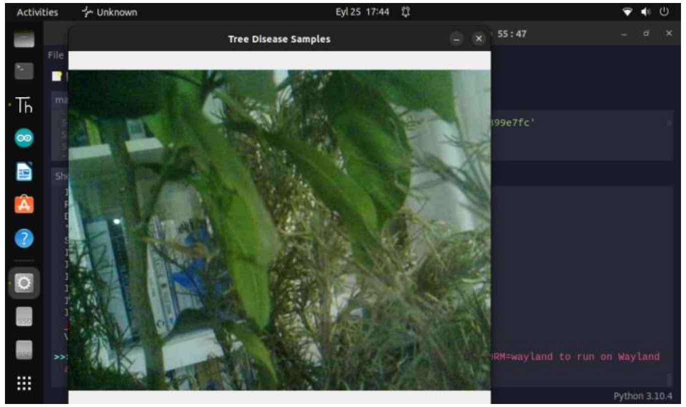
Leaf Rusts
Leaf Spots
Leaf Blisters
Powdery Mildew
Needle Rusts
Needle Casts
Tar Spots
Rusts
Anthracnose

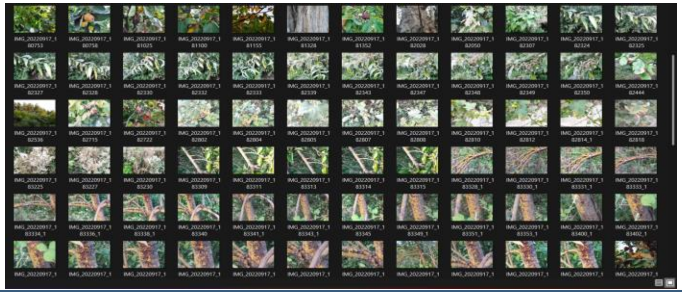










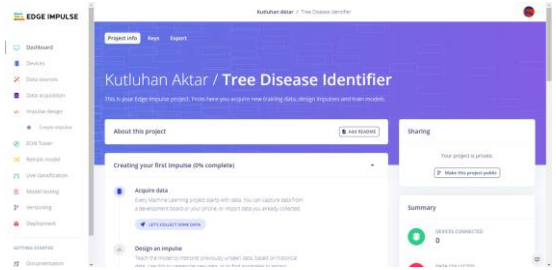


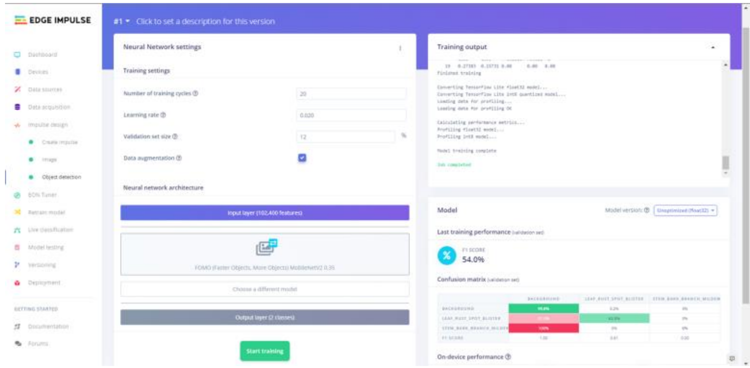
Building an object detection (FOMO) model with Edge Impulse

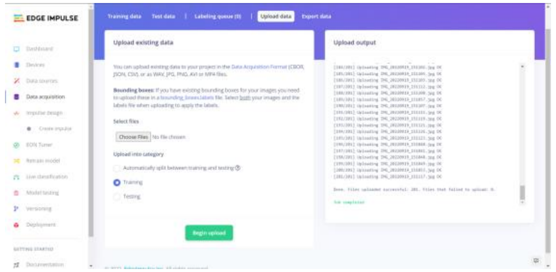
leaf_rust_spot_blister

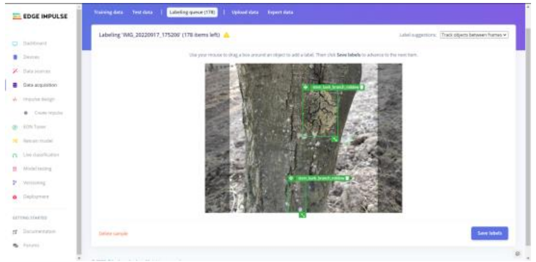
stem_bark_branch_mildew

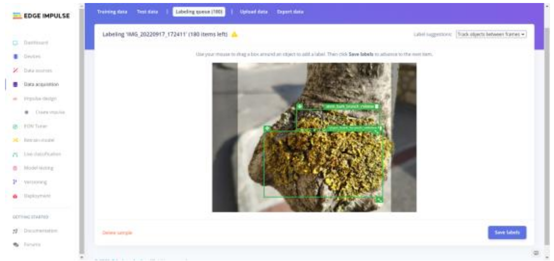
- Uploaded the collected images (samples)
- Labeled targeted objects on the samples
- Trained the FOMO model on contagious tree diseases
- Evaluated the model accuracy
- Deployed the trained model as a Linux x86_64 application
- Set up the deployed model on LattePanda 3 Delta

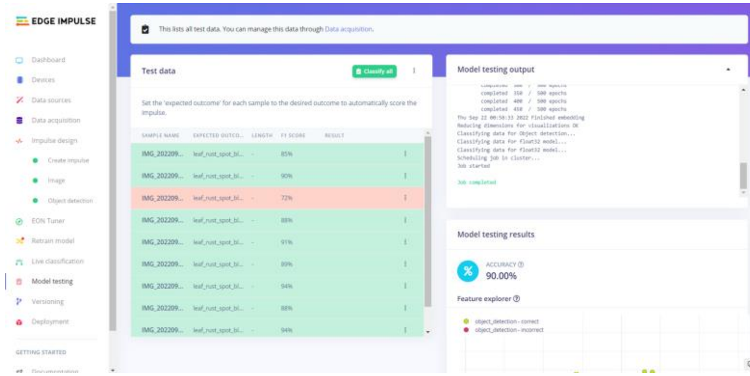


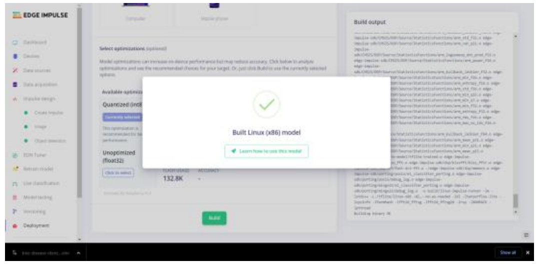


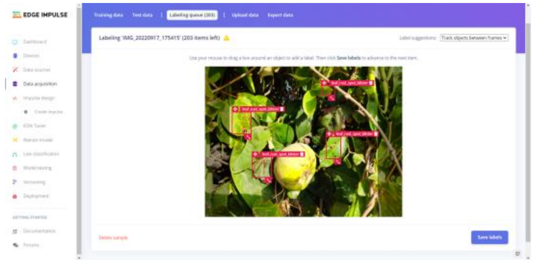


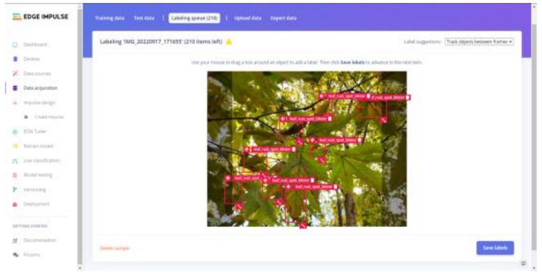






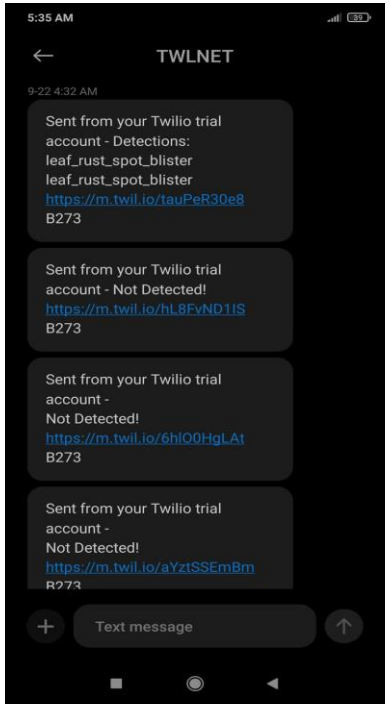


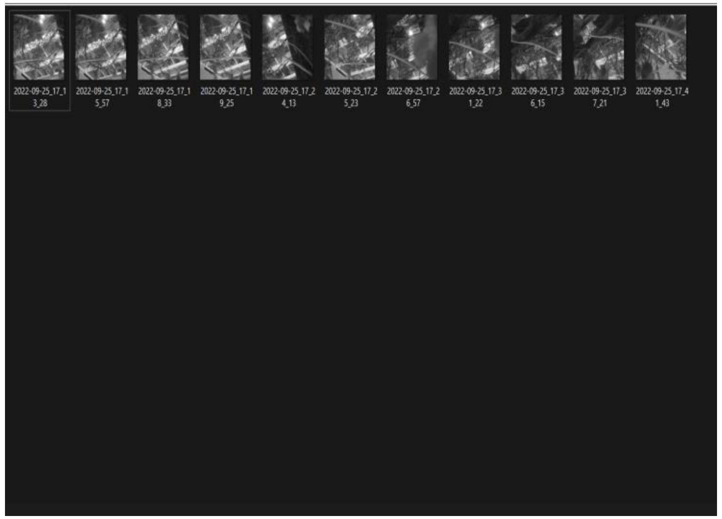


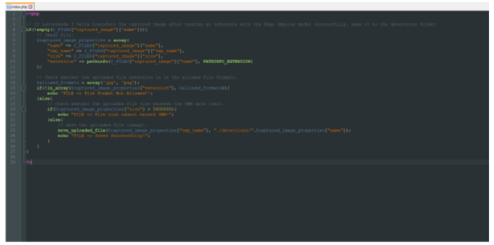


Informing the user of the model detection results via MMS

- Employed LattePanda 3 Delta to run an inference, modify the captured image by adding bounding boxes, and send the modified image via an HTTP POST request
- Created a Twilio account to send MMS
- Developed a web application in PHP to save the modified image transferred by LattePanda 3 Delta
- Then, utilized the application to send an MMS, including the detected classes and the modified image, to the verified phone number via Twilio's API







Further Discussions

By applying object detection models trained on captured infected tree images in detecting potential tree diseases, we can achieve to:

avoid crop yield loss, animal deaths, widespread infectious epidemics	prevent land degradation due to soil erosion	mitigate deforestation	protect wildlife

Project Tutorial

TINY ML TALKS

IoT AI-driven Yogurt Processing & Texture Prediction | Blynk

Issue

Yogurt is produced by bacterial fermentation of milk, which can be of cow, goat, ewe, sheep, etc.

