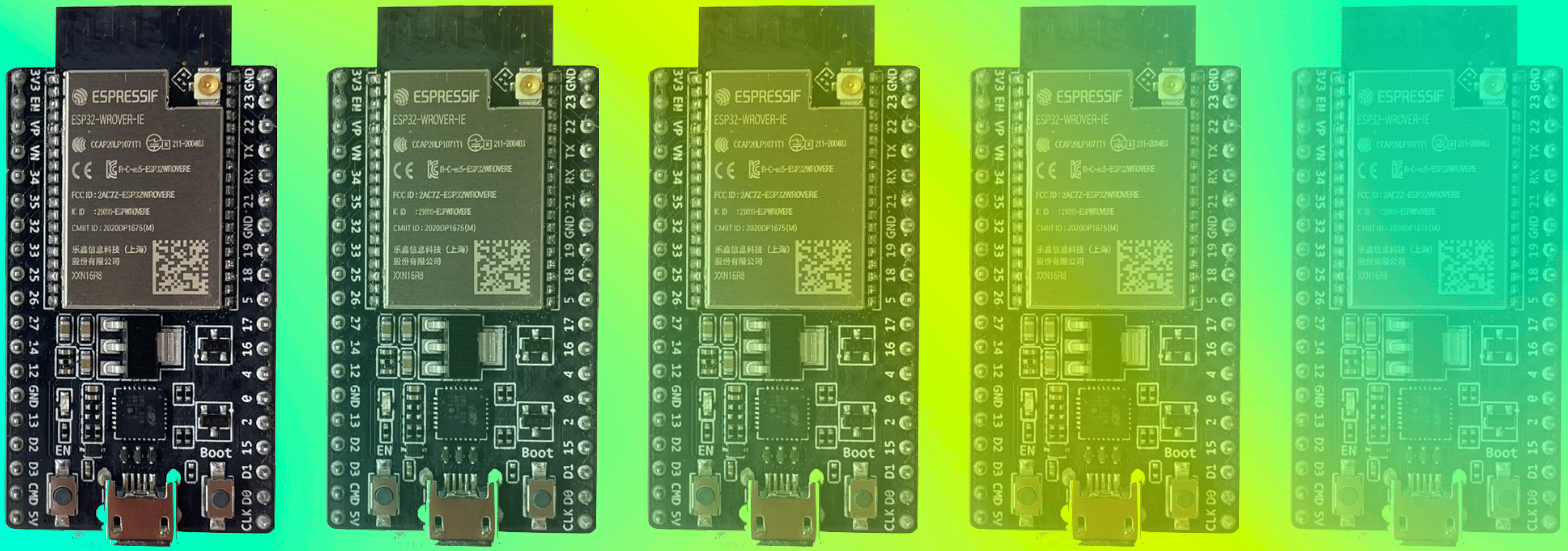
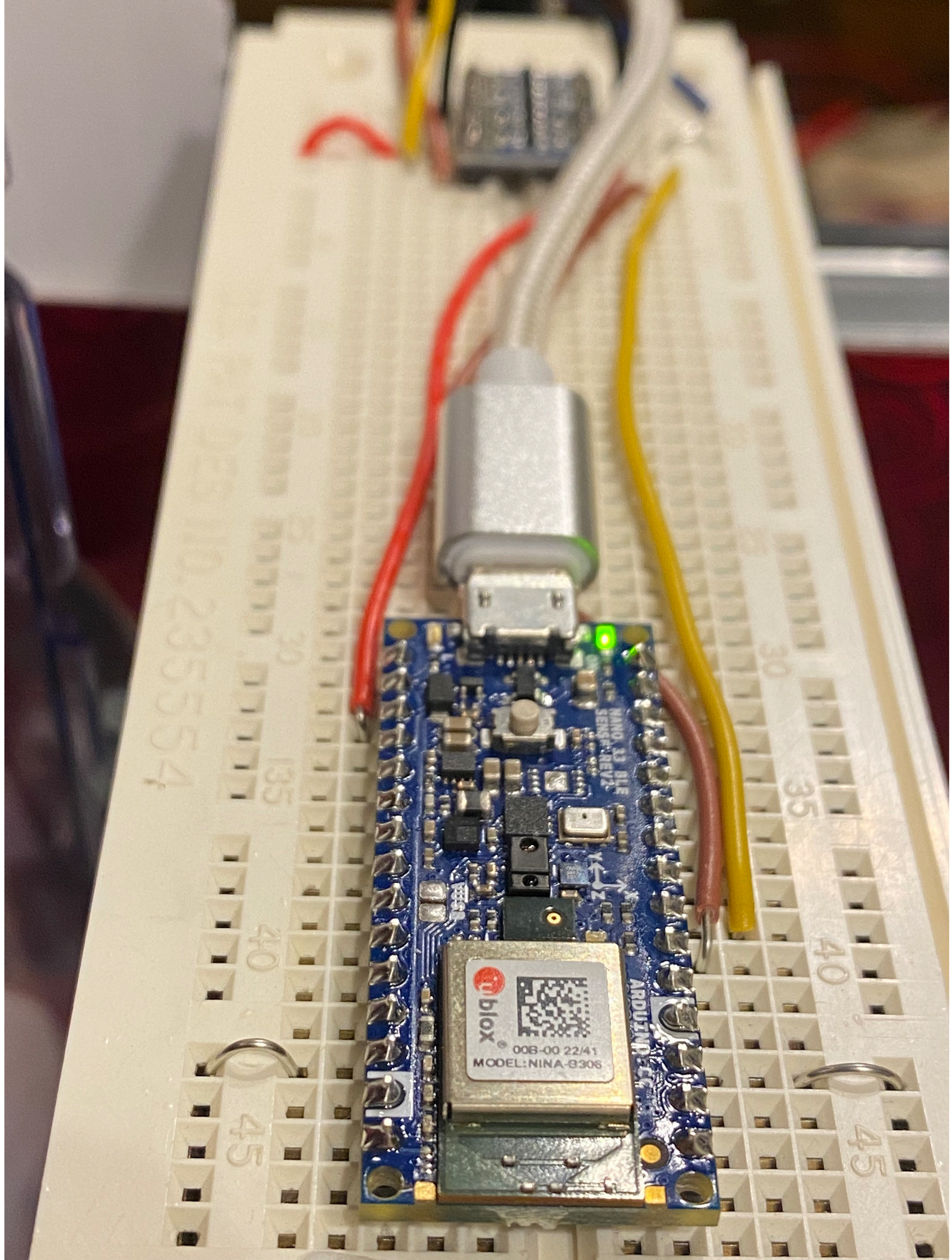
# TinyML WITH ESP32

MACHINE LEARNING WITH IOT FOR EVERYONE
BY MICHELLE KOETH - BS ECE, JD, OG COMPUTER NERD



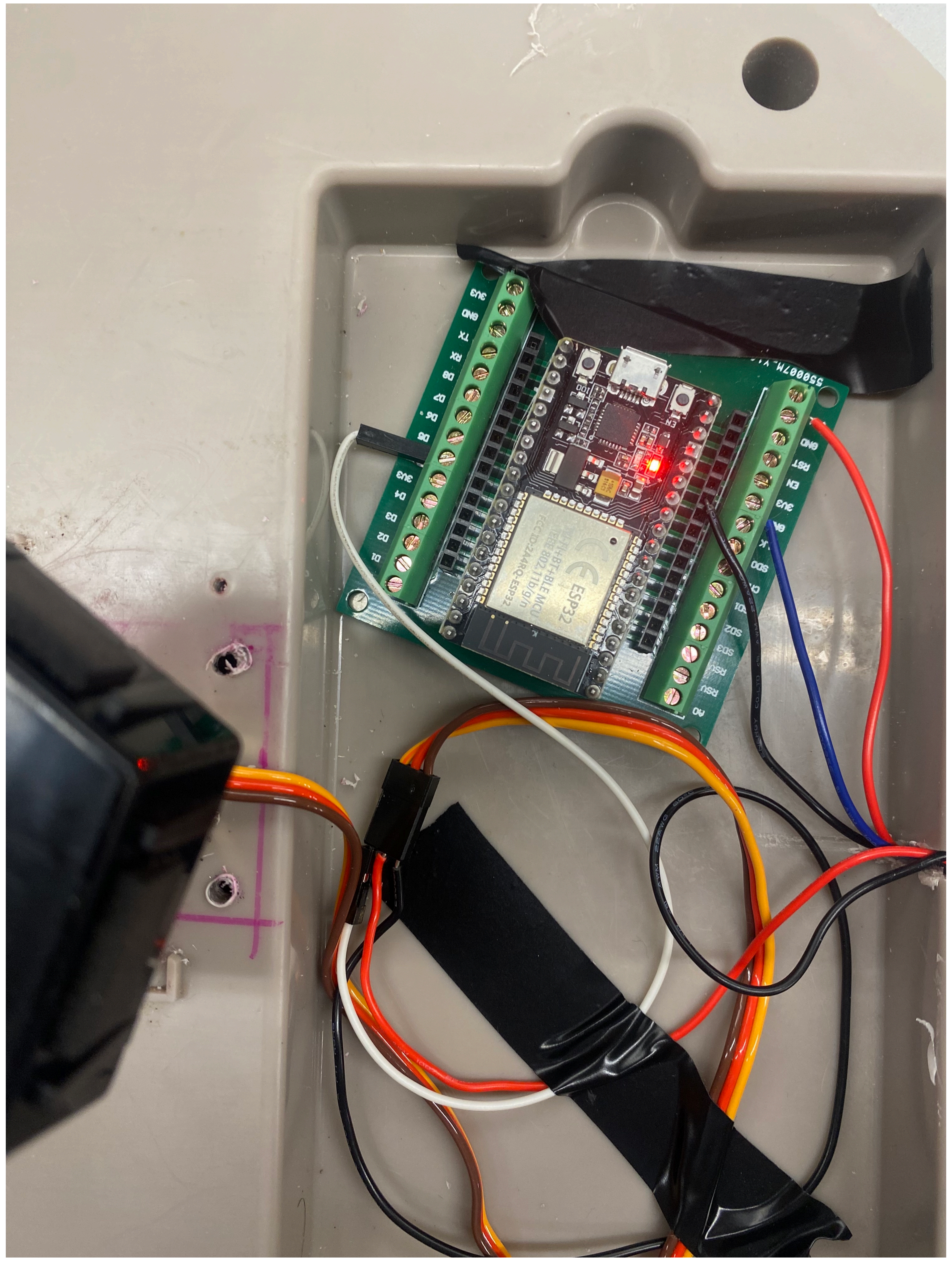
# TOPICS WE'LL DISCUSS:

- Edge Computing
- TinyML
- Microprocessors that run TinyML
- Bird species classification by TinyML
- A Neural Network used in image classifying
- Edge Impulse ML Model building interface
- Market considerations of Edge devices
- The future of TinyML

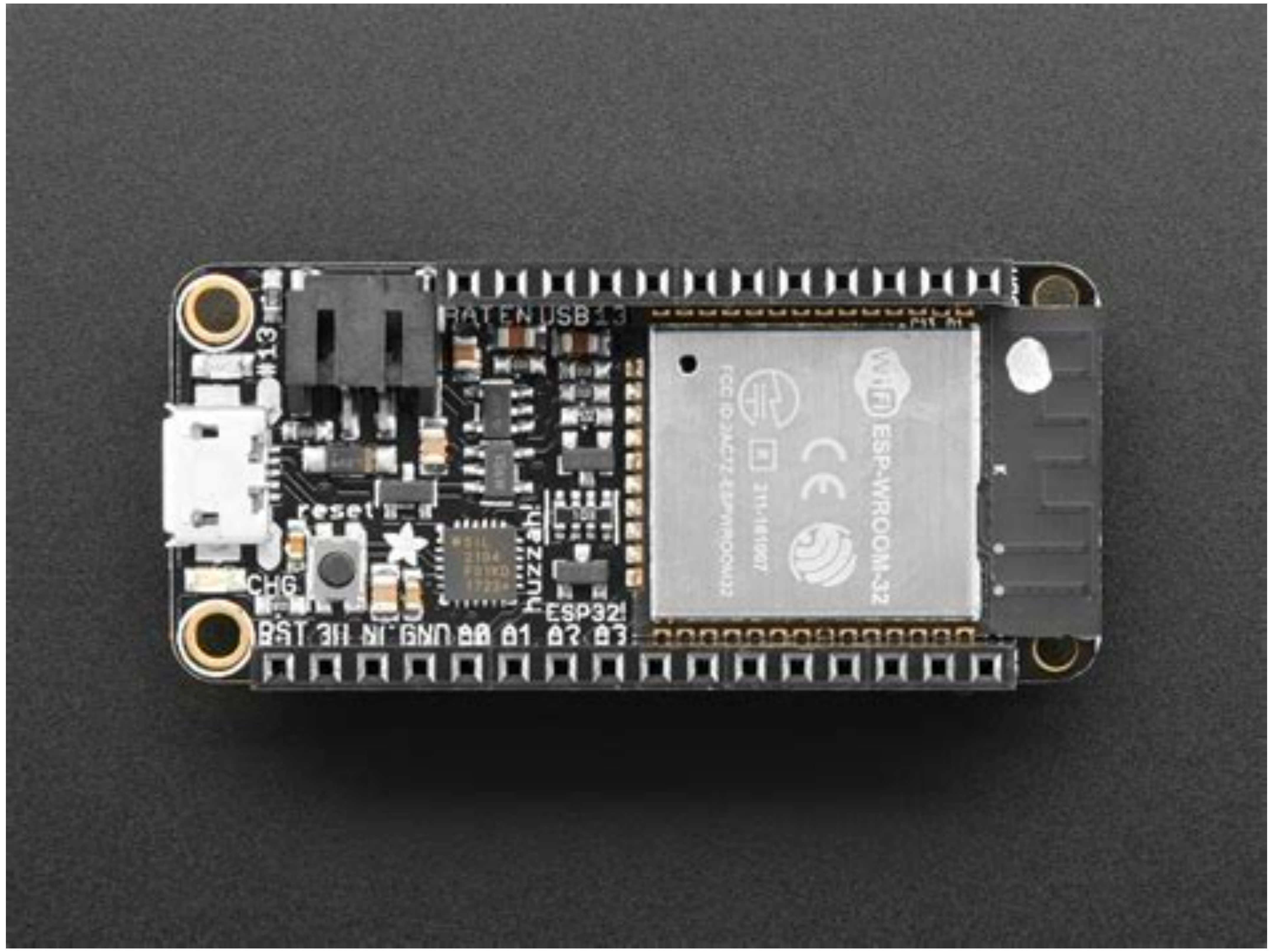


# “EDGE COMPUTING”

- Collect and process data at the source
- Greater and higher quality data collection
- Less  central processing
- Less  network traffic & thus less latency
- Data privacy / security
- Specialized hardware
- Redundancy and scalability



# ESP32 VS. RASPBERRY PI



Adafruit HUZZAH32 ESP32 Feather breakout board - $20

Raspberry Pi 4 Model B - $55

# WHAT IS TINYML?

Machine learning on "the edge"

IoT microprocessor based machine learning:

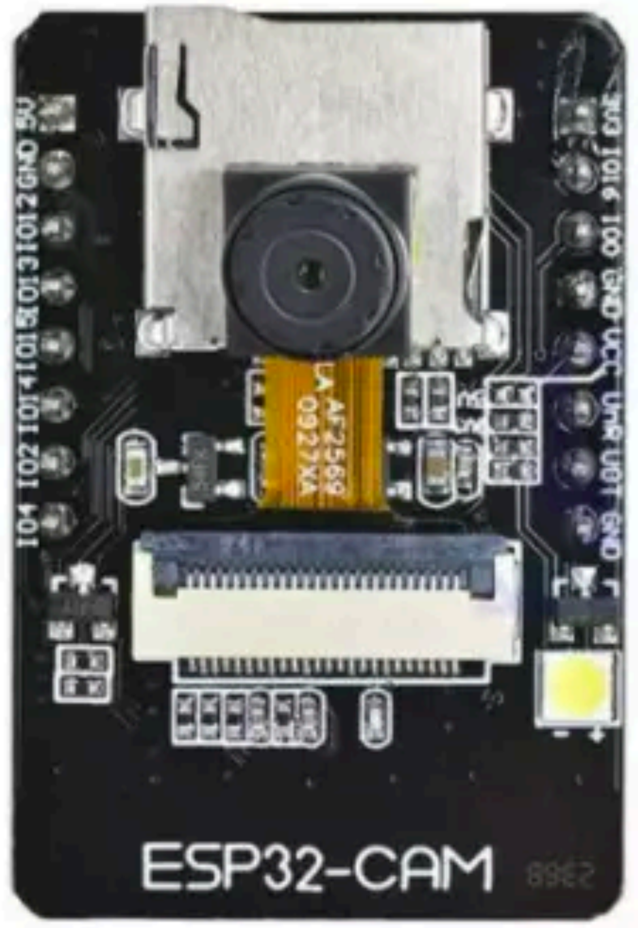
- Low power (mW range) Battery op

- Always-on use cases

"Inferencing" standalone or in a client-server configuration

data collection in "the field" (edge)

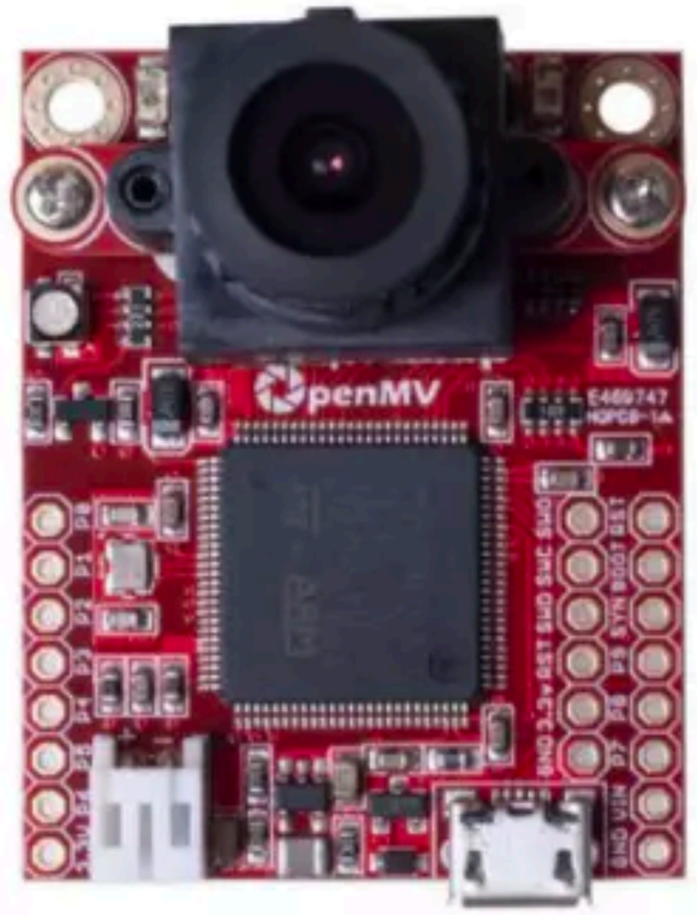
More realistic data = better models



**ESP32-CAM**
CPU: Dual core, 32-bit, 240 MHz, 4 MB Flash, 520 KB SRAM
Camera: 2 MP
Price: ≈ $ 8

**Sipeed MAIX Bit**
CPU: RISC-V Dual Core, 64-bit, 400 Mhz, 16 MB Flash, 8 MB SRAM
Camera: 2 MP
Price: ≈ $ 21

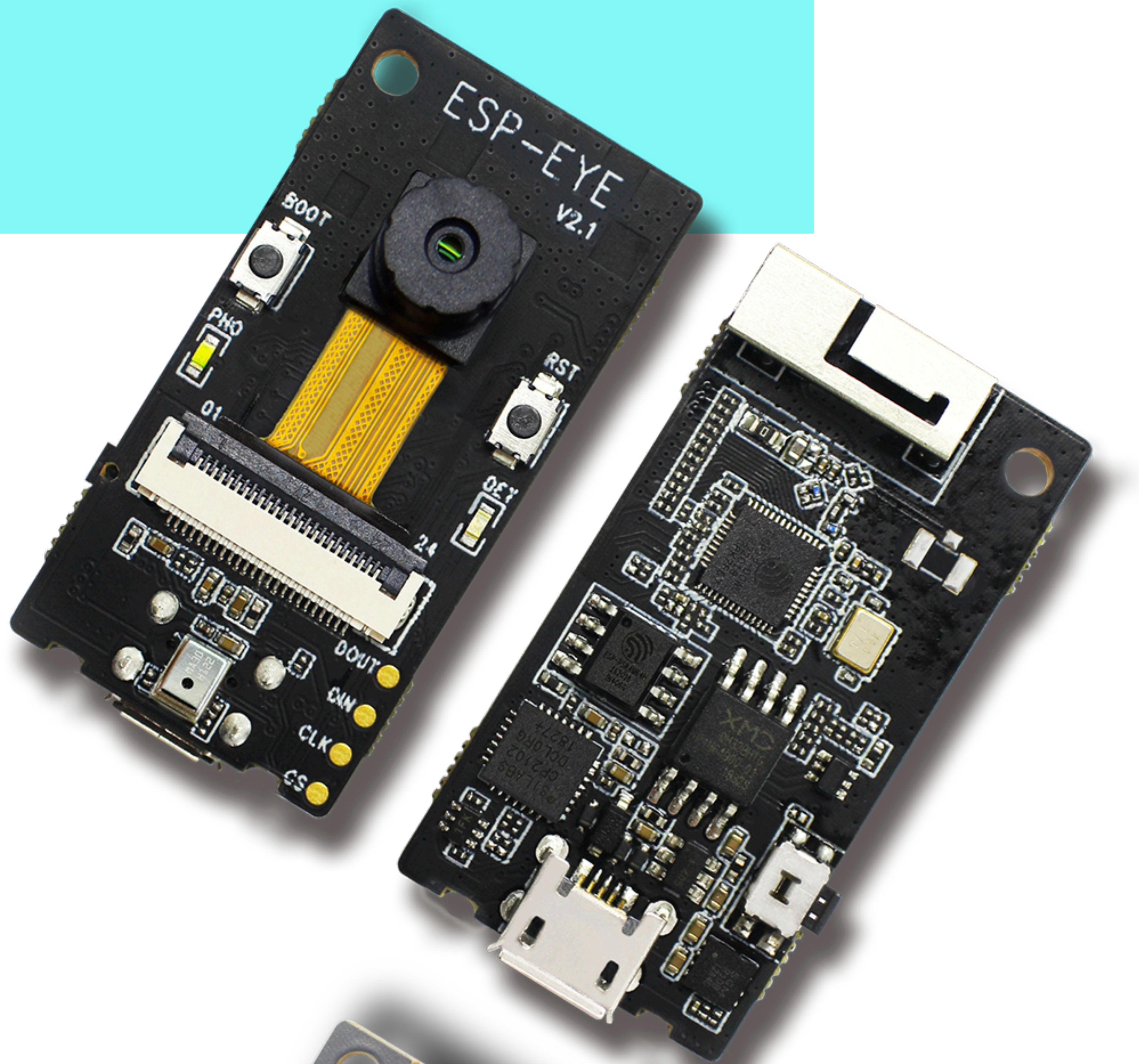
**M5StickV AI Camera**
CPU: RISC-V, 64-bit, 400 MHz, Dual FPU, 16 MB Flash, 8 MB SRAM
Camera: VGA
Price: ≈ $ 30



**OpenMV Cam H7**
CPU: ARM Cortex-M7, 32-bit, 480 MHz, 2 MB Flash, 1 MB SRAM
Camera: 5 MP
Price: ≈ $ 65

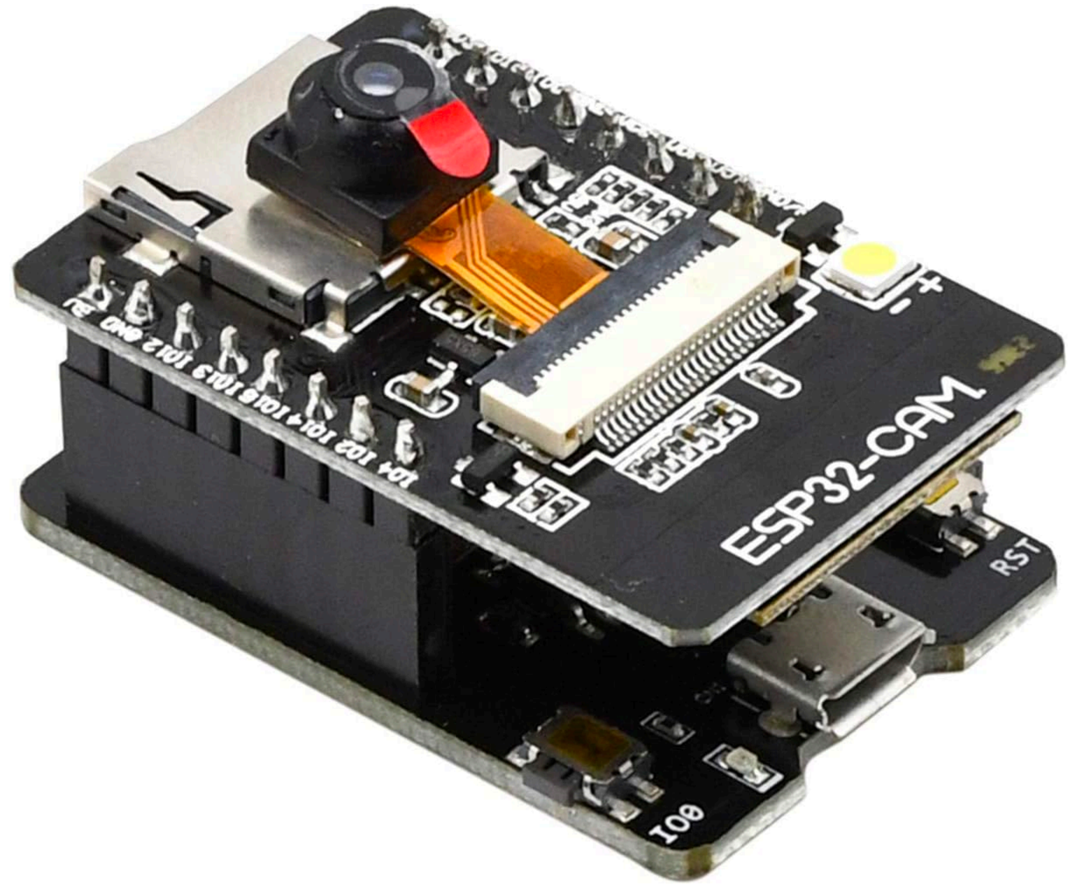
# THE HARDWARE:

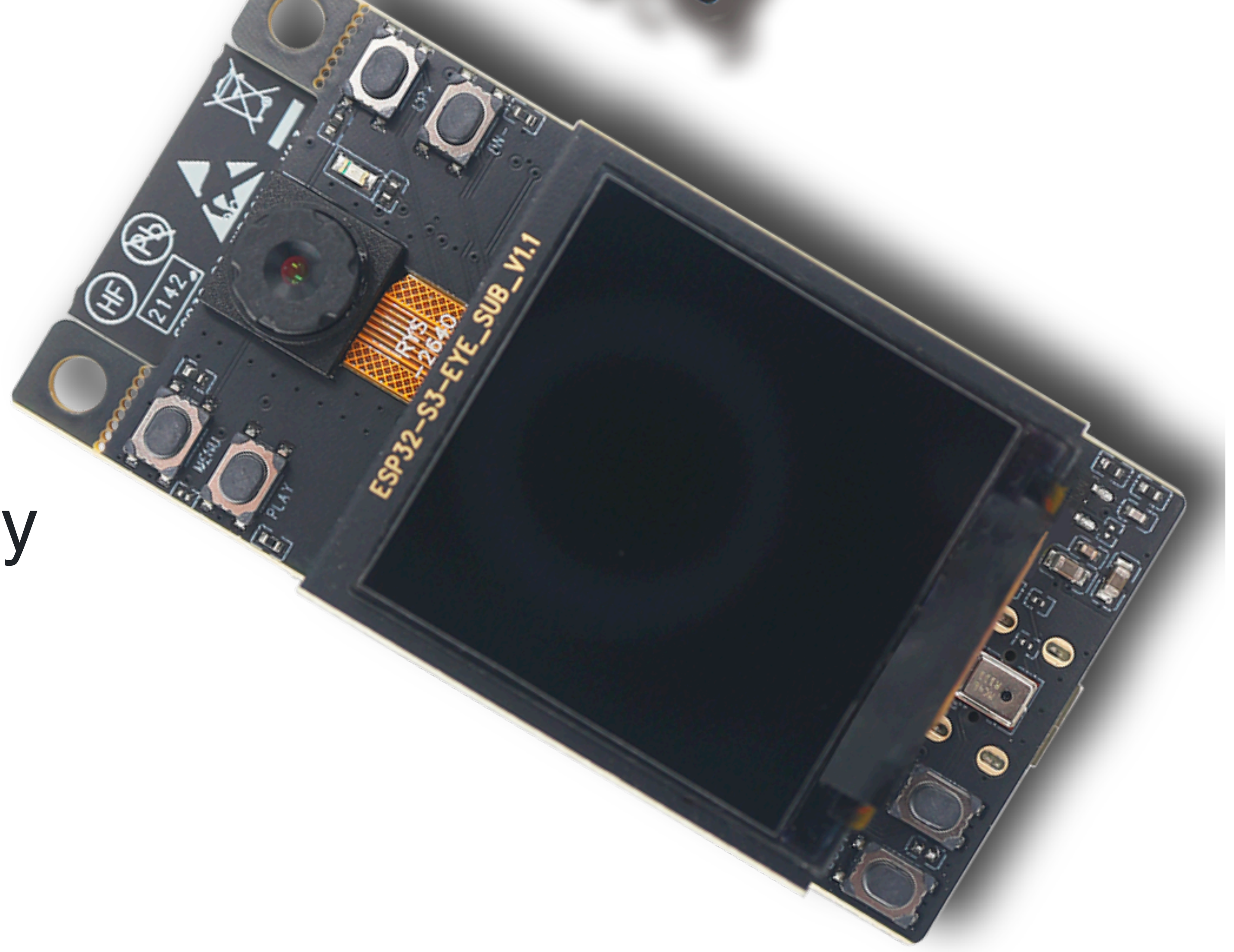
## “AI-IOT” SYSTEM ON A CHIP (SOC)



### ESP32-CAM (LATE 2019):

- Espressif ESP32 @ 160Mhz
- 2.4 GHz Wi-Fi and Bluetooth
- 520KB SRAM and 4MB PSRAM memory
- TF card (microSD) socket
- 2 Megapixel Camera
- Costs $8