Even though yogurt production and manufacturing look like a simple task, achieving precise yogurt texture (consistency) can be arduous and strenuous since various factors affect the fermentation process.

Most companies employ food (chemical) additives while mass-producing yogurt to maintain its freshness, taste, texture, and appearance, including diluents, water, artificial flavorings, rehashed starch, sugar, and gelatine.

Due to the surge in food awareness and health regulations, companies were coerced into changing their yogurt production methods.

Yogurt production without any additives can be expensive and demanding for small businesses trying to gain a foothold in the dairy industry.

Solution

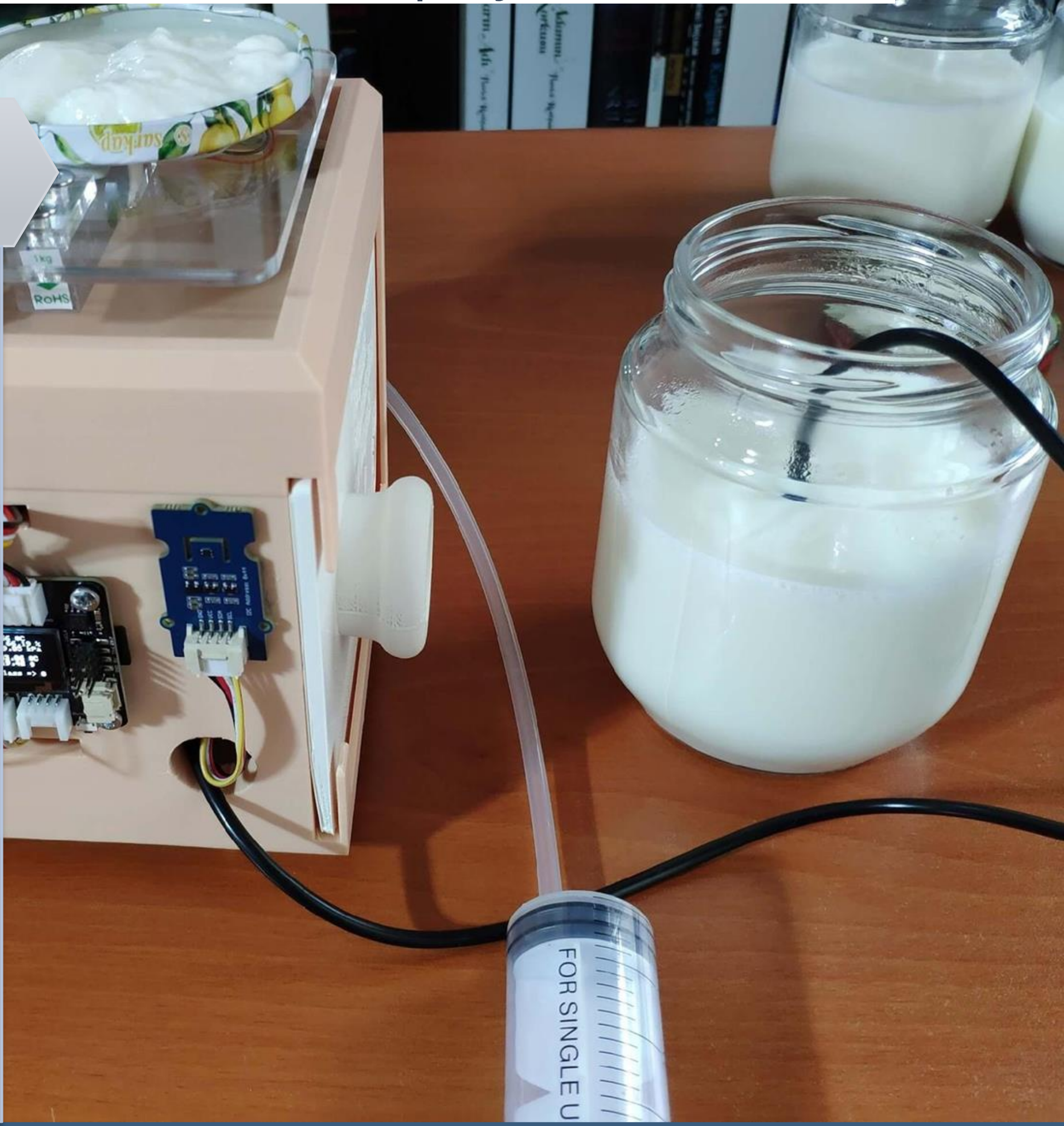
Detect yogurt consistency levels before fermentation.

Assist dairies in reducing the total cost and improving product quality.

Collect data while producing yogurt in different conditions to train a neural network model with notable validity.

Inform the user of the detection results after running the neural network model.

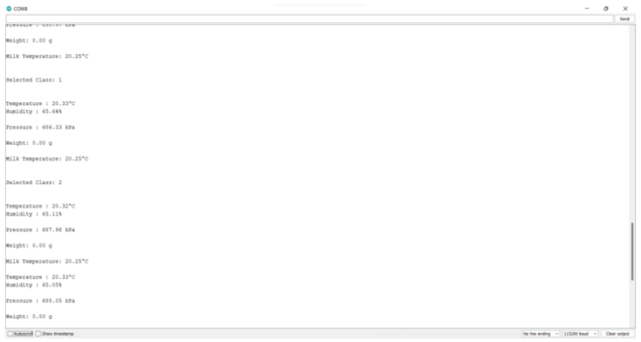
Design a 3D printable case to make the device as sturdy and robust as possible while operating in a dairy.

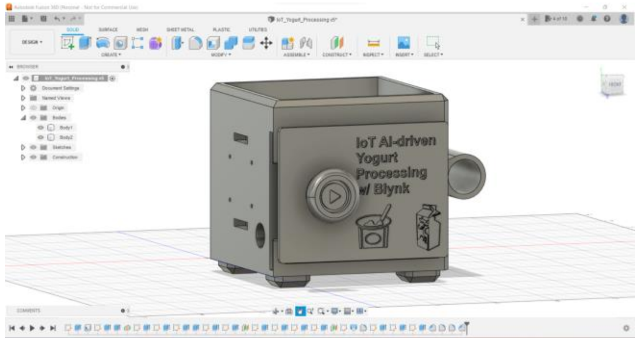


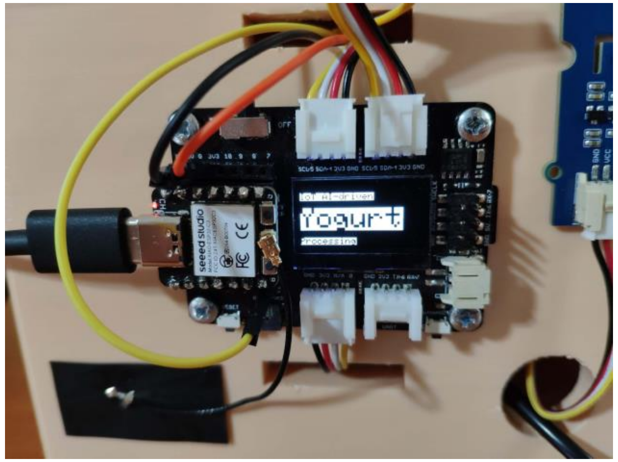
Collecting yogurt processing data w/ XIAO ESP32C3

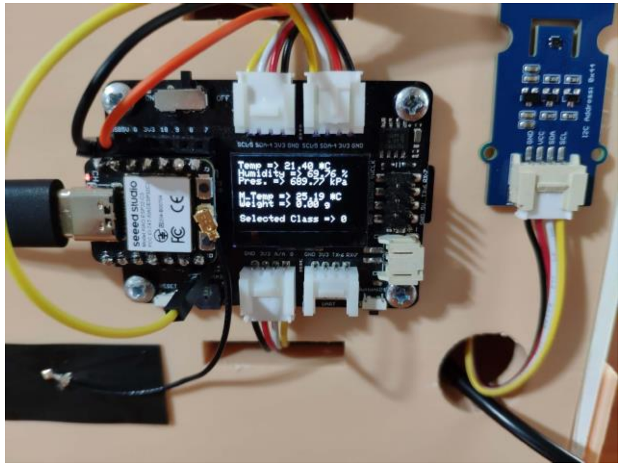
- Configured XIAO ESP32C3 on Arduino IDE
- Assembled the dairy-themed case
- Programmed XIAO ESP32C3 to save the collected environmental factor measurements and the culture (starter) amount to a CSV file on the SD card
- Utilized the built-in button on the XIAO expansion board in two different modes (long press and short press) to select a class and save data records
- Displayed the collected yogurt processing data and the selected class number

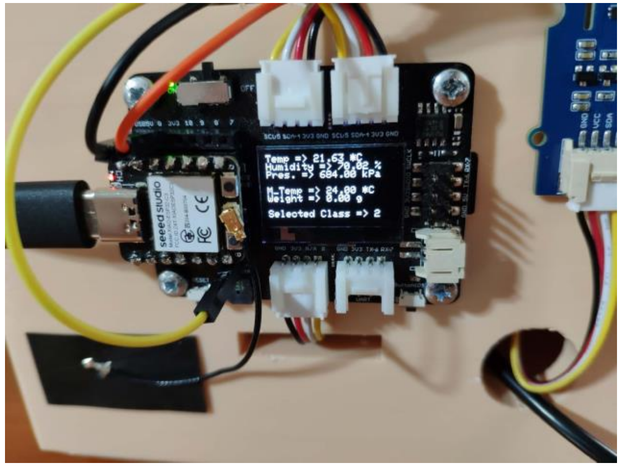
Temperature (°C)
Humidity (%)
Pressure (kPa)
Milk Temperature (°C)
Starter Weight (g)