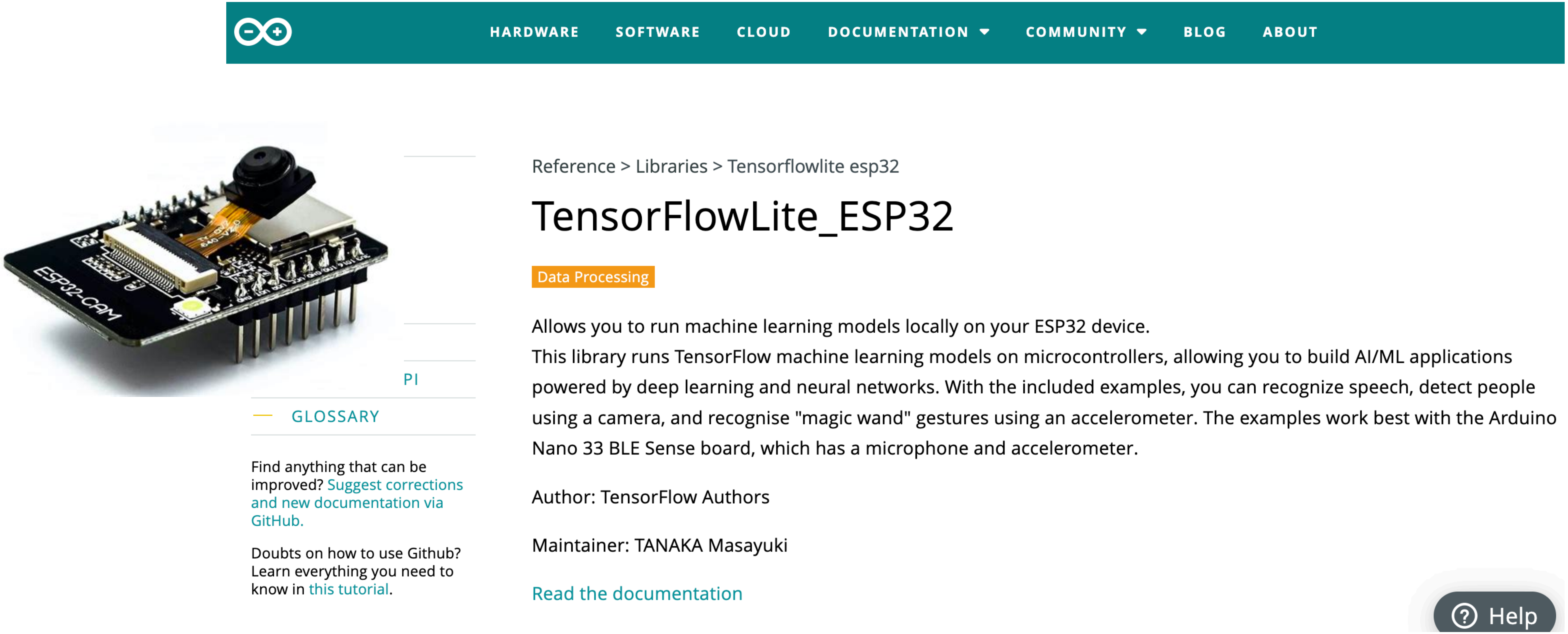
### ESP-EYE (2020):

- Espressif ESP32 @ 240Mhz
- 2.4 GHz Wi-Fi and Bluetooth
- 4MB Flash and 8MB PSRAM memory
- 2 Megapixel Camera
- MEMS microphone
- Costs $20

### ESP32-S3-EYE (2021):

- Espressif ESP32-S3 @ 240Mhz
- 2.4 GHz Wi-Fi and Bluetooth
- 8MB Flash and 8MB PSRAM memory
- 2 Megapixel Camera
- MEMS microphone
- LCD display
- Accelerometer
- Costs $48

# ESP32-CAM: GATEWAY TO TINYML



Reference > Libraries > Tensorflowlite esp32

## TensorFlowLite_ESP32

Data Processing

Allows you to run machine learning models locally on your ESP32 device.
This library runs TensorFlow machine learning models on microcontrollers, allowing you to build AI/ML applications powered by deep learning and neural networks. With the included examples, you can recognize speech, detect people using a camera, and recognise "magic wand" gestures using an accelerometer. The examples work best with the Arduino Nano 33 BLE Sense board, which has a microphone and accelerometer.

Author: TensorFlow Authors

Maintainer: TANAKA Masayuki

Read the documentation

Find anything that can be improved? Suggest corrections and new documentation via GitHub.

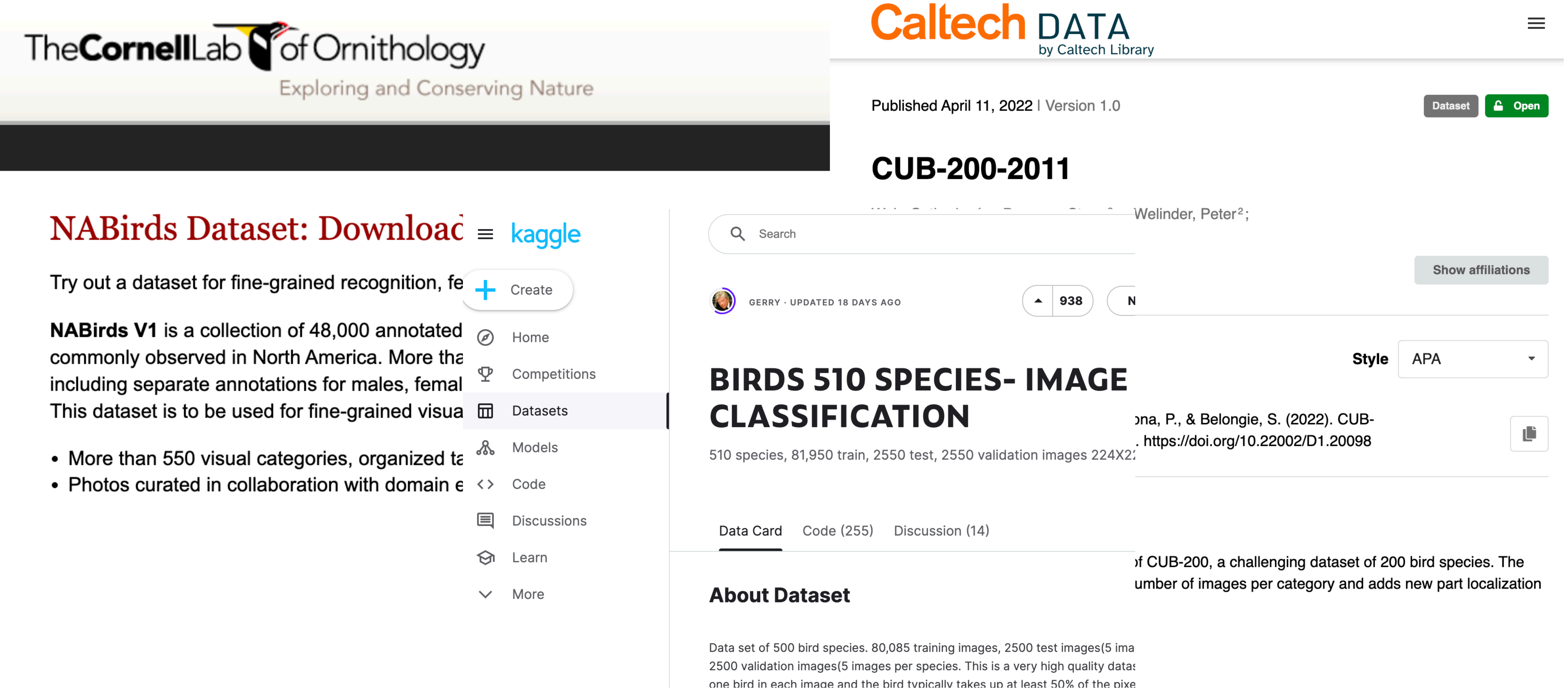
Doubts on how to use Github? Learn everything you need to know in this tutorial.

# THE CHALLENGE:

**RECOGNIZE "BULLY BIRDS" FROM A BIRD FEEDER USING AIoT & TINYML**

# THE DATA:

## TINYML IS “FOR THE BIRDS” OR, IMAGE RECOGNIZING THEM ANYWAY...



TheCornellLab of Ornithology

Exploring and Conserving Nature

Caltech DATA by Caltech Library

Published April 11, 2022 | Version 1.0

Dataset Open

CUB-200-2011

Welinder, Peter$^2$;

Show affiliations

Style APA

ona, P., & Belongie, S. (2022). CUB-

. https://doi.org/10.22002/D1.20098

of CUB-200, a challenging dataset of 200 bird species. The

umber of images per category and adds new part localization

NABirds Dataset: Download

Try out a dataset for fine-grained recognition, fe

**NABirds V1** is a collection of 48,000 annotated commonly observed in North America. More tha including separate annotations for males, femal This dataset is to be used for fine-grained visua

- More than 550 visual categories, organized ta
- Photos curated in collaboration with domain e

BIRDS 510 SPECIES- IMAGE CLASSIFICATION

510 species, 81,950 train, 2550 test, 2550 validation images 224X2

Data Card Code (255) Discussion (14)

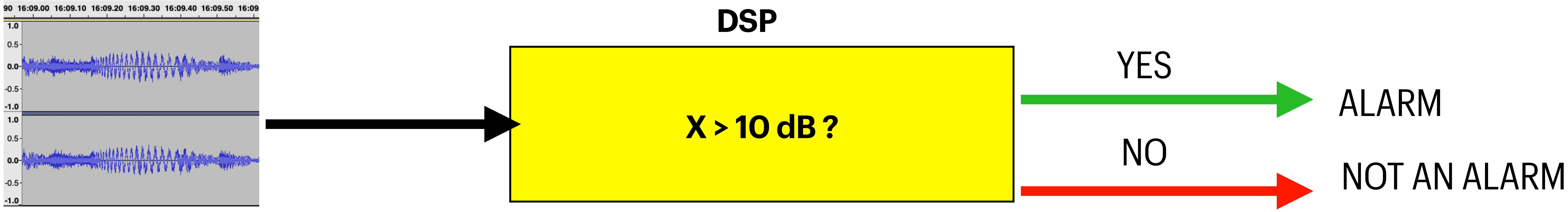
About Dataset

Data set of 500 bird species. 80,085 training images, 2500 test images(5 ima 2500 validation images(5 images per species. This is a very high quality datas one bird in each image and the bird typically takes up at least 50% of the pixe

“THE FIELD OF STUDY THAT GIVES COMPUTERS THE ABILITY TO LEARN WITHOUT EXPLICITLY BEING PROGRAMMED.”

ARTHUR SAMUEL

# NEURAL NETWORKS AS NEXT-GEN DIGITAL SIGNAL PROCESSING



DSP

X > 10 dB ?

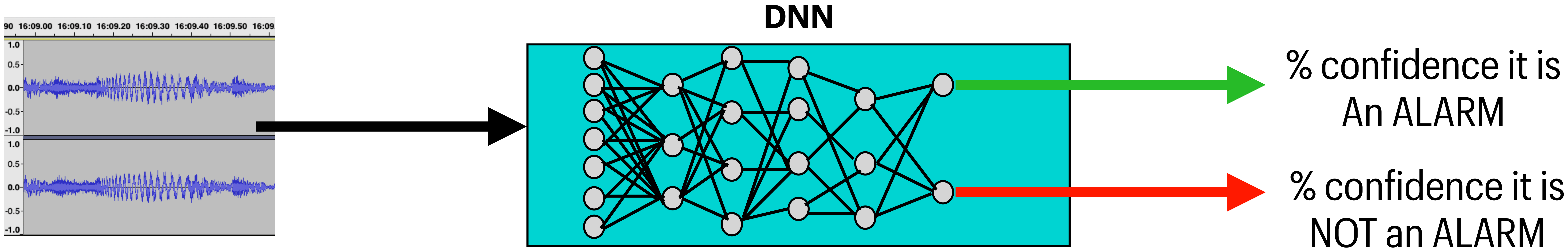
YES

ALARM

NO

NOT AN ALARM

**Filter with rule-based parameters**
**e.g. sound an alarm when the volume (1 feature) is above 10 decibels**



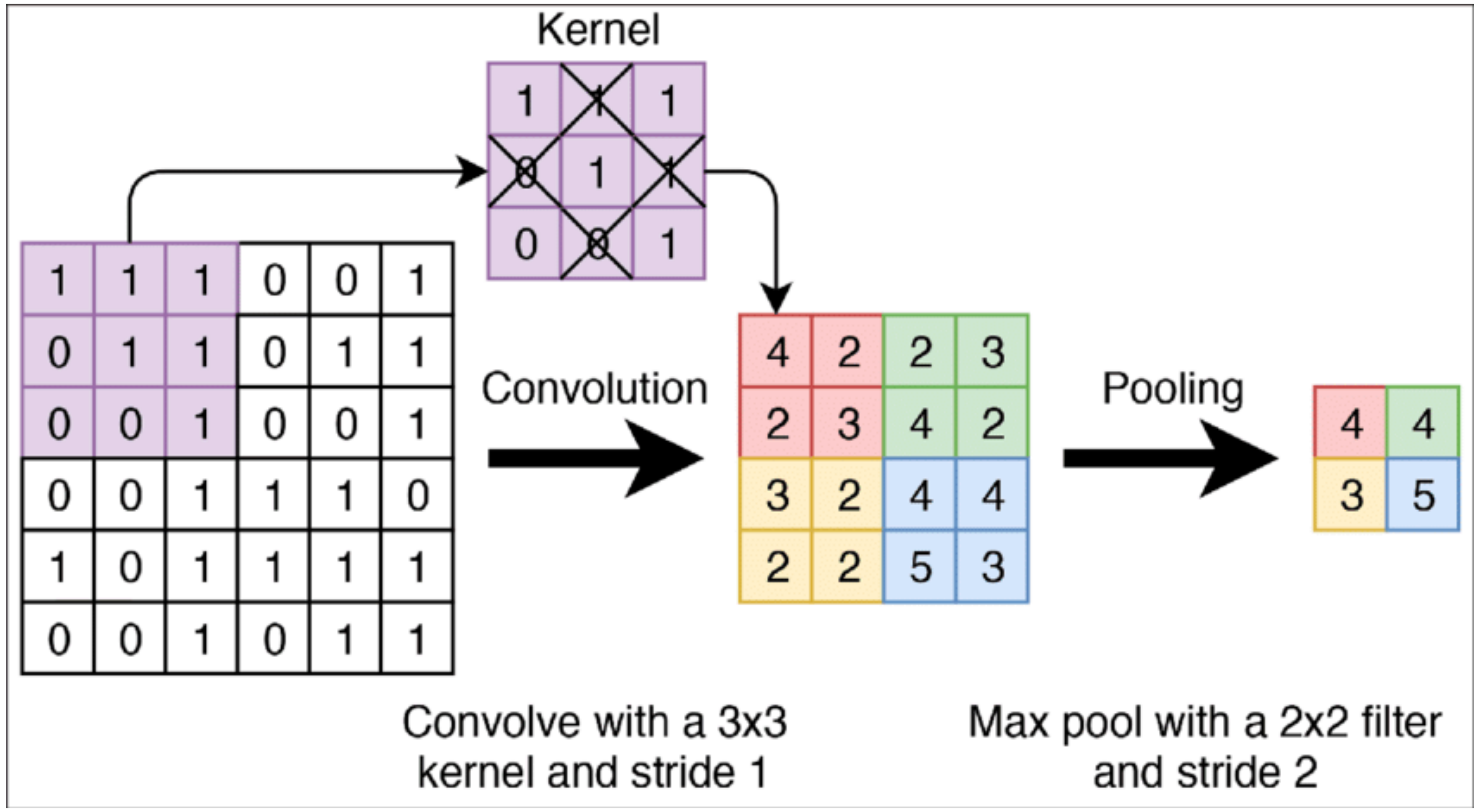
DNN

% confidence it is An ALARM

% confidence it is NOT an ALARM

**Filter with learned weights from training data/a training process**
**e.g. does the set of features (including volume, frequency, time, and combinations of those) from the input signal resemble those that an alarm situation would have?**

# C IS FOR CONVOLUTION:



Kernel

|  |  |  |
|---|---|---|
| 1 | ~~1~~ | 1 |
| ~~0~~ | 1 | ~~1~~ |
| 0 | ~~0~~ | 1 |

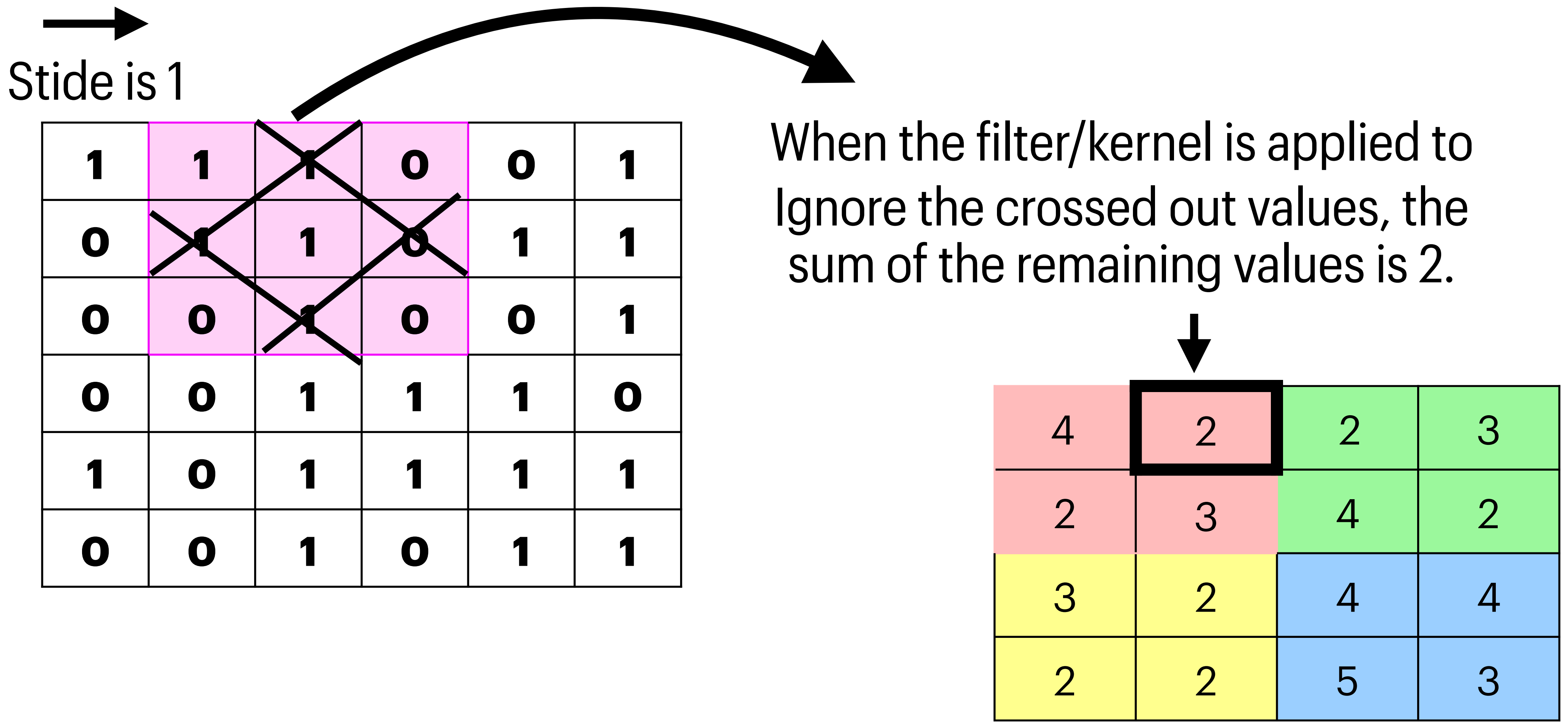
|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |

|  |  |  |  |
|---|---|---|---|
| 4 | 2 | 2 | 3 |
| 2 | 3 | 4 | 2 |
| 3 | 2 | 4 | 4 |
| 2 | 2 | 5 | 3 |

|  |  |
|---|---|
| 4 | 4 |
| 3 | 5 |

Convolve with a 3x3 kernel and stride 1

Max pool with a 2x2 filter and stride 2



|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 1 | 1 | ~~1~~ | 0 | 0 | 1 |
| 0 | ~~1~~ | 1 | ~~0~~ | 1 | 1 |
| 0 | 0 | ~~1~~ | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |

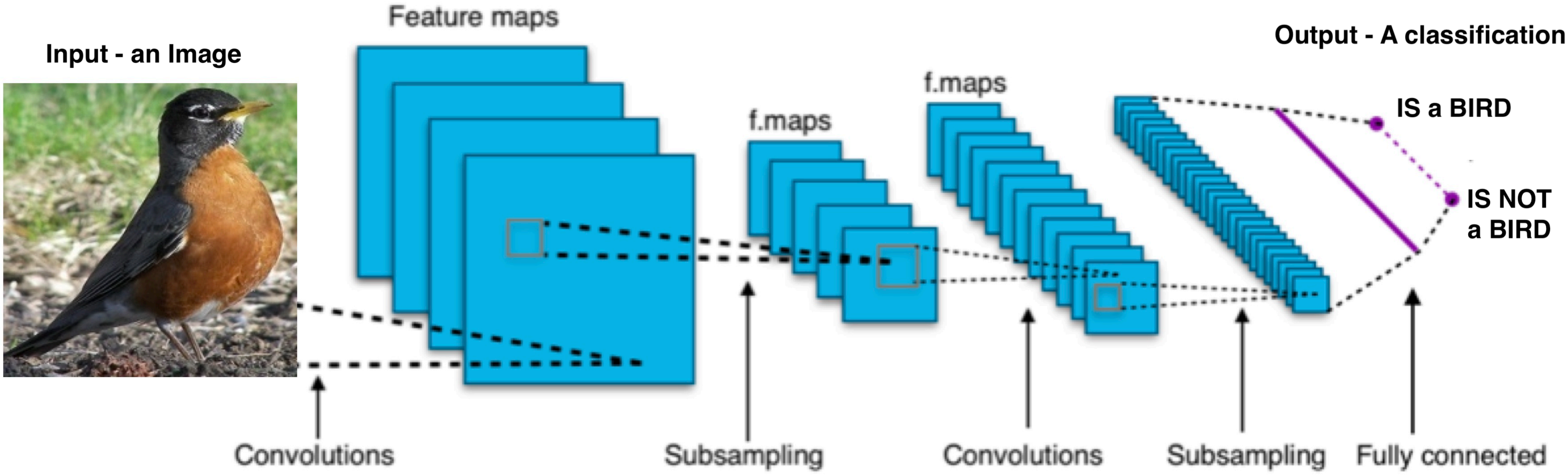
When the filter/kernel is applied to Ignore the crossed out values, the sum of the remaining values is 2.

|  |  |  |  |
|---|---|---|---|
| 4 | 2 | 2 | 3 |
| 2 | 3 | 4 | 2 |
| 3 | 2 | 4 | 4 |
| 2 | 2 | 5 | 3 |

**Convolution**:

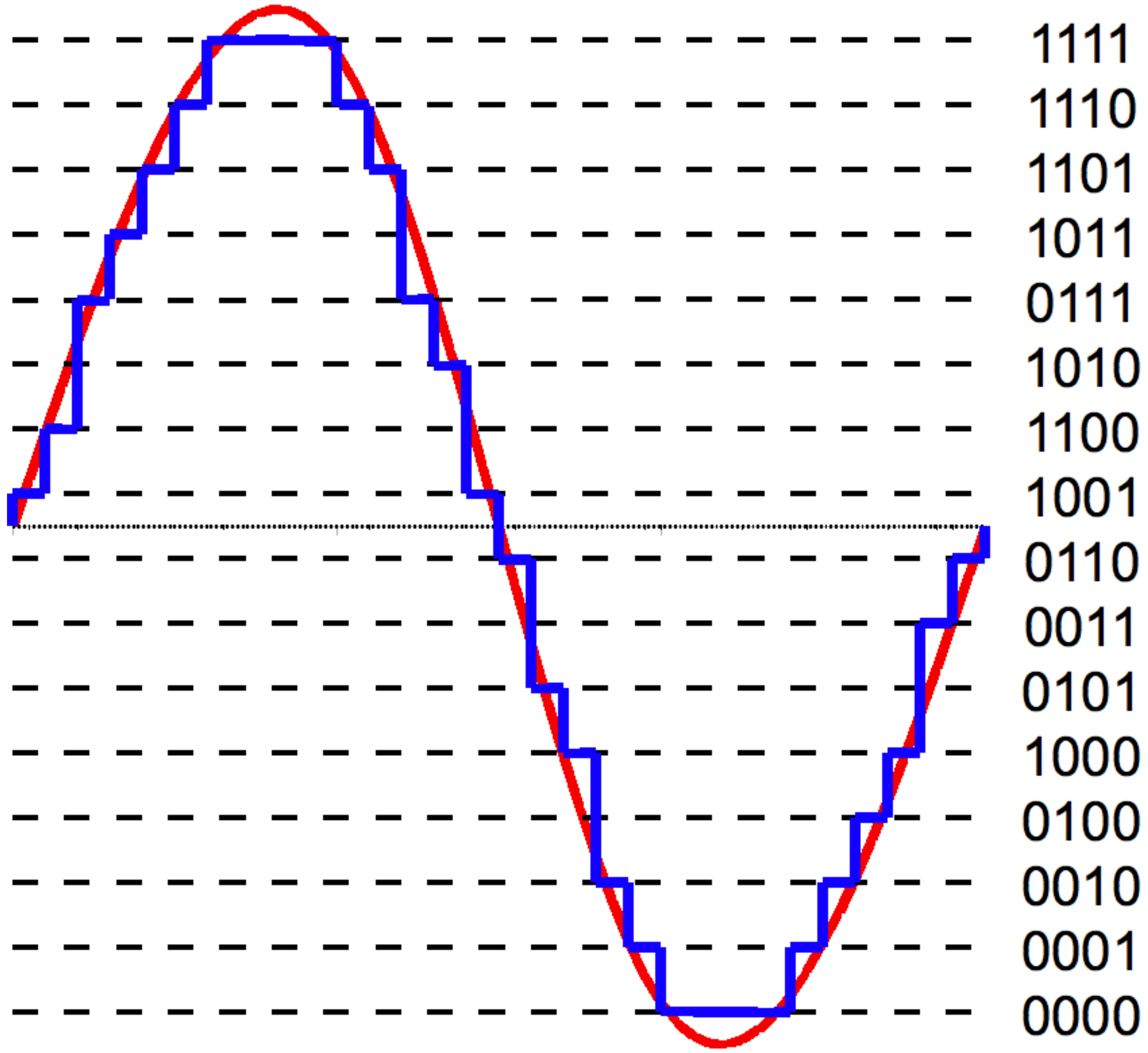
A filter (kernel) is used to add values of a pixel within an image to its neighboring pixels (based on a certain filter) resulting in feature maps. The subsequent subsampling (pooling) layer reduces the dimensions of these feature maps.

# CNN FOR IMAGE CLASSIFICATION

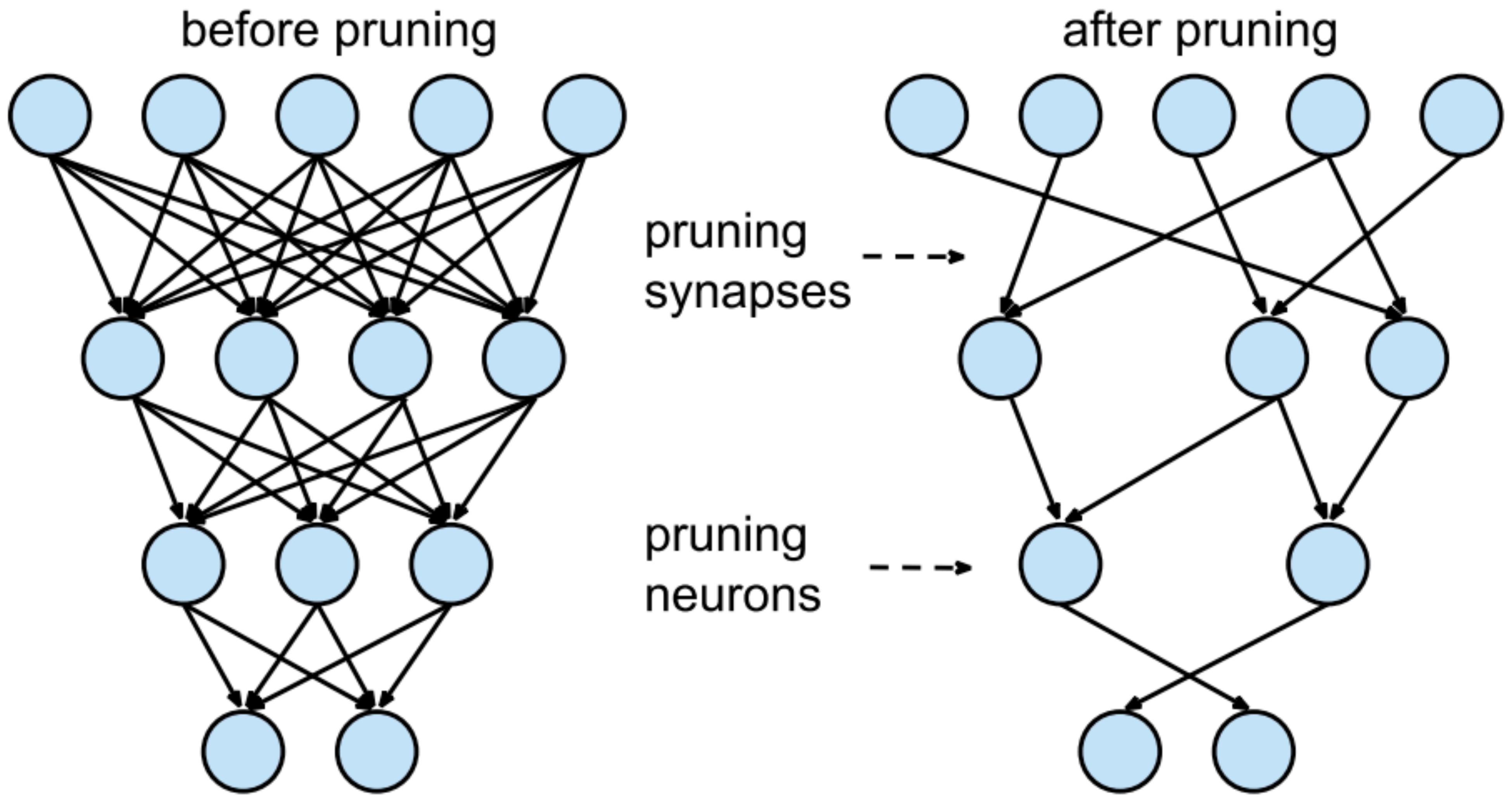


# MINIATURIZED ML - TINYML

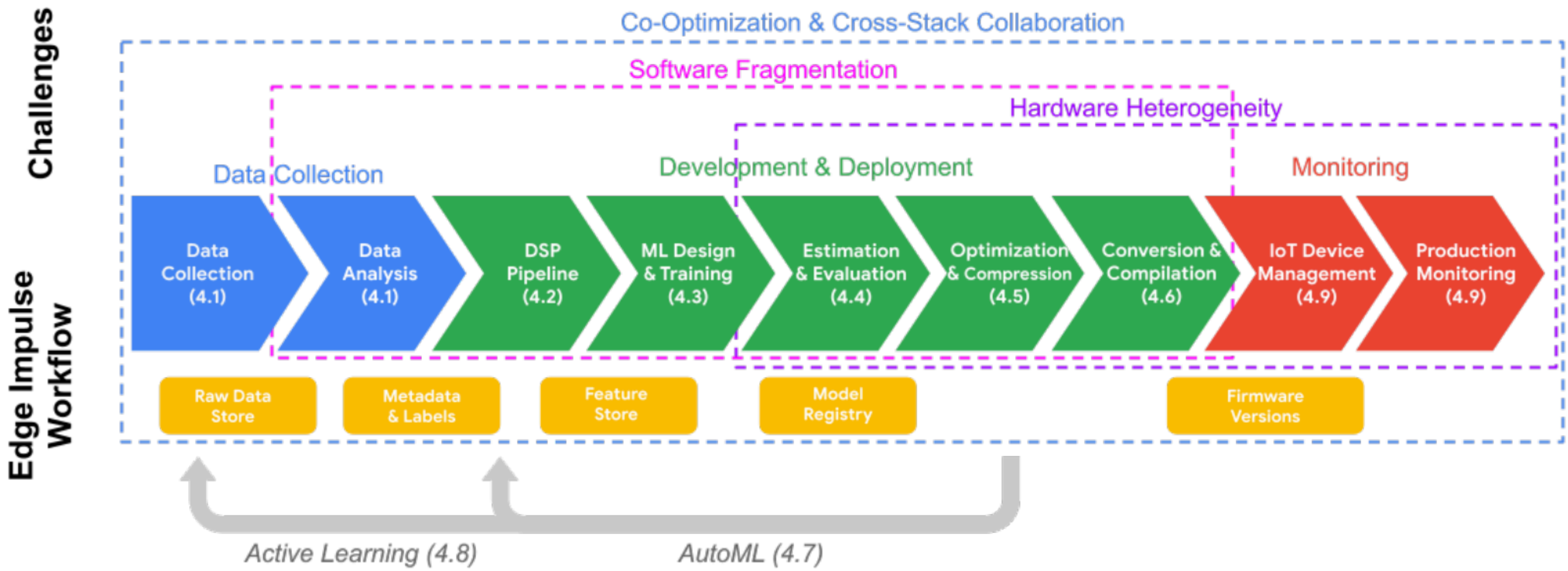
## Quantization:



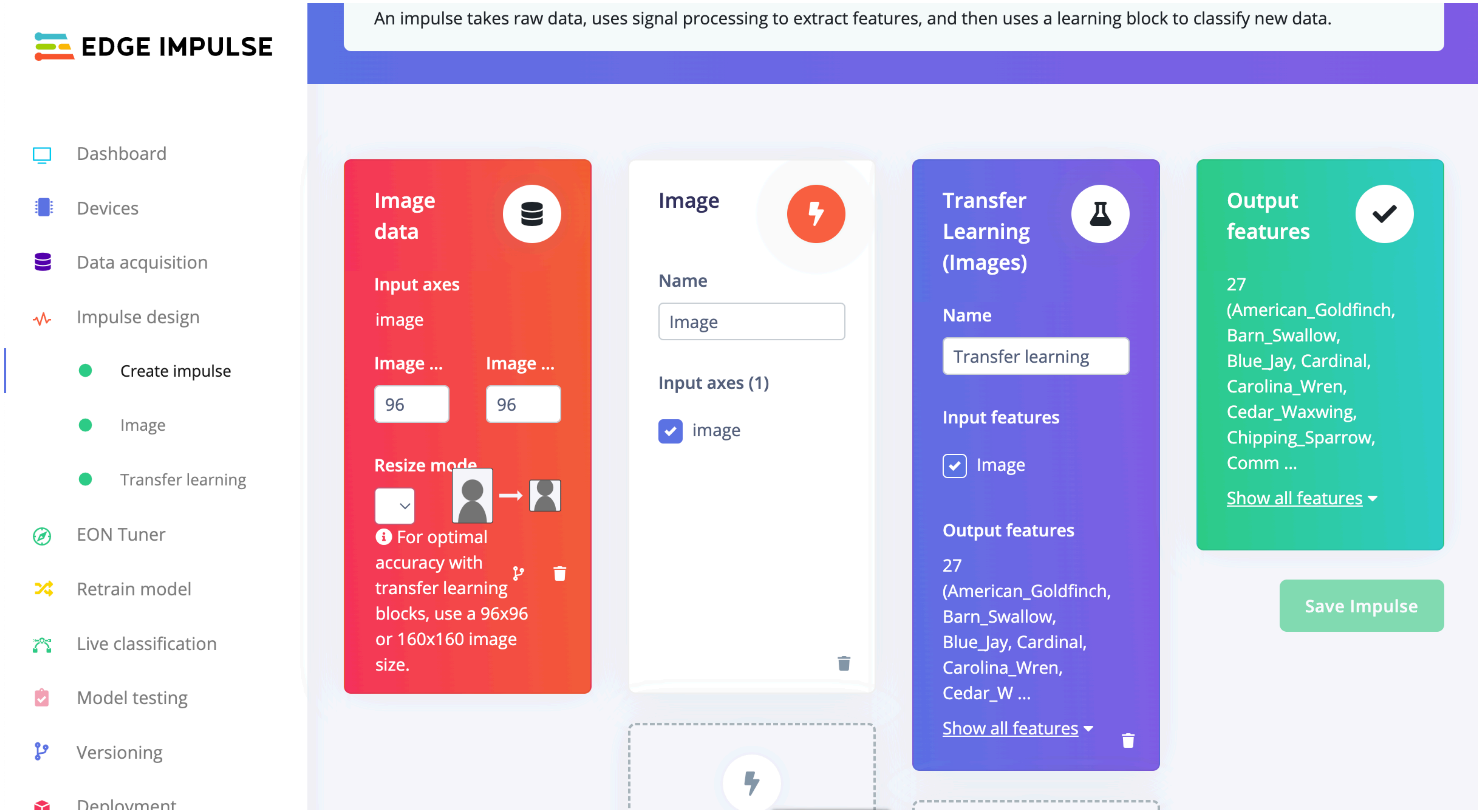
## Pruning::



# EDGE IMPULSE - THE PLATFORM



# EDGE IMPULSE USER INTERFACE



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

**Image data**

Input axes: image

Image ...: 96 Image ...: 96

Resize mode

For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.

**Image**

Name: Image

Input axes (1)

☑ image

**Transfer Learning (Images)**

Name: Transfer learning

Input features

☑ Image

Output features

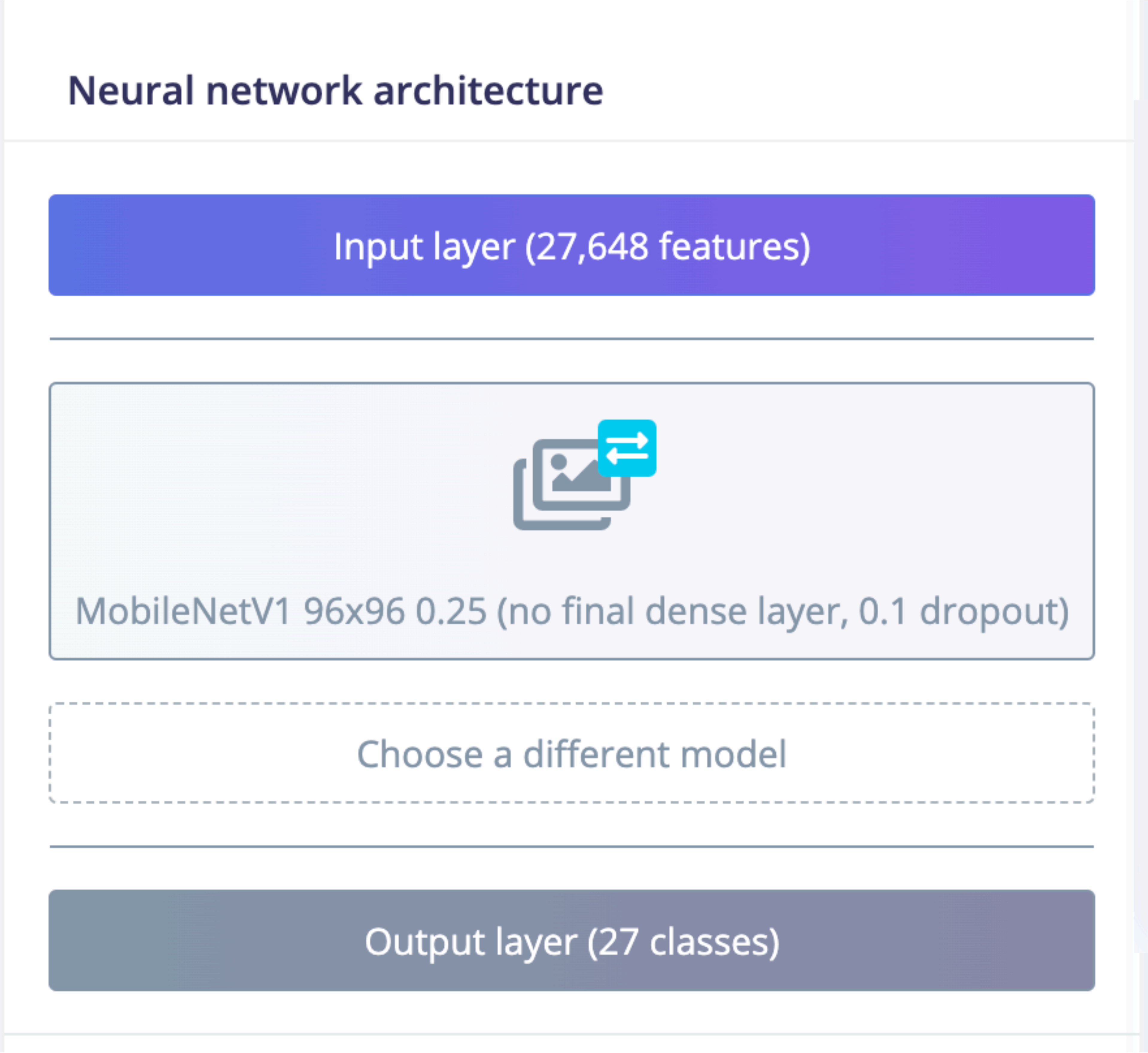
27 (American_Goldfinch, Barn_Swallow, Blue_Jay, Cardinal, Carolina_Wren, Cedar_W ...

**Output features**

27 (American_Goldfinch, Barn_Swallow, Blue_Jay, Cardinal, Carolina_Wren, Cedar_Waxwing, Chipping_Sparrow, Comm ...

# MOBILENET V1

## THE TINYML MODEL ARCHITECTURE



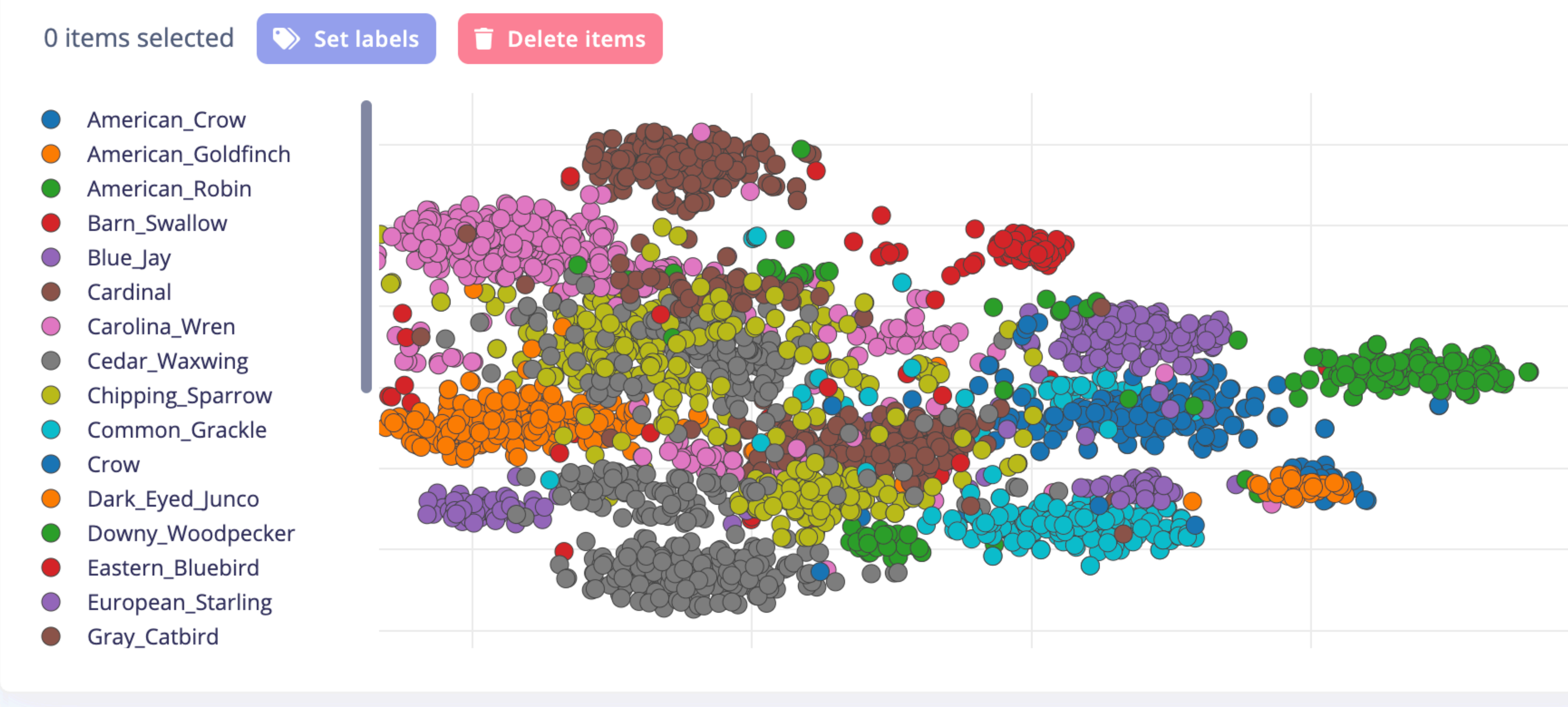
Neural network architecture

Input layer (27,648 features)

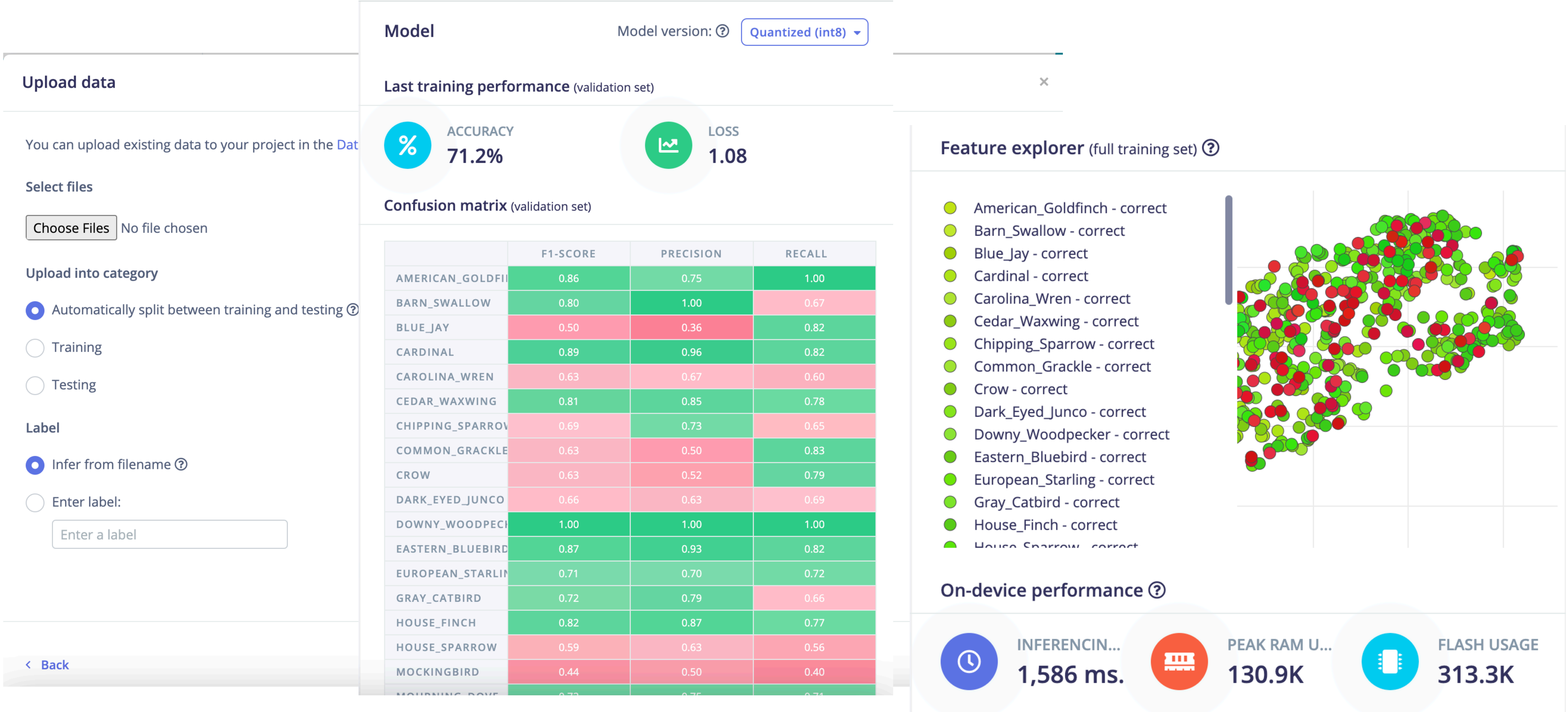
MobileNetV1 96x96 0.25 (no final dense layer, 0.1 dropout)

Choose a different model

Output layer (27 classes)



# ML TRAINING ON EDGE IMPULSE



Upload data

You can upload existing data to your project in the Dat

Select files

Choose Files No file chosen

Upload into category

● Automatically split between training and testing ?