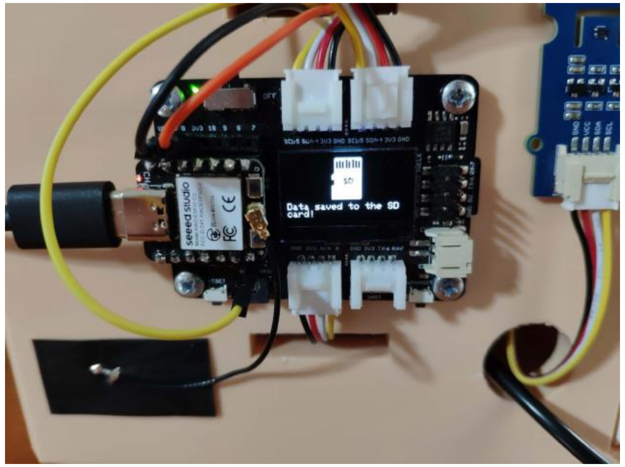








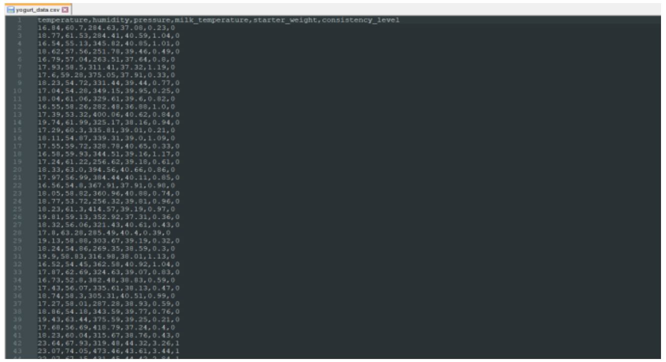




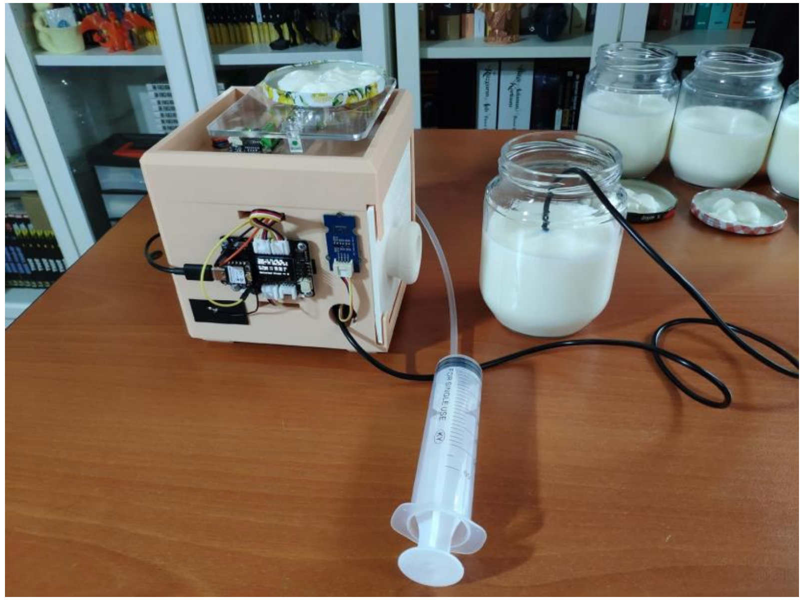
Constructing a data set with notable validity

- Collected yogurt processing data from nearly 30 different batches
- Always used cow milk but changed the milk temperature, yogurt culture amount, and environmental factors while conducting my experiments
- Obtained my data set as a CSV file on the SD card, including labels for each data record

Thinner [0]
Optimum [1]
Curdling [2]


yogurt_data.csv

```
temperature,humidity,pressure,milk_temperature,starter_weight,consistency_level
16.84,60.7,284.63,37.08,0.23,0
18.77,61.53,284.41,40.59,1.04,0
16.54,55.13,345.82,40.85,1.01,0
18.62,57.56,251.78,39.46,0.49,0
16.79,57.04,263.51,37.64,0.8,0
17.93,58.5,311.41,37.32,1.19,0
17.6,59.28,375.05,37.91,0.33,0
18.23,54.72,331.44,39.44,0.77,0
17.04,54.28,349.15,39.95,0.25,0
18.04,61.06,329.61,39.6,0.82,0
16.55,58.26,282.48,36.88,1.0,0
17.39,53.32,400.06,40.62,0.84,0
19.74,61.99,325.17,38.16,0.94,0
17.29,60.3,335.81,39.01,0.21,0
18.11,54.87,339.31,39.0,1.09,0
17.55,59.72,328.78,40.65,0.33,0
16.58,59.93,344.51,39.16,1.17,0
17.24,61.22,256.62,39.18,0.61,0
18.33,63.0,394.56,40.66,0.86,0
17.97,56.99,384.44,40.11,0.85,0
16.56,54.8,367.91,37.91,0.98,0
18.05,58.82,360.96,40.88,0.74,0
18.77,53.72,256.32,39.81,0.96,0
18.23,61.3,414.57,39.19,0.97,0
19.81,59.13,352.92,37.31,0.36,0
18.32,56.06,321.43,40.61,0.43,0
17.8,63.28,285.49,40.4,0.39,0
19.13,58.88,303.67,39.19,0.32,0
18.24,54.86,269.35,38.59,0.3,0
19.9,58.83,316.98,38.01,1.13,0
16.52,54.45,362.58,40.92,1.04,0
17.87,62.69,324.63,39.07,0.83,0
16.73,52.8,382.48,38.83,0.59,0
17.43,56.07,335.61,38.13,0.47,0
18.74,58.3,305.31,40.51,0.99,0
17.27,58.01,287.28,38.93,0.59,0
18.86,54.18,343.59,39.77,0.76,0
19.43,63.44,375.59,39.25,0.21,0
17.68,56.69,418.79,37.24,0.4,0
18.23,60.04,315.67,38.76,0.43,0
23.64,67.93,319.48,44.32,3.26,1
23.07,74.05,473.46,43.61,3.44,1
```



Building a neural network model with Edge Impulse

TINY ML TALKS webcast

- Developed a Python application to scale and preprocess my data set to create individual CSV files for each data record
- Uploaded the formatted samples
- Trained the model on yogurt consistency levels
- Evaluated the model accuracy
- Deployed the trained model as an Arduino library
- Set up the deployed model on XIAO ESP32C3