○ Training

○ Testing

Label

● Infer from filename ?

○ Enter label:

Enter a label

‹ Back

Model

Model version: ? Quantized (int8)

Last training performance (validation set)

ACCURACY 71.2%

LOSS 1.08

Confusion matrix (validation set)

| | F1-SCORE | PRECISION | RECALL |
|---|---|---|---|
| AMERICAN_GOLDFI | 0.86 | 0.75 | 1.00 |
| BARN_SWALLOW | 0.80 | 1.00 | 0.67 |
| BLUE_JAY | 0.50 | 0.36 | 0.82 |
| CARDINAL | 0.89 | 0.96 | 0.82 |
| CAROLINA_WREN | 0.63 | 0.67 | 0.60 |
| CEDAR_WAXWING | 0.81 | 0.85 | 0.78 |
| CHIPPING_SPARROV | 0.69 | 0.73 | 0.65 |
| COMMON_GRACKLE | 0.63 | 0.50 | 0.83 |
| CROW | 0.63 | 0.52 | 0.79 |
| DARK_EYED_JUNCO | 0.66 | 0.63 | 0.69 |
| DOWNY_WOODPECI | 1.00 | 1.00 | 1.00 |
| EASTERN_BLUEBIRD | 0.87 | 0.93 | 0.82 |
| EUROPEAN_STARLII | 0.71 | 0.70 | 0.72 |
| GRAY_CATBIRD | 0.72 | 0.79 | 0.66 |
| HOUSE_FINCH | 0.82 | 0.87 | 0.77 |
| HOUSE_SPARROW | 0.59 | 0.63 | 0.56 |
| MOCKINGBIRD | 0.44 | 0.50 | 0.40 |
| MOURNING_DOVE | 0.72 | 0.75 | 0.71 |

Feature explorer (full training set) ?

- American_Goldfinch - correct
- Barn_Swallow - correct
- Blue_Jay - correct
- Cardinal - correct
- Carolina_Wren - correct
- Cedar_Waxwing - correct
- Chipping_Sparrow - correct
- Common_Grackle - correct
- Crow - correct
- Dark_Eyed_Junco - correct
- Downy_Woodpecker - correct
- Eastern_Bluebird - correct
- European_Starling - correct
- Gray_Catbird - correct
- House_Finch - correct
- House_Sparrow - correct

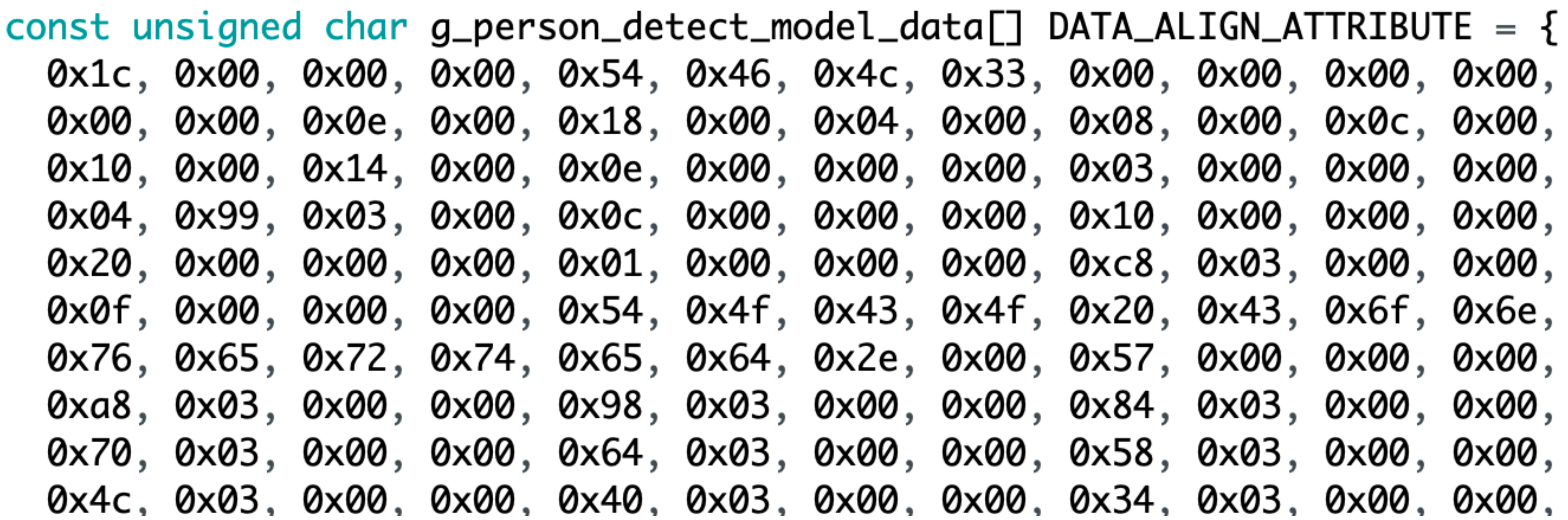
On-device performance ?

INFERENCIN... 1,586 ms.

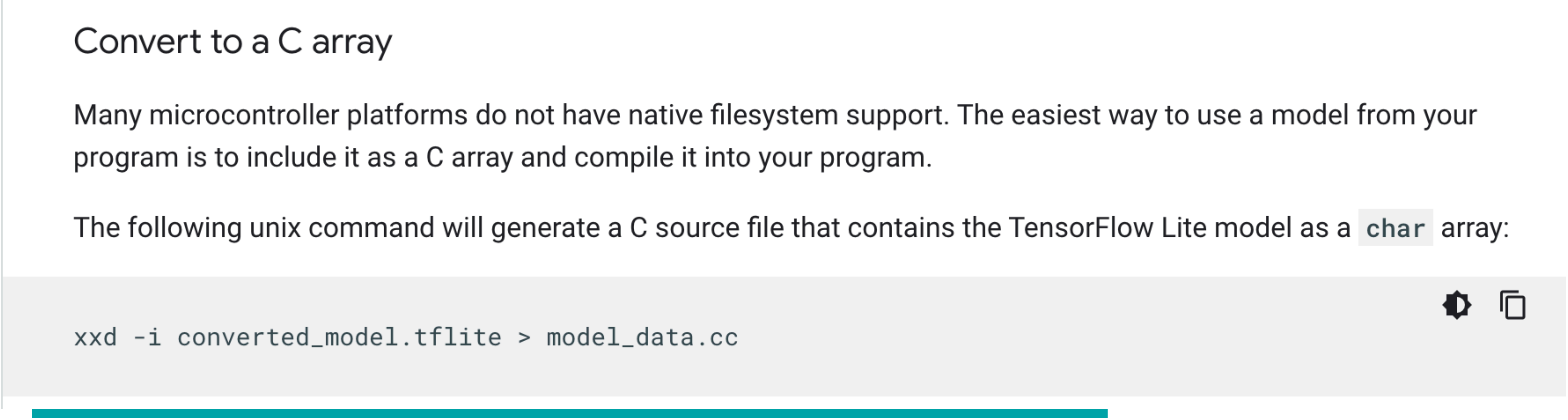
PEAK RAM U... 130.9K

FLASH USAGE 313.3K

# TENSORFLOW LITE ESP32 LIBRARY



```
const unsigned char g_person_detect_model_data[] DATA_ALIGN_ATTRIBUTE = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0e, 0x00, 0x18, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x10, 0x00, 0x14, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
  0x04, 0x99, 0x03, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00,
  0x20, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0xc8, 0x03, 0x00, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x54, 0x4f, 0x43, 0x4f, 0x20, 0x43, 0x6f, 0x6e,
  0x76, 0x65, 0x72, 0x74, 0x65, 0x64, 0x2e, 0x00, 0x57, 0x00, 0x00, 0x00,
  0xa8, 0x03, 0x00, 0x00, 0x98, 0x03, 0x00, 0x00, 0x84, 0x03, 0x00, 0x00,
  0x70, 0x03, 0x00, 0x00, 0x64, 0x03, 0x00, 0x00, 0x58, 0x03, 0x00, 0x00,
  0x4c, 0x03, 0x00, 0x00, 0x40, 0x03, 0x00, 0x00, 0x34, 0x03, 0x00, 0x00,
```



## Convert to a C array

Many microcontroller platforms do not have native filesystem support. The easiest way to use a model from your program is to include it as a C array and compile it into your program.

The following unix command will generate a C source file that contains the TensorFlow Lite model as a `char` array:

```
xxd -i converted_model.tflite > model_data.cc
```

# DEPLOYING E-I ML ON ESP32-EYE

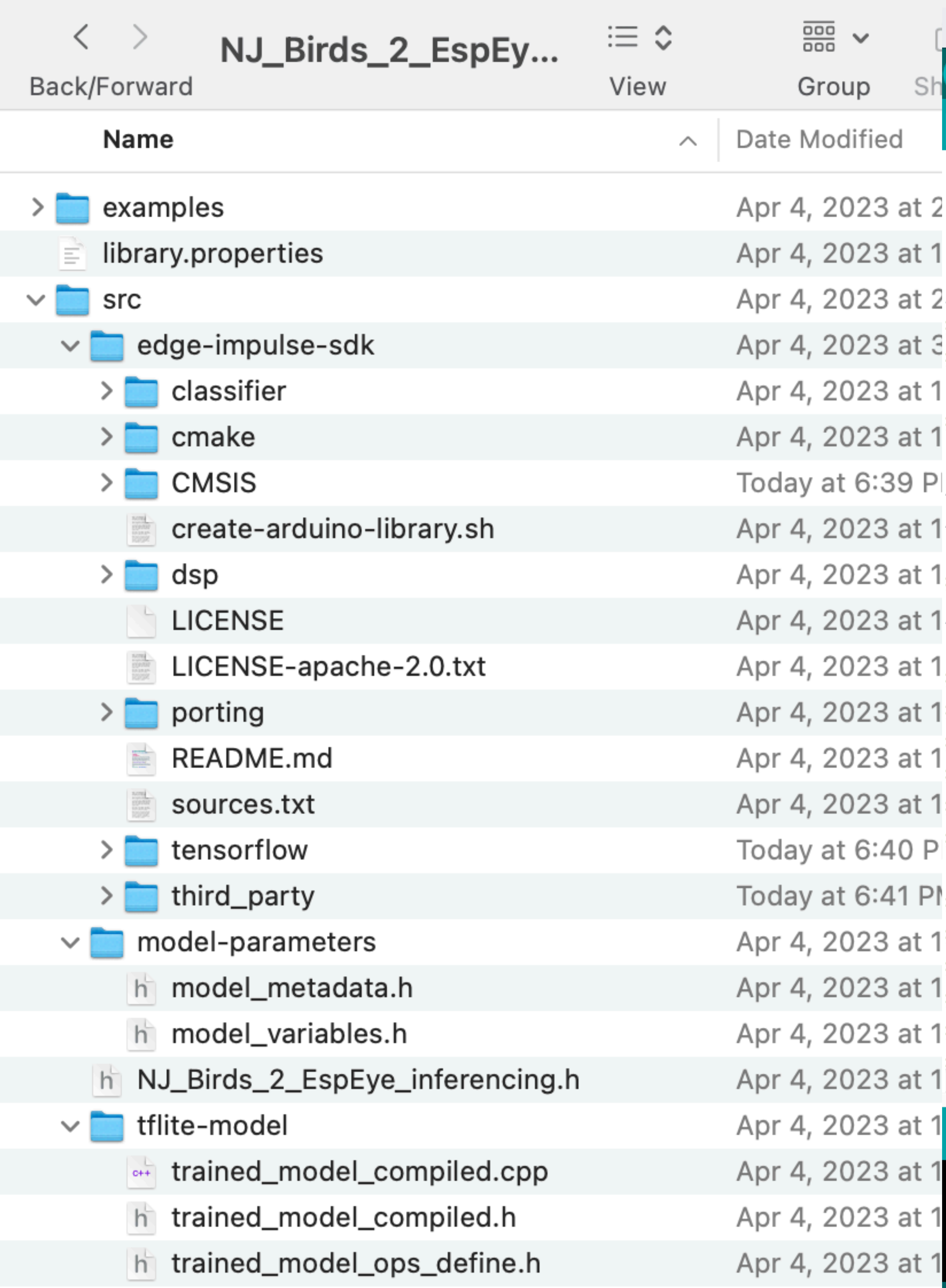
## Built Arduino library

Add this library through the Arduino IDE via:

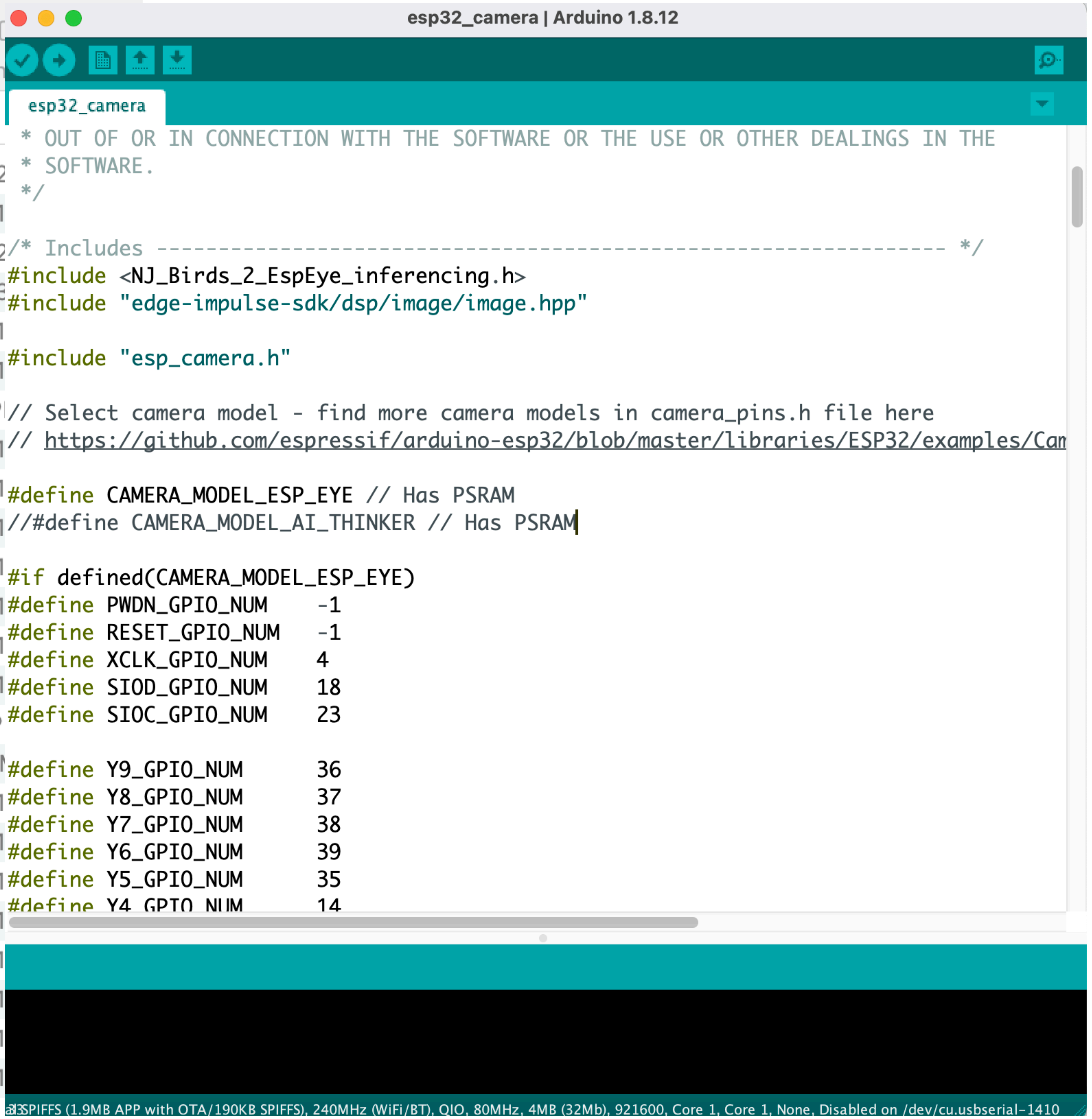
**`Sketch > Include Library > Add .ZIP Librai`**

Examples can then be found under:

**`File > Examples > NJ_Birds_2_EspEye_infere`**



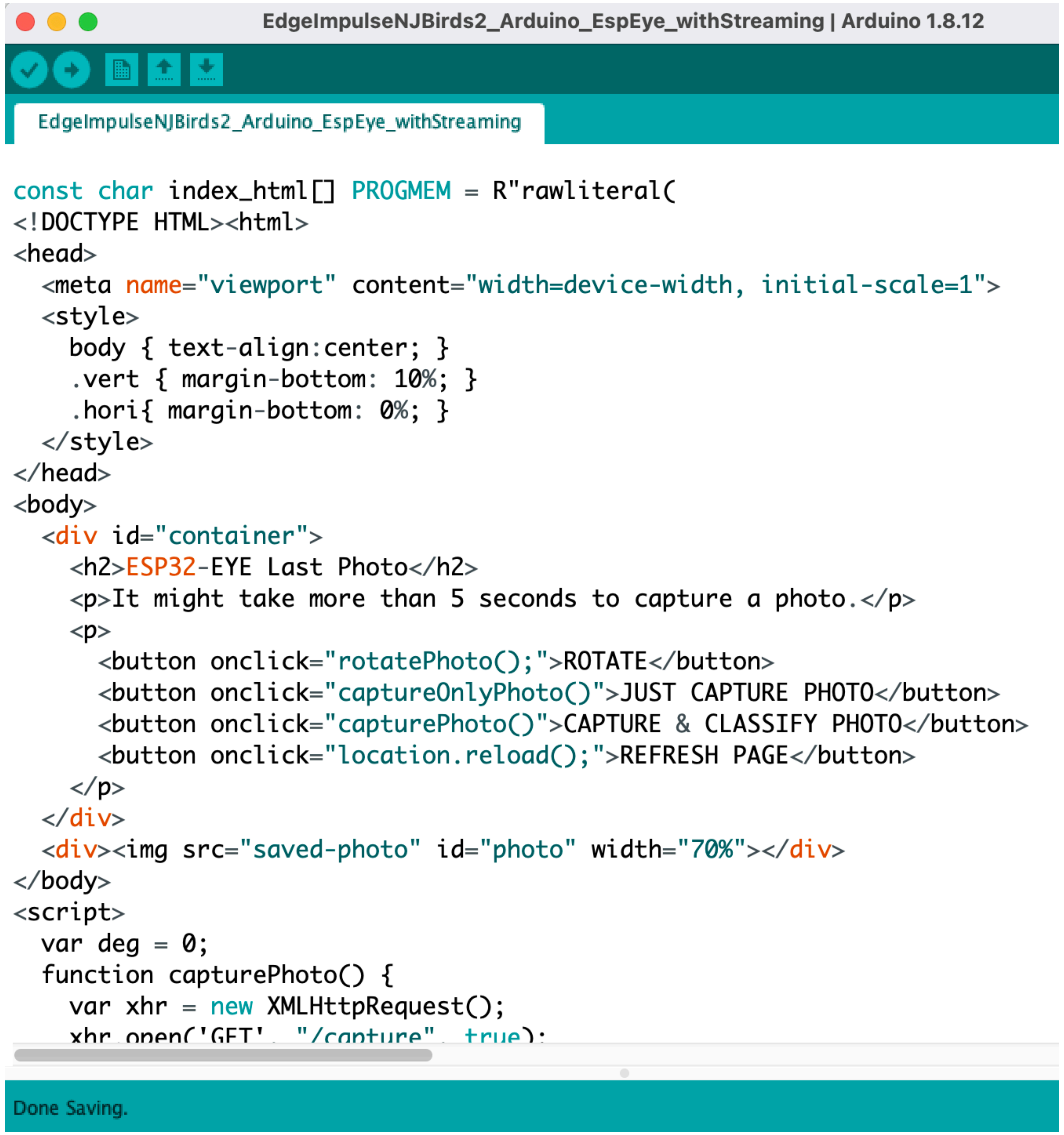
| Name | Date Modified |
|---|---|
| examples | Apr 4, 2023 at 2 |
| library.properties | Apr 4, 2023 at 1 |
| src | Apr 4, 2023 at 2 |
| edge-impulse-sdk | Apr 4, 2023 at 3 |
| classifier | Apr 4, 2023 at 1 |
| cmake | Apr 4, 2023 at 1 |
| CMSIS | Today at 6:39 P |
| create-arduino-library.sh | Apr 4, 2023 at 1 |
| dsp | Apr 4, 2023 at 1 |
| LICENSE | Apr 4, 2023 at 1 |
| LICENSE-apache-2.0.txt | Apr 4, 2023 at 1 |
| porting | Apr 4, 2023 at 1 |
| README.md | Apr 4, 2023 at 1 |
| sources.txt | Apr 4, 2023 at 1 |
| tensorflow | Today at 6:40 P |
| third_party | Today at 6:41 PI |
| model-parameters | Apr 4, 2023 at 1 |
| model_metadata.h | Apr 4, 2023 at 1 |
| model_variables.h | Apr 4, 2023 at 1 |
| NJ_Birds_2_EspEye_inferencing.h | Apr 4, 2023 at 1 |
| tflite-model | Apr 4, 2023 at 1 |
| trained_model_compiled.cpp | Apr 4, 2023 at 1 |
| trained_model_compiled.h | Apr 4, 2023 at 1 |
| trained_model_ops_define.h | Apr 4, 2023 at 1 |



```
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

/* Includes ---------------------------------------------------------------- */
#include <NJ_Birds_2_EspEye_inferencing.h>
#include "edge-impulse-sdk/dsp/image/image.hpp"

#include "esp_camera.h"

// Select camera model - find more camera models in camera_pins.h file here
// https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP32/examples/Cam

#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#if defined(CAMERA_MODEL_ESP_EYE)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    4
#define SIOD_GPIO_NUM    18
#define SIOC_GPIO_NUM    23

#define Y9_GPIO_NUM      36
#define Y8_GPIO_NUM      37
#define Y7_GPIO_NUM      38
#define Y6_GPIO_NUM      39
#define Y5_GPIO_NUM      35
#define Y4_GPIO_NUM      14
```

```
a33SPIFFS (1.9MB APP with OTA/190KB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on /dev/cu.usbserial-1410
```