Thinner.sample_1.csv

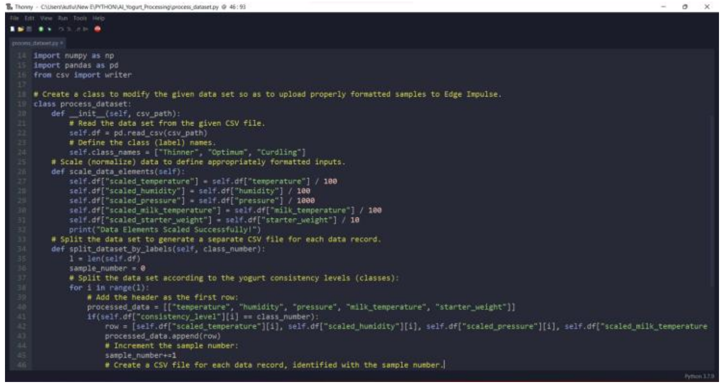
Thinner.sample_2.csv

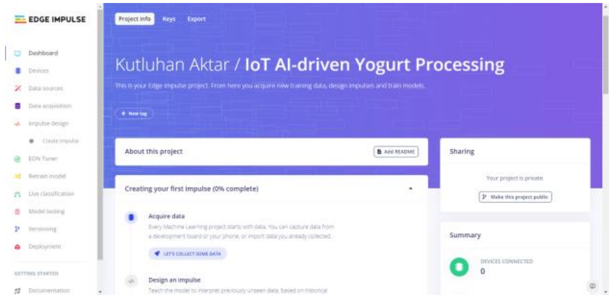
Optimum.sample_1.csv

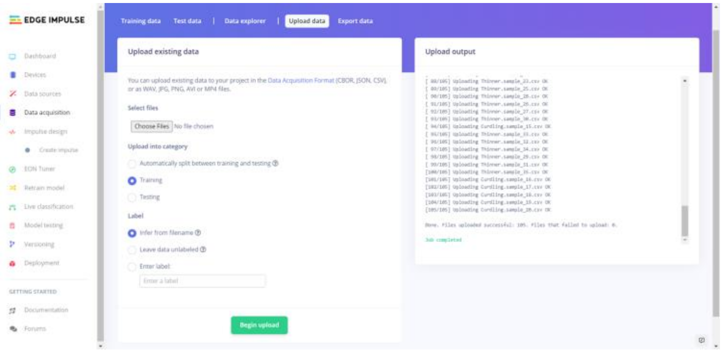
Optimum.sample_2.csv

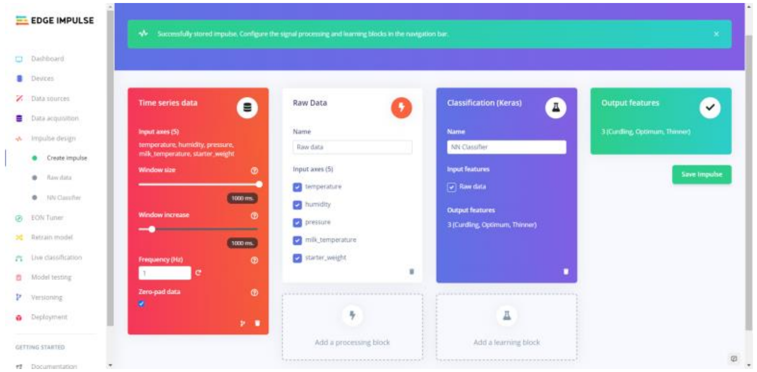
Curdling.sample_1.csv

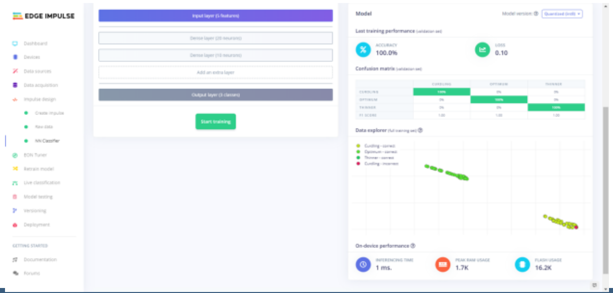
Curdling.sample_2.csv

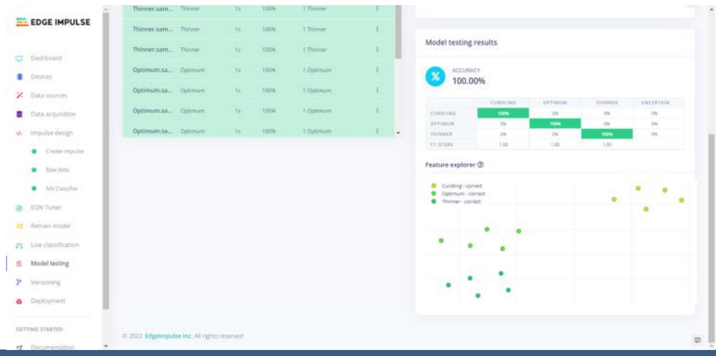


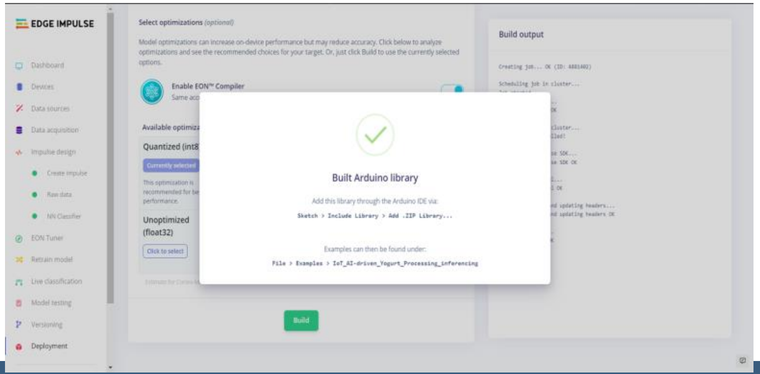






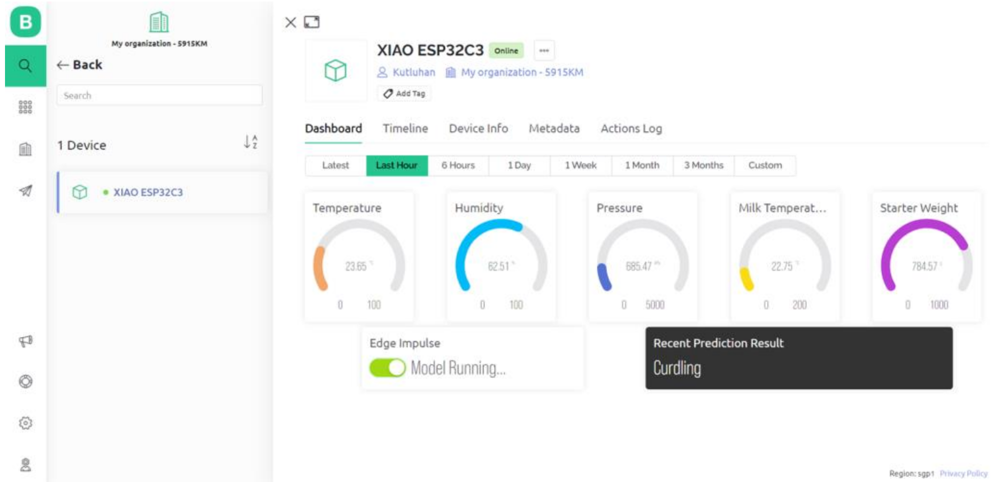




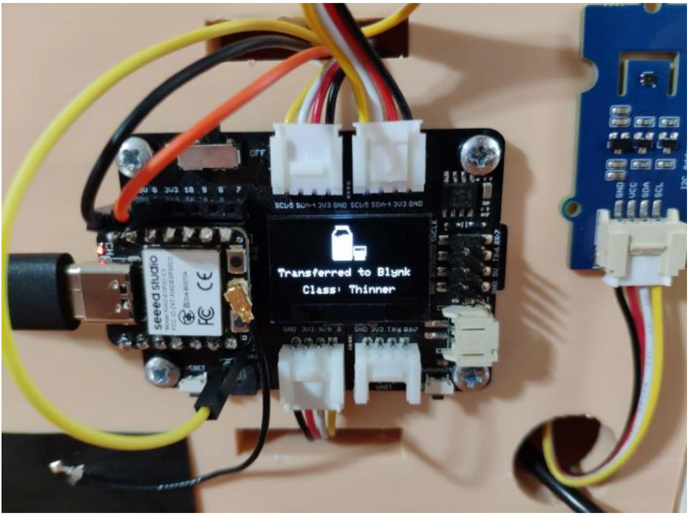


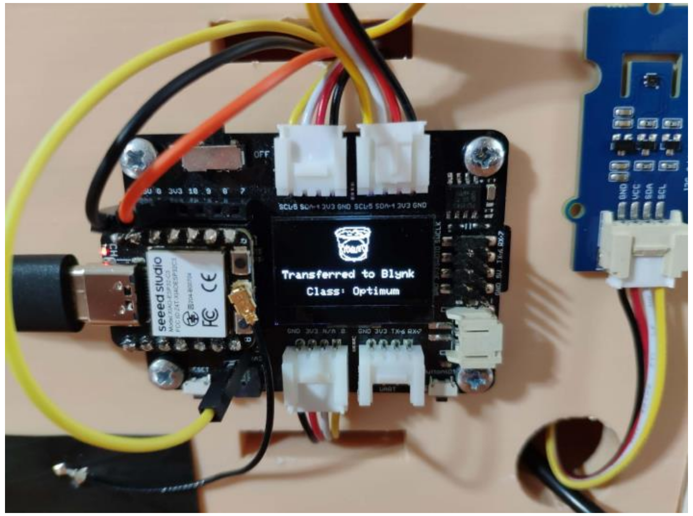
Informing the user of the model detection results via Blynk

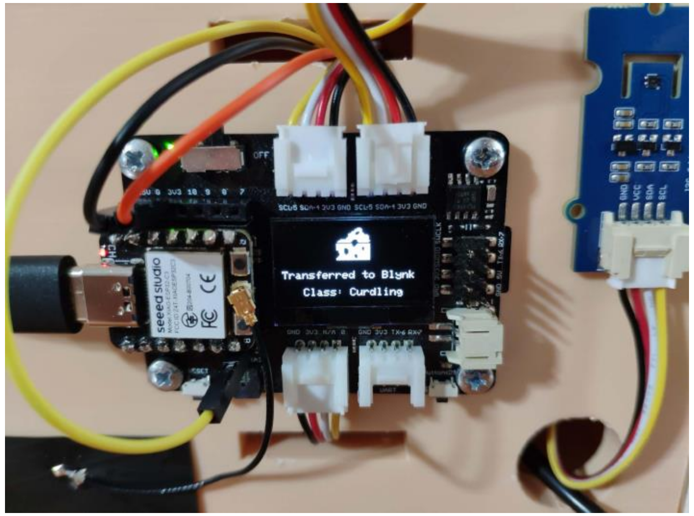
TINY ML TALKS webcast

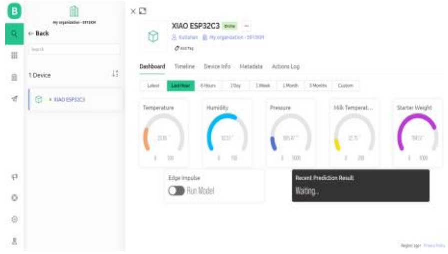


- Created a Blynk application and user interface
- Employed XIAO ESP32C3 to send the collected data to the Blynk application
- Utilized the switch (button) widget on the Blynk dashboard to run an inference with XIAO ESP32C3
- Then, displayed the model detection results on the Blynk dashboard









Further Discussions

By applying object detection models trained on captured infected tree images in detecting potential tree diseases, we can achieve to:			
avoid crop yield loss, animal deaths, widespread infectious epidemics	prevent land degradation due to soil erosion	mitigate deforestation	protect wildlife

[Project Tutorial]

AI-assisted Pipeline Diagnostics and Inspection w/ mmWave

Issue

Pipeline system maintenance has been crucial to keeping machine operations sustainable, profitable, and stable even though all machine parts and control units evolved from occupying rooms to fitting in our packets.

A faulty pipeline system can engender various manufacturing problems while running machine operations, especially for small businesses with limited budgets not enough to cover expensive overhauling costs.

During machine operations, mechanical and thermal stress cause minute defects in pipelines due to fatigue.

When small defects accumulate, the outcome mostly results in a varying turbulent pressure, which leads to slight form disfigurations, resulting in gradual deficiency over time due to tension.

Although there are different external pipeline inspection methods, they cannot be applied interchangeably to different pipeline systems.

Solution

Establish an efficient and accurate pipeline diagnostics mechanism conforming to general maintenance regulations.

Assist technicians in keeping machines durable and prevent companies from squandering their resources on replacing machine components due to pipeline defects.

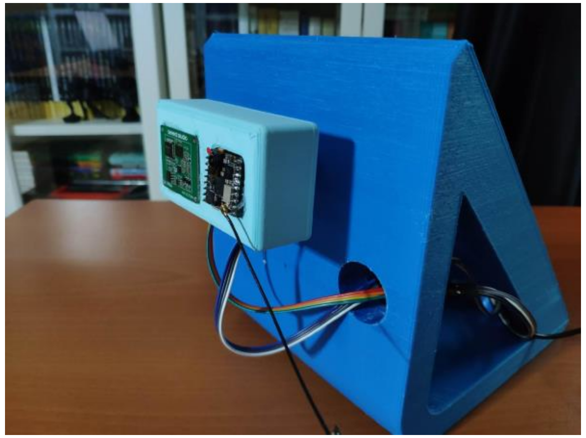
Since pipeline system failures can be detected by inspecting fluctuating vibrations as a non-destructive method, collect accurate vibration measurements from a pipeline system to train a neural network model with notable validity.

Inform the user of the detection results with images of the deformed pipes after running the neural network model.

Considering harsh operating conditions, design a unique PCB and a 3D printable case to make the device as sturdy and compact as possible.

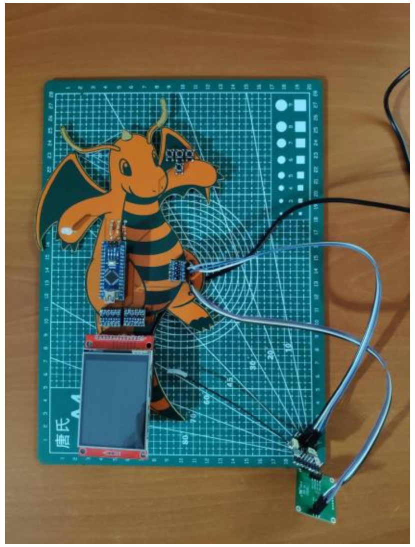
TINY ML TALKS webcast

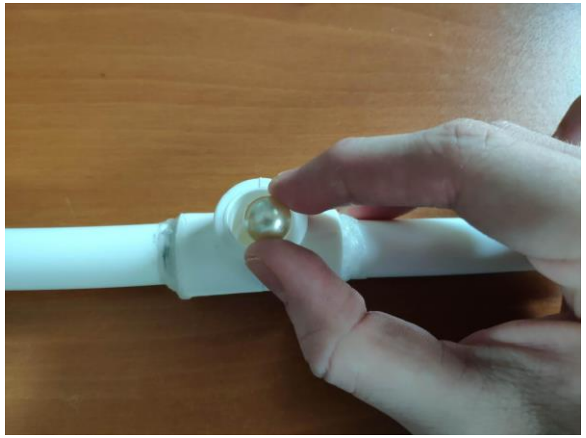
Collecting mmWave data parameters w/ Arduino Nano

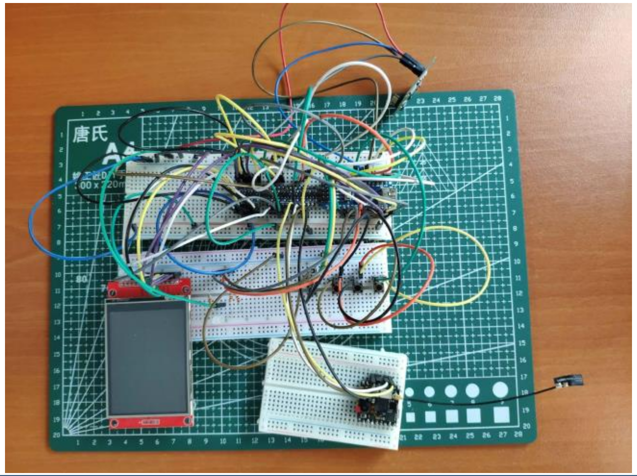


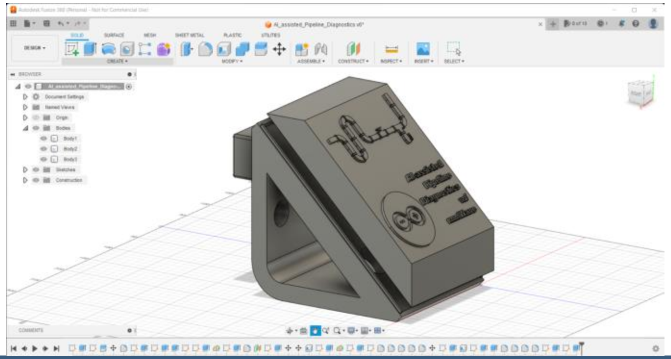
Blue Section	➞ Cracked
Red Section	➞ Clogged
Green Section	➞ Leakage

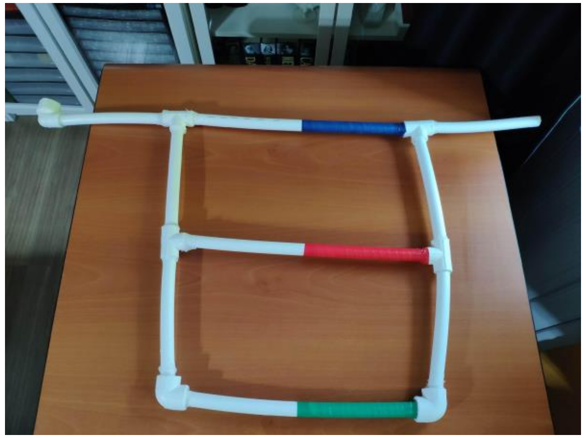
- Soldered the Dragonite-inspired PCB
- Assembled the liquid-themed case
- Developed a full-fledged web application in PHP to provide an outstanding user interface
- Hosted the web application on LattePanda 3 Delta
- Built a basic pipeline system demonstrating different defects

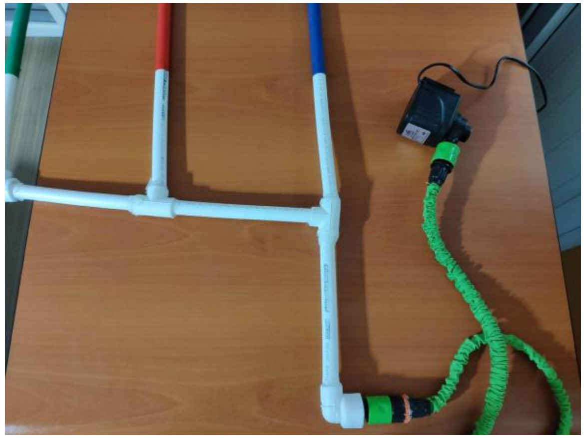








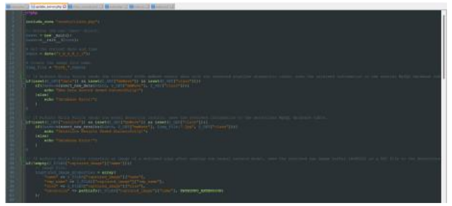




TINY ML TALKS webcast

Constructing a data set with notable validity

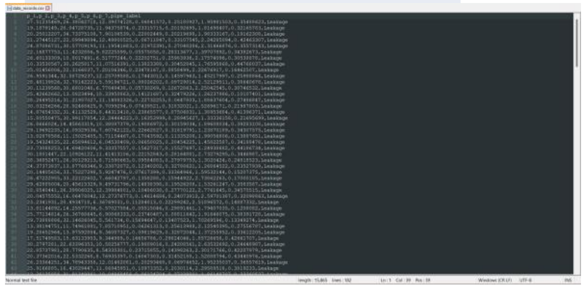
- Programmed Arduino Nano to extract data parameters from a 60GHz mmWave radar module and transmit the collected data parameters to Nicla Vision via serial communication
- Utilized the control buttons connected to Arduino Nano to select a class
- Employed Nicla Vision to transfer the received mmWave data parameters and selected classes to the web application via HTTP GET requests
- Used the web application to store the received information in a particular MySQL database table, display the saved data records, and let the user generate a CSV file, including all stored data records as samples

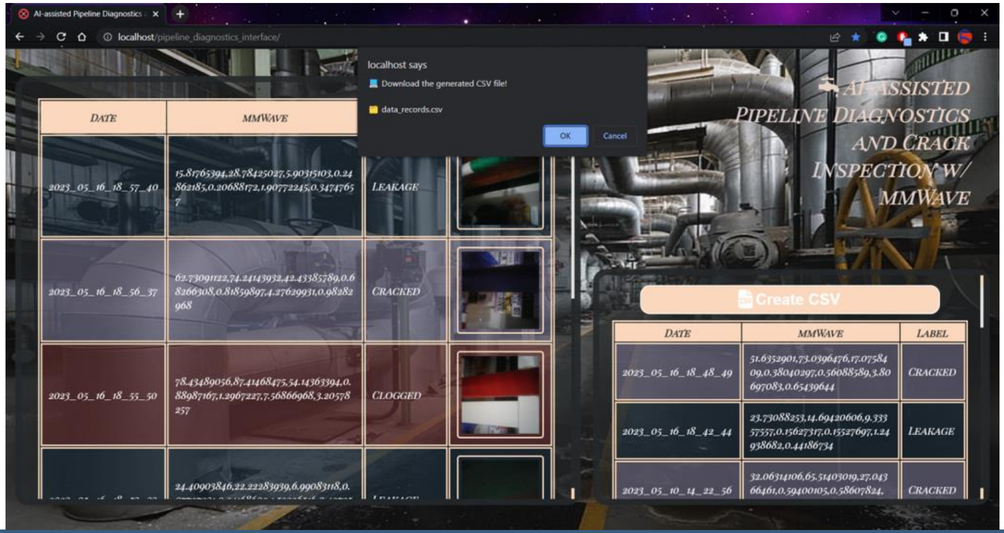


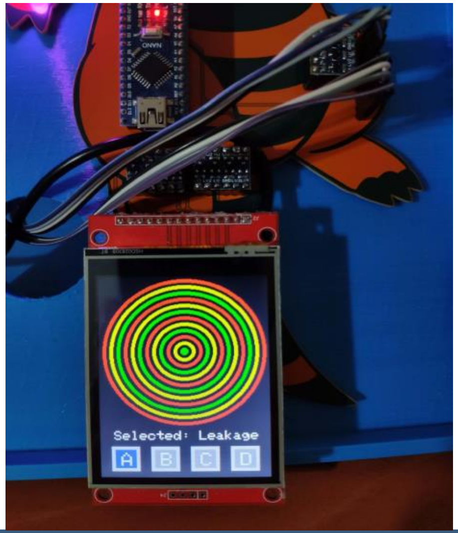
Leakage

Cracked

Clogged







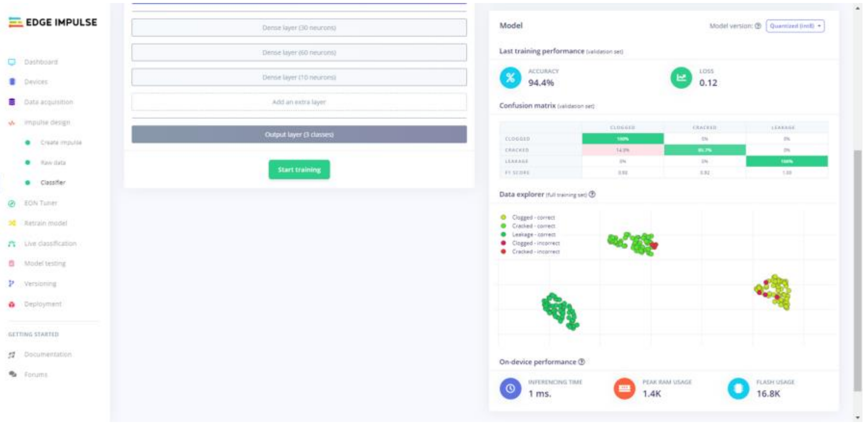
Building a neural network model with Edge Impulse

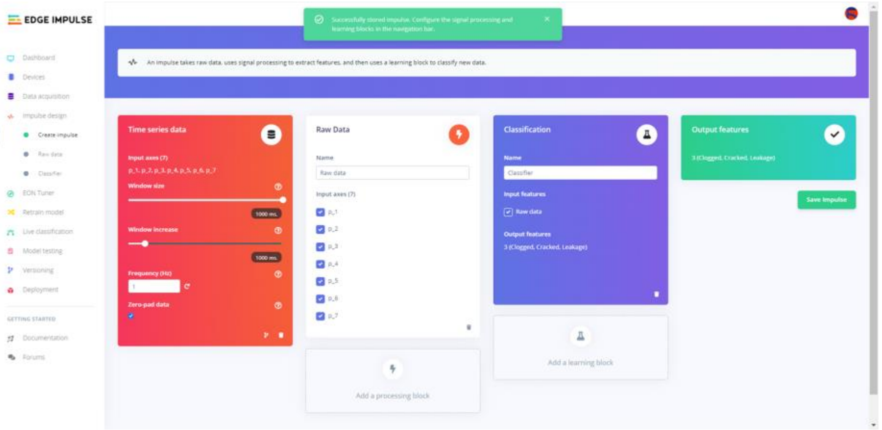
- Utilized the web application to generate a CSV file from all data records stored in the MySQL database table
- Uploaded the generated CSV file via the CSV Wizard
- Trained the model on various pipeline defects
- Evaluated the trained model accuracy
- Deployed the trained model as an Arduino library
- Set up the deployed model on Nicla Vision

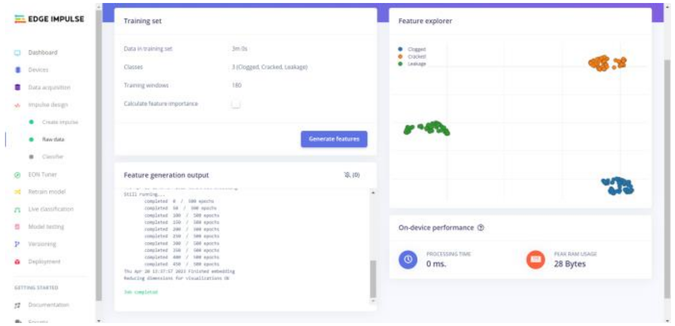
Clogged

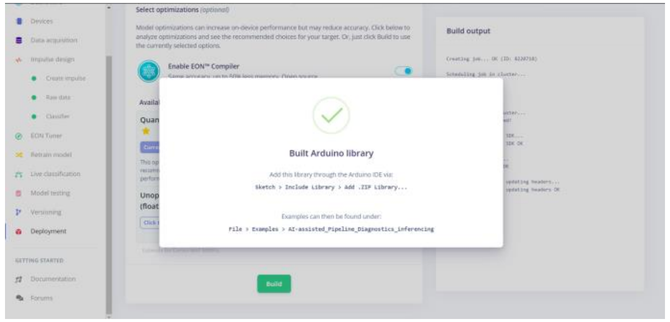
Cracked

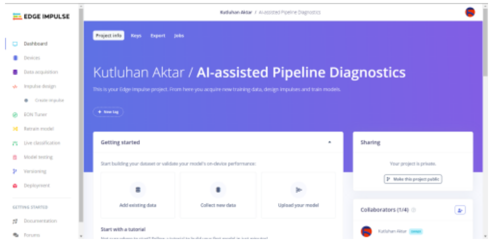
Leakage

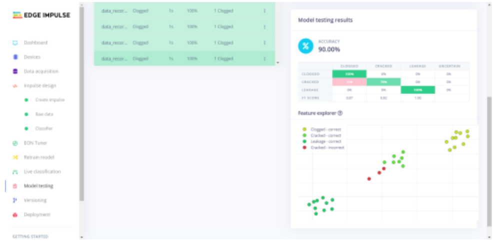


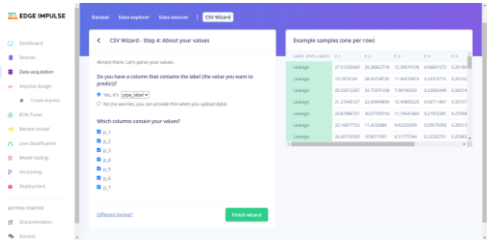






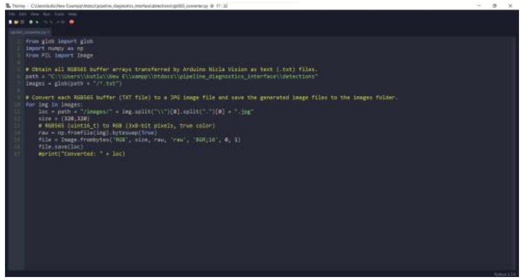


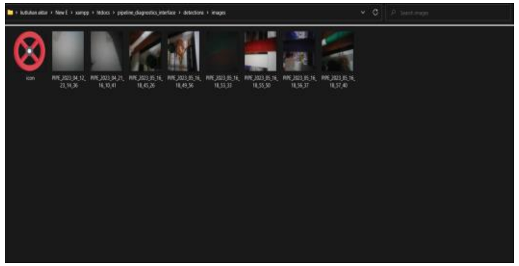


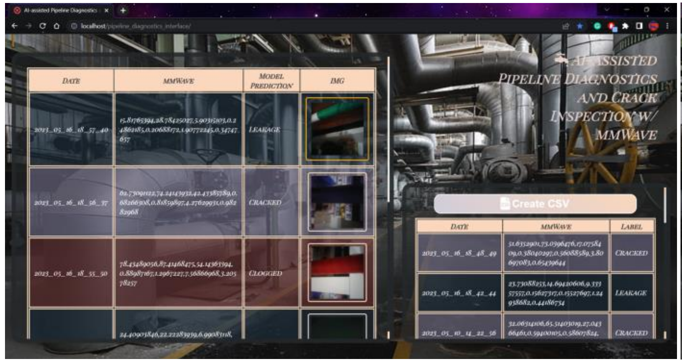


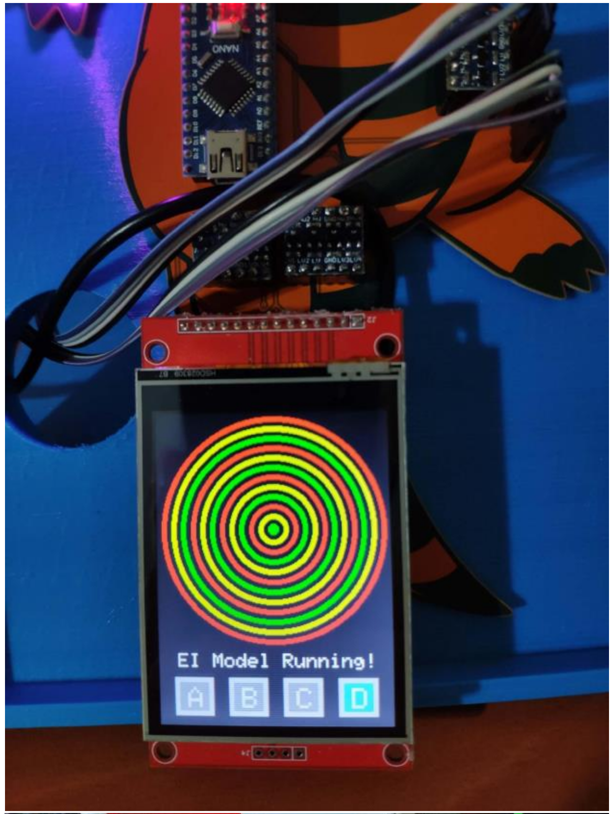
Informing the user of the detection results and deformed pipe images via the web application

- Programmed Nicla Vision to run an inference and capture a picture of the deformed pipe for further examination
- Also, employed Nicla Vision to transfer the received data parameters, the detection results, and the captured image of the deformed pipe (raw image buffer) to the web application via an HTTP POST request
- Then, utilized the web application to convert the received raw image buffer to a JPG file, save all obtained information to a particular MySQL database table, and show stored information and images on the application interface









Further Discussions

By applying neural network models trained on pipeline diagnostic classes in detecting pipeline system defects, we can achieve to:

keep machine operations sustainable, profitable, and stable	prevent faulty pipeline systems from engendering expensive manufacturing problems	assist small businesses with limited budgets in establishing an efficient and accurate pipeline diagnostics mechanism	reduce repair costs of high-value machine components	provide a non-destructive testing and evaluation (NDT&E) mechanism based on vibration characteristics

Project Tutorial

AI-driven LoRaWAN Fertilizer Pollution Detector w/ WhatsApp

Issue

Due to the evergrowing human population and declining fertile lands, farmers started to utilize organic fertilizers in conjunction with chemical fertilizers to improve crop yield, even to the extent of causing soil contamination.

Organic fertilizers behave differently depending on their manufacturing conditions and change the degree of soil permeability of different soil types, such as loamy, peaty, silty, chalky, etc.

Applying chemical fertilizers to the soil structurally altered by organic fertilizers without painstaking attention to soil test reports can be noxious.

Chemical fertilizers directly affect soil integrity, permeate through water bodies, and contaminate the groundwater and the environment.

When chemical fertilizers disperse throughout water bodies, they increase macronutrients in the environment, such as nitrogen, potassium, and phosphorus, resulting in contamination and eutrophication.

Solution

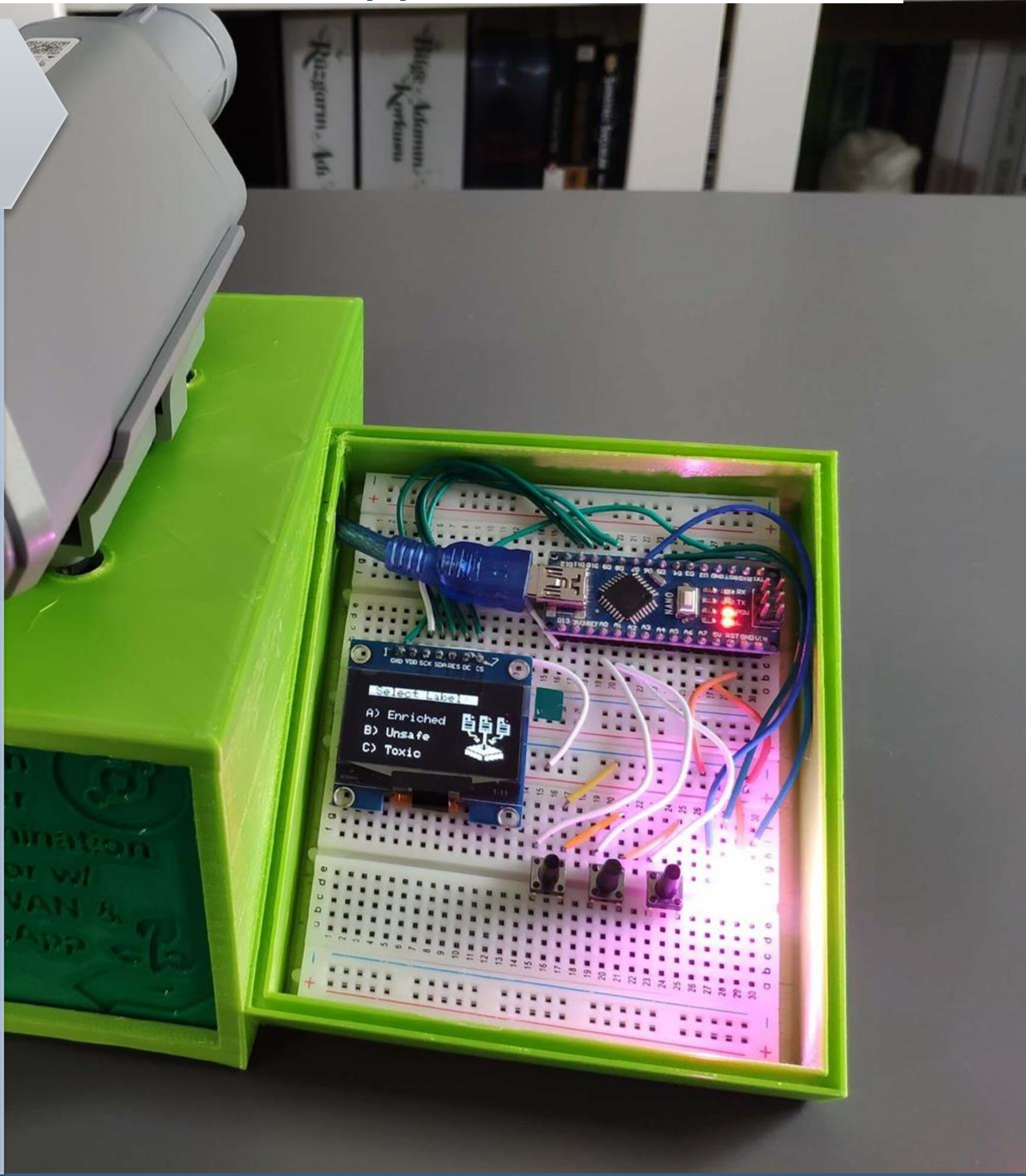
Detect the excessive use of chemical fertilizers in the presence of organic fertilizers and provide real-time detection results for further inspection.

Prewarn farmers to avert the detrimental effects of synthetic fertilizer overuse in relation to the applied organic fertilizers.

Collect data from a controlled environment manifesting different soil conditions to train an object detection model with notable validity.

Inform the user of the detection results after running the object detection model.

Design a 3D printable case to make the device as robust and sturdy as possible while operating outdoors.

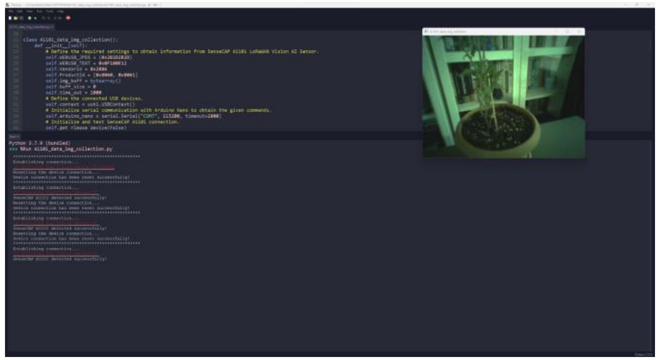


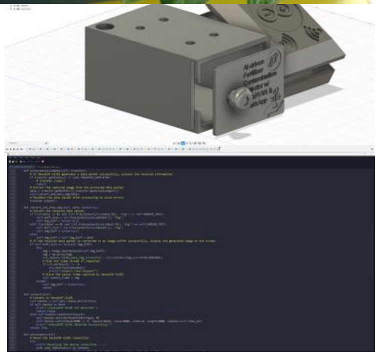
Capturing fertilizer-exerted soil images w/ SenseCAP A1101

Fresh (1 month)
Active (3 months)
Old (6 months)
Calcium Nitrate
Magnesium Sulphate
Ammonium Sulphate
Ammonium Phosphate


- Assembled the plant-themed case
- Produced organic fertilizers from quail manure in different decomposition stages
- Added chemical fertilizers and sowed tomato seedlings in three separate flowerpots
- Set up SenseCAP A1101 on LattePanda 3 Delta
- Programmed SenseCAP A1101 via LattePanda 3 Delta to obtain the commands transferred by Arduino Nano via serial communication and capture pictures of fertilizer-exerted soils



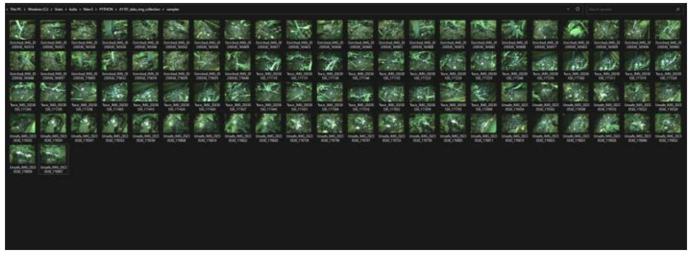



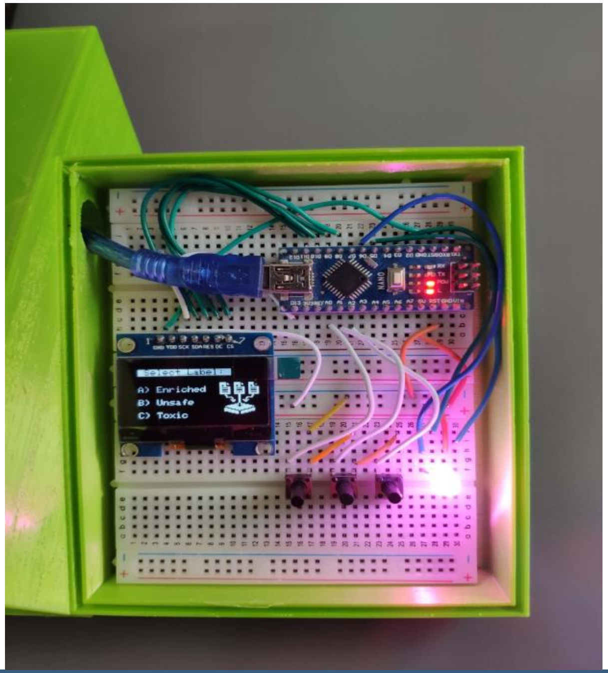



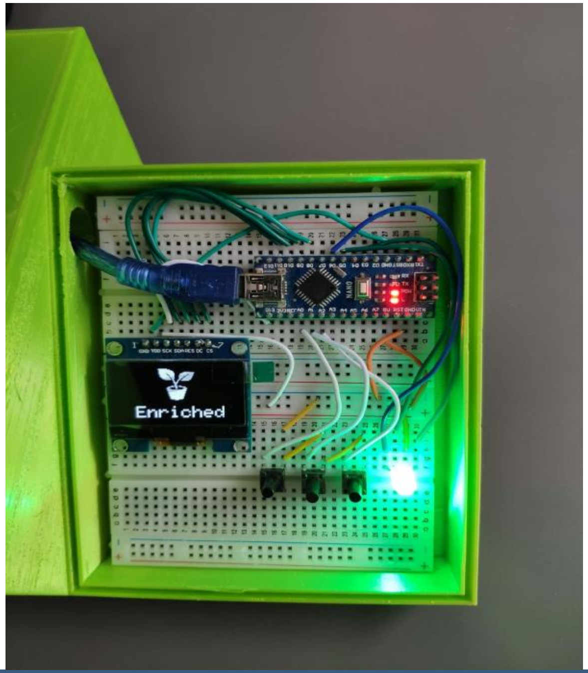
Constructing a data set with notable validity

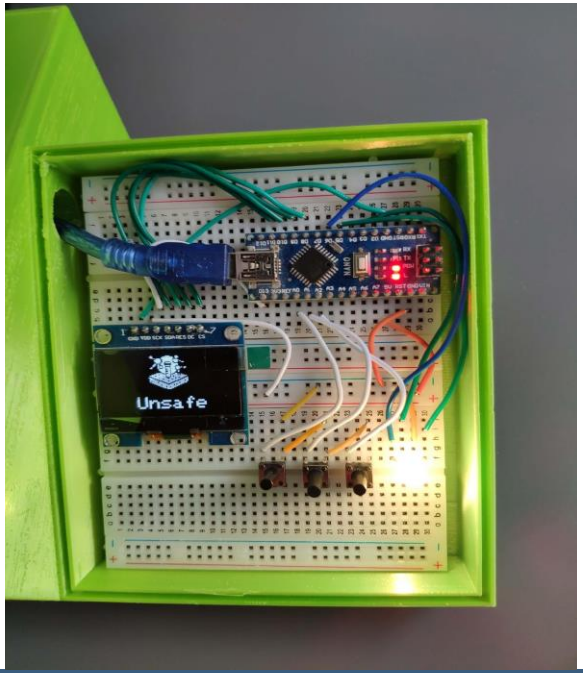
- Programmed Arduino Nano as a remote control to transfer commands to LattePanda 3 Delta via serial communication
- Utilized three control buttons connected to Arduino Nano to select classes
- Obtained my data set after capturing images of fertilizer-exerted soils in three separate flowerpots for nearly two months

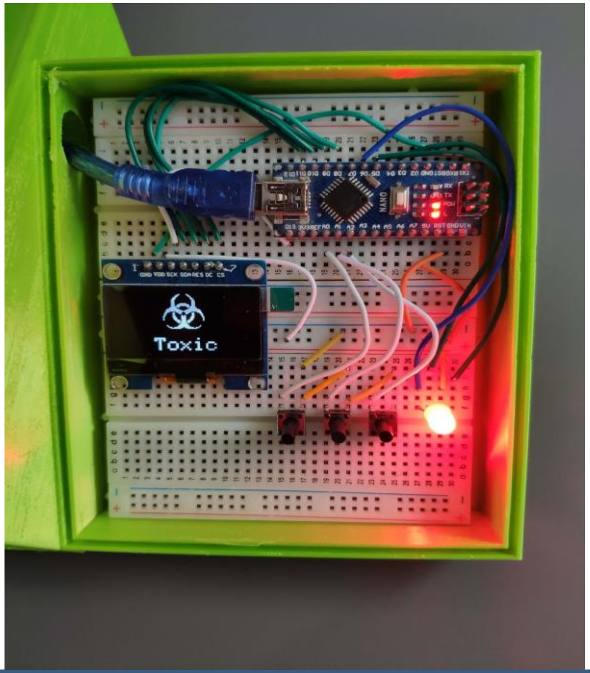
Enriched
Unsafe
Toxic







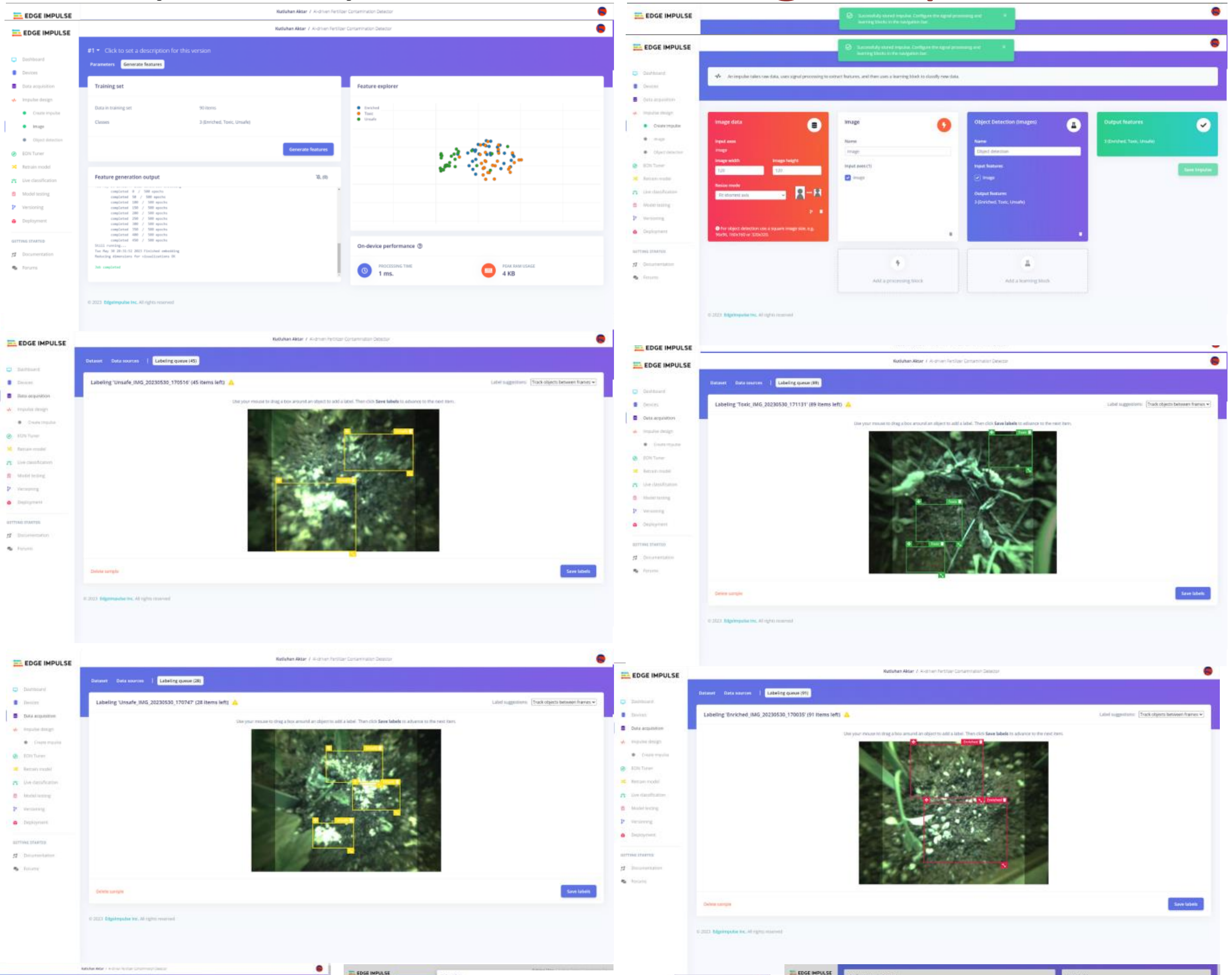


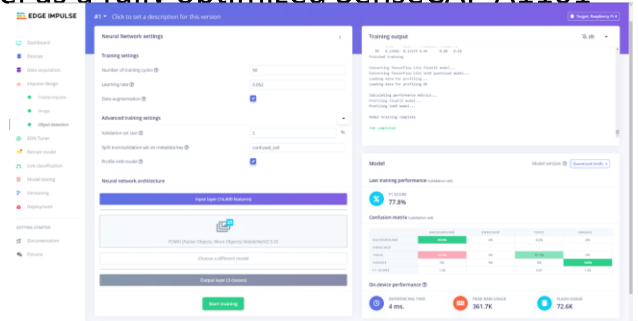


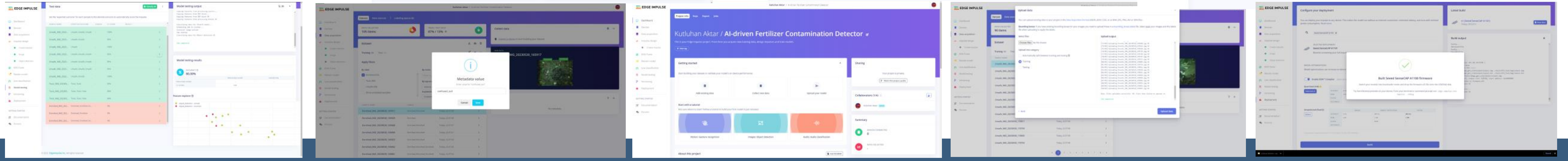
Building an object detection (FOMO) model with Edge Impulse

TINY ML TALKS webcast

- Uploaded the collected images (samples)
- Labeled targeted objects on the samples
- Trained the FOMO model on the fertilizer-exerted soil images
- Applied a metadata key to prevent leaking data between my train and validation sets due to some faulty samples
- Evaluated the model accuracy
- Deployed the trained model as a fully optimized SenseCAP A1101 firmware (UF2)



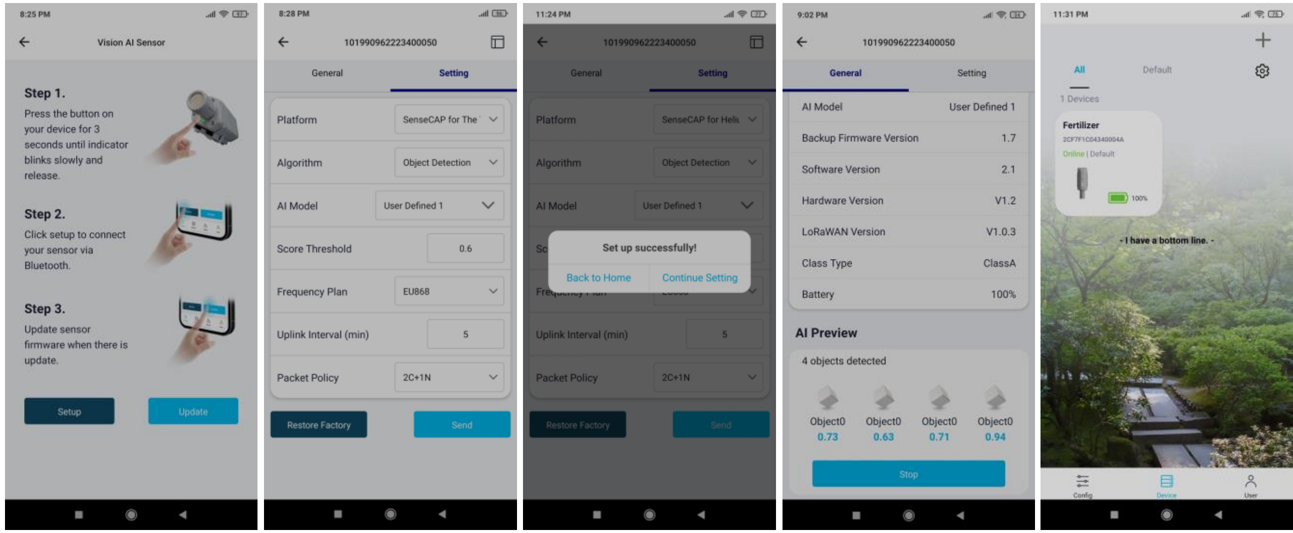


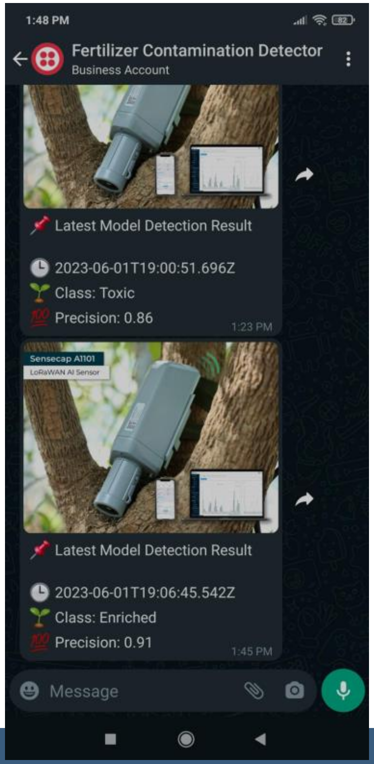


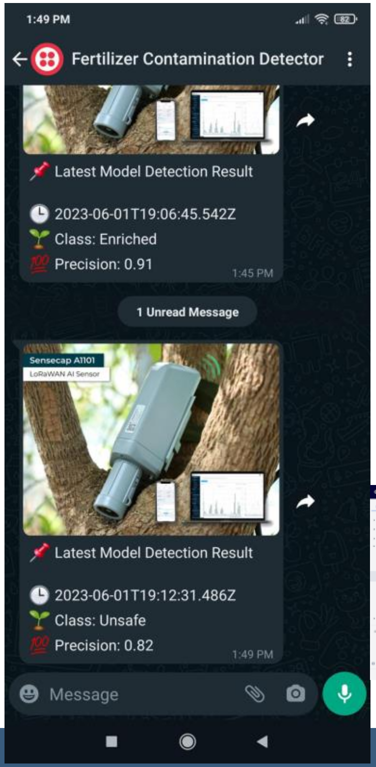
Informing the user of the detection results over WhatsApp via the LoRaWAN network

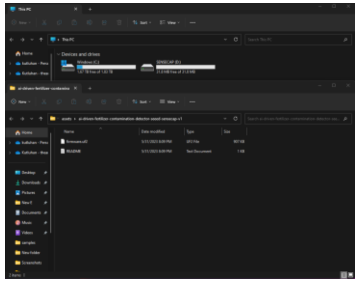
TINY ML TALKS webcast

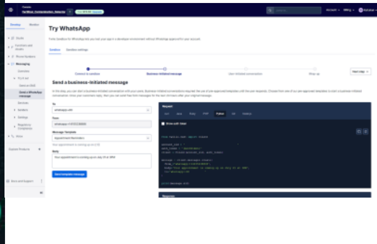
- Connected SenseCAP A1101 to SenseCAP Mate App
- Configured the Edge Impulse FOMO model on SenseCAP A1101
- Utilized SenseCAP A1101 to transfer the detection results to the SenseCAP Portal via the Helium LongFi Network
- Created a Twilio account to utilize Twilio's WhatsApp API
- Developed a Python application to obtain the model detection results from the SenseCAP Portal by making HTTP GET requests to the SenseCAP HTTP API
- Then, utilized the application to transfer the retrieved model detection results to the verified phone number

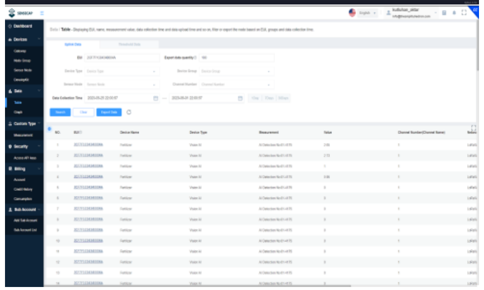












TINY ML TALKS webcast

Further Discussions

By applying object detection models trained on numerous fertilizer-exerted soil images in detecting the excessive use of chemical fertilizers, we can achieve to:			
prevent chemical fertilizers from contaminating the groundwater and the environment	avoid chemical fertilizers from dispersing throughout water bodies and increasing macronutrients in the environment	mitigate the risk of severe health issues due to nitrite-contaminated water, such as DNA damage, lipid peroxidation, and oxidative stress	protect wildlife from the execrable effects of excessive chemical fertilizer use

Project Tutorial

TINY ML TALKS webcast

For Further Questions

hackster.io/kutluhan-aktar

Kutluhan Aktar

@ThAmplituhedron

Thank you, **tinyML Strategic Partners**,
for committing to take tinyML to the next Level, together

TINY ML TALKS webcast

AIZIP
ANALOG DEVICES AHEAD OF WHAT'S POSSIBLE™
ARDUINO
arm AI
brainchip
EDGE IMPULSE
GREENWAVES TECHNOLOGIES
Grovety Inc.
IBM
imagimob
infineon
innatera
Microsoft
Nota AI
NXP
POLYN TECHNOLOGY
Qeexo
Qualcomm
RENESAS
Schneider Electric
SensiML™
SONY
SILICON LABS
ST life.augmented
synaptics
SYNTIANT

Copyright Notice

www.tinyml.org

Copyright Notice