# CUSTOMIZATION

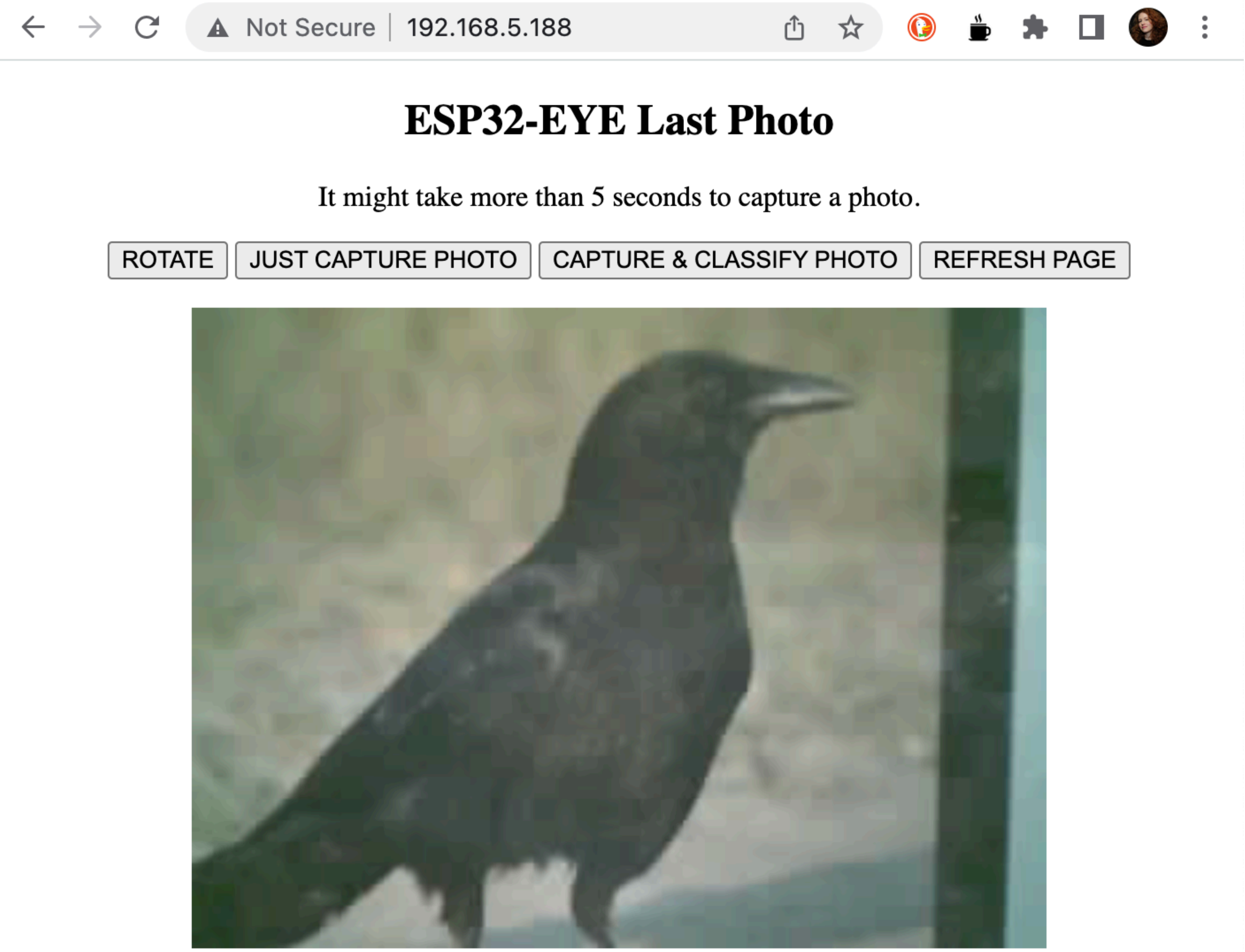


EdgeImpulseNJBirds2_Arduino_EspEye_withStreaming | Arduino 1.8.12

EdgeImpulseNJBirds2_Arduino_EspEye_withStreaming

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body { text-align:center; }
    .vert { margin-bottom: 10%; }
    .hori{ margin-bottom: 0%; }
  </style>
</head>
<body>
  <div id="container">
    <h2>ESP32-EYE Last Photo</h2>
    <p>It might take more than 5 seconds to capture a photo.</p>
    <p>
      <button onclick="rotatePhoto();">ROTATE</button>
      <button onclick="captureOnlyPhoto()">JUST CAPTURE PHOTO</button>
      <button onclick="capturePhoto()">CAPTURE & CLASSIFY PHOTO</button>
      <button onclick="location.reload();">REFRESH PAGE</button>
    </p>
  </div>
  <div><img src="saved-photo" id="photo" width="70%"></div>
</body>
<script>
  var deg = 0;
  function capturePhoto() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', "/capture", true);
```

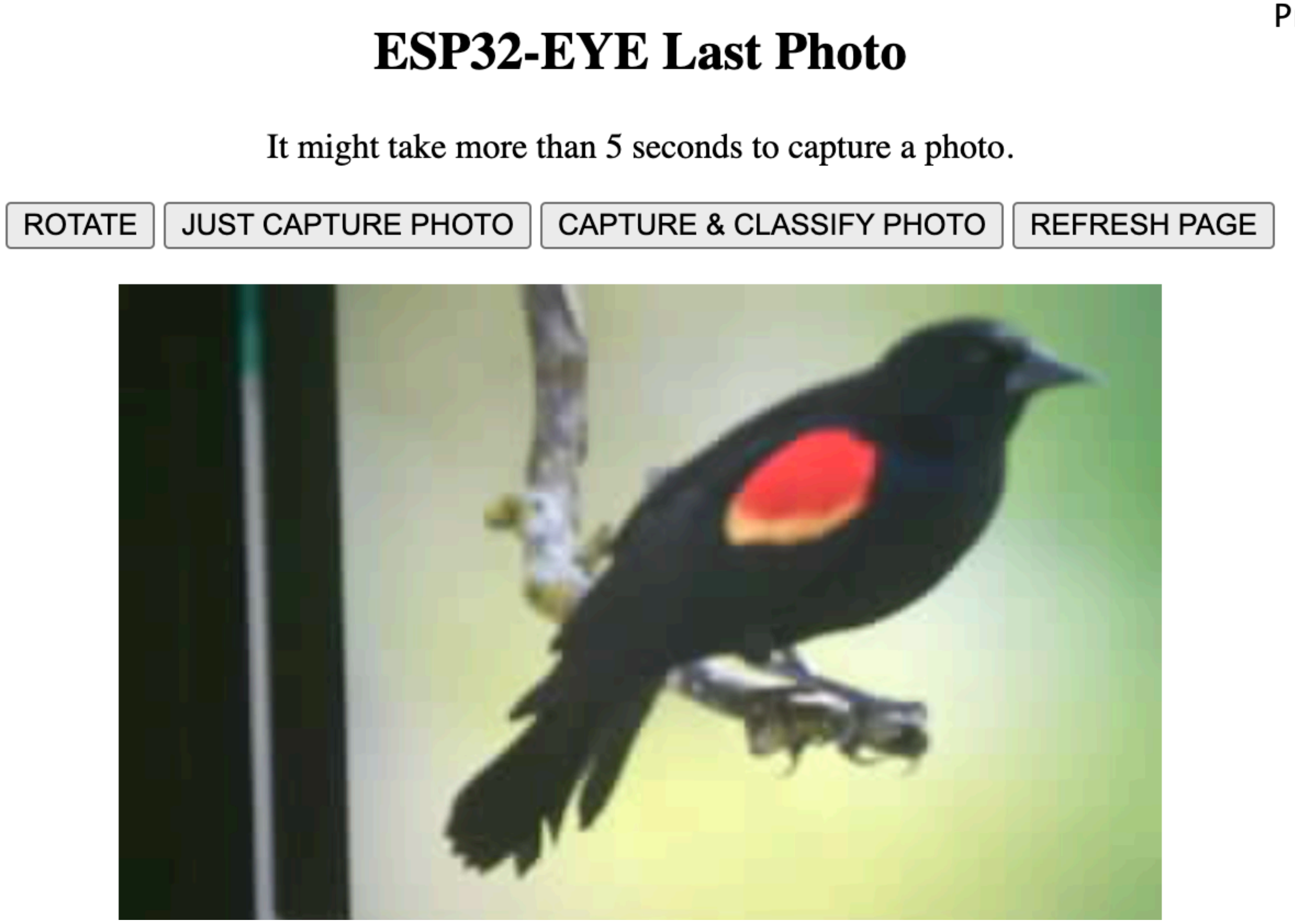
Done Saving.



ESP32-EYE Last Photo

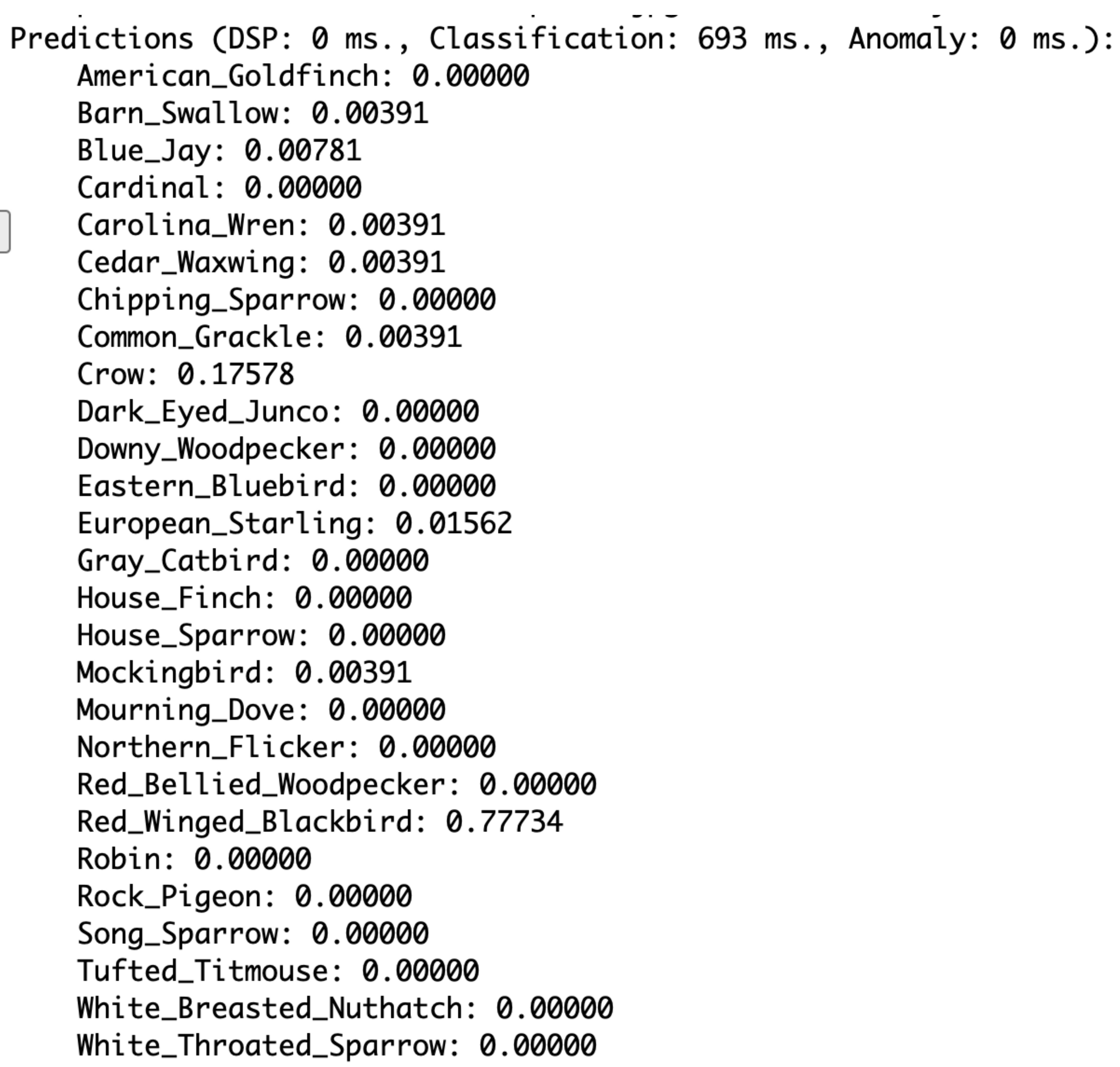
It might take more than 5 seconds to capture a photo.

# TEST OF MODEL ON ESP32-EYE



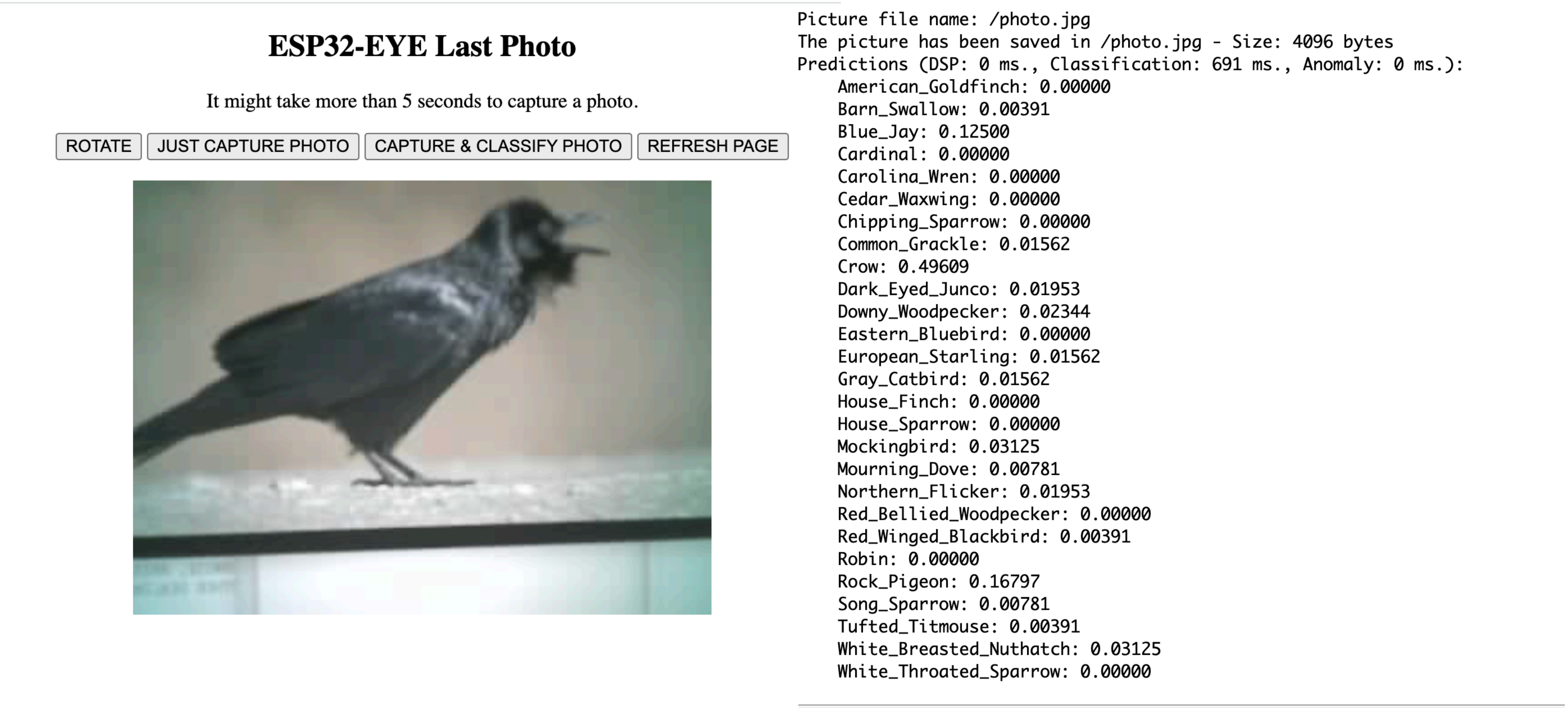
ESP32-EYE Last Photo

It might take more than 5 seconds to capture a photo.



```
Predictions (DSP: 0 ms., Classification: 693 ms., Anomaly: 0 ms.):
    American_Goldfinch: 0.00000
    Barn_Swallow: 0.00391
    Blue_Jay: 0.00781
    Cardinal: 0.00000
    Carolina_Wren: 0.00391
    Cedar_Waxwing: 0.00391
    Chipping_Sparrow: 0.00000
    Common_Grackle: 0.00391
    Crow: 0.17578
    Dark_Eyed_Junco: 0.00000
    Downy_Woodpecker: 0.00000
    Eastern_Bluebird: 0.00000
    European_Starling: 0.01562
    Gray_Catbird: 0.00000
    House_Finch: 0.00000
    House_Sparrow: 0.00000
    Mockingbird: 0.00391
    Mourning_Dove: 0.00000
    Northern_Flicker: 0.00000
    Red_Bellied_Woodpecker: 0.00000
    Red_Winged_Blackbird: 0.77734
    Robin: 0.00000
    Rock_Pigeon: 0.00000
    Song_Sparrow: 0.00000
    Tufted_Titmouse: 0.00000
    White_Breasted_Nuthatch: 0.00000
    White_Throated_Sparrow: 0.00000
```

# ITERATIONS

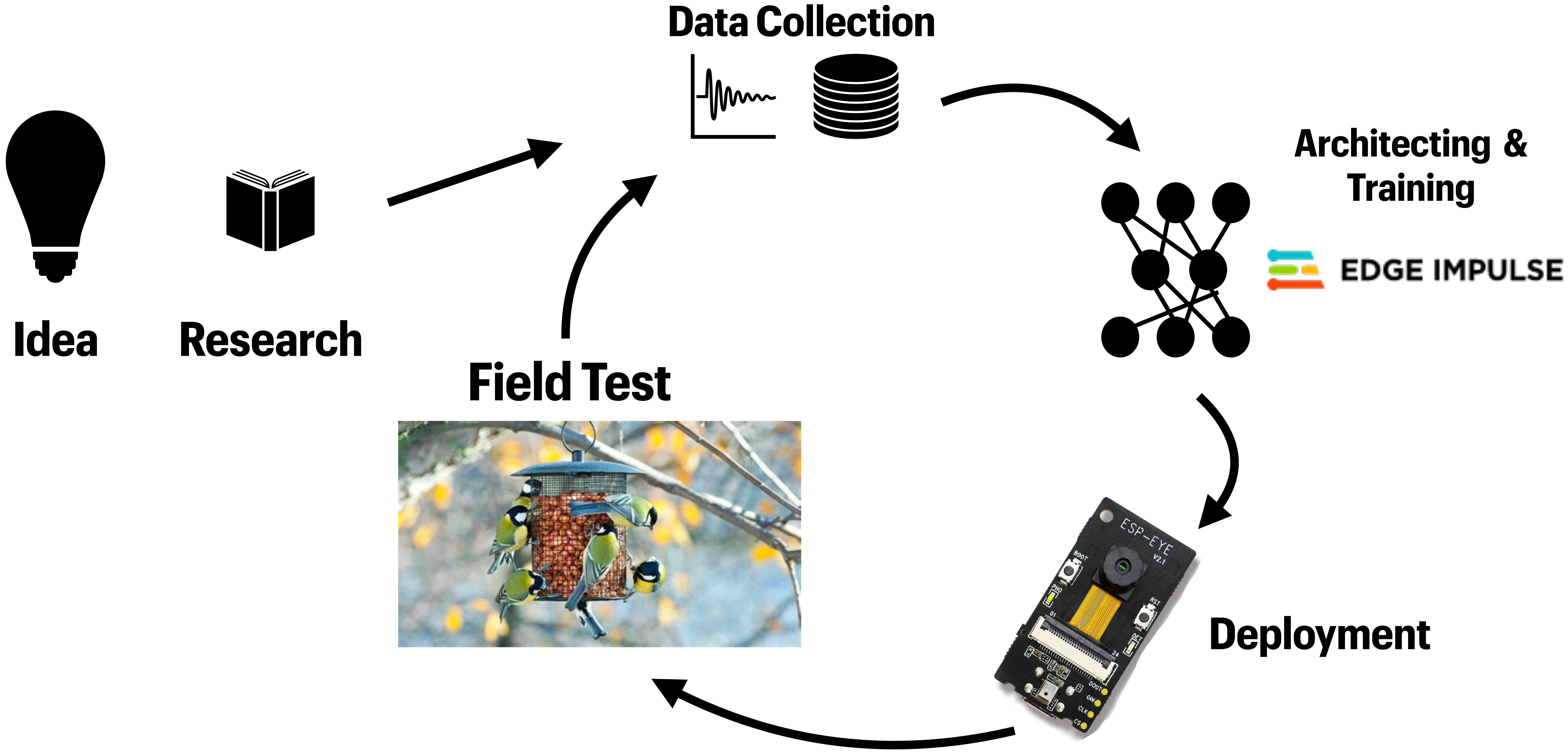


## ESP32-EYE Last Photo

It might take more than 5 seconds to capture a photo.

```
Picture file name: /photo.jpg
The picture has been saved in /photo.jpg - Size: 4096 bytes
Predictions (DSP: 0 ms., Classification: 691 ms., Anomaly: 0 ms.):
    American_Goldfinch: 0.00000
    Barn_Swallow: 0.00391
    Blue_Jay: 0.12500
    Cardinal: 0.00000
    Carolina_Wren: 0.00000
    Cedar_Waxwing: 0.00000
    Chipping_Sparrow: 0.00000
    Common_Grackle: 0.01562
    Crow: 0.49609
    Dark_Eyed_Junco: 0.01953
    Downy_Woodpecker: 0.02344
    Eastern_Bluebird: 0.00000
    European_Starling: 0.01562
    Gray_Catbird: 0.01562
    House_Finch: 0.00000
    House_Sparrow: 0.00000
    Mockingbird: 0.03125
    Mourning_Dove: 0.00781
    Northern_Flicker: 0.01953
    Red_Bellied_Woodpecker: 0.00000
    Red_Winged_Blackbird: 0.00391
    Robin: 0.00000
    Rock_Pigeon: 0.16797
    Song_Sparrow: 0.00781
    Tufted_Titmouse: 0.00391
    White_Breasted_Nuthatch: 0.03125
    White_Throated_Sparrow: 0.00000
```

# DEVELOPMENT CYCLE



Idea

Research

Data Collection

Architecting & Training

EDGE IMPULSE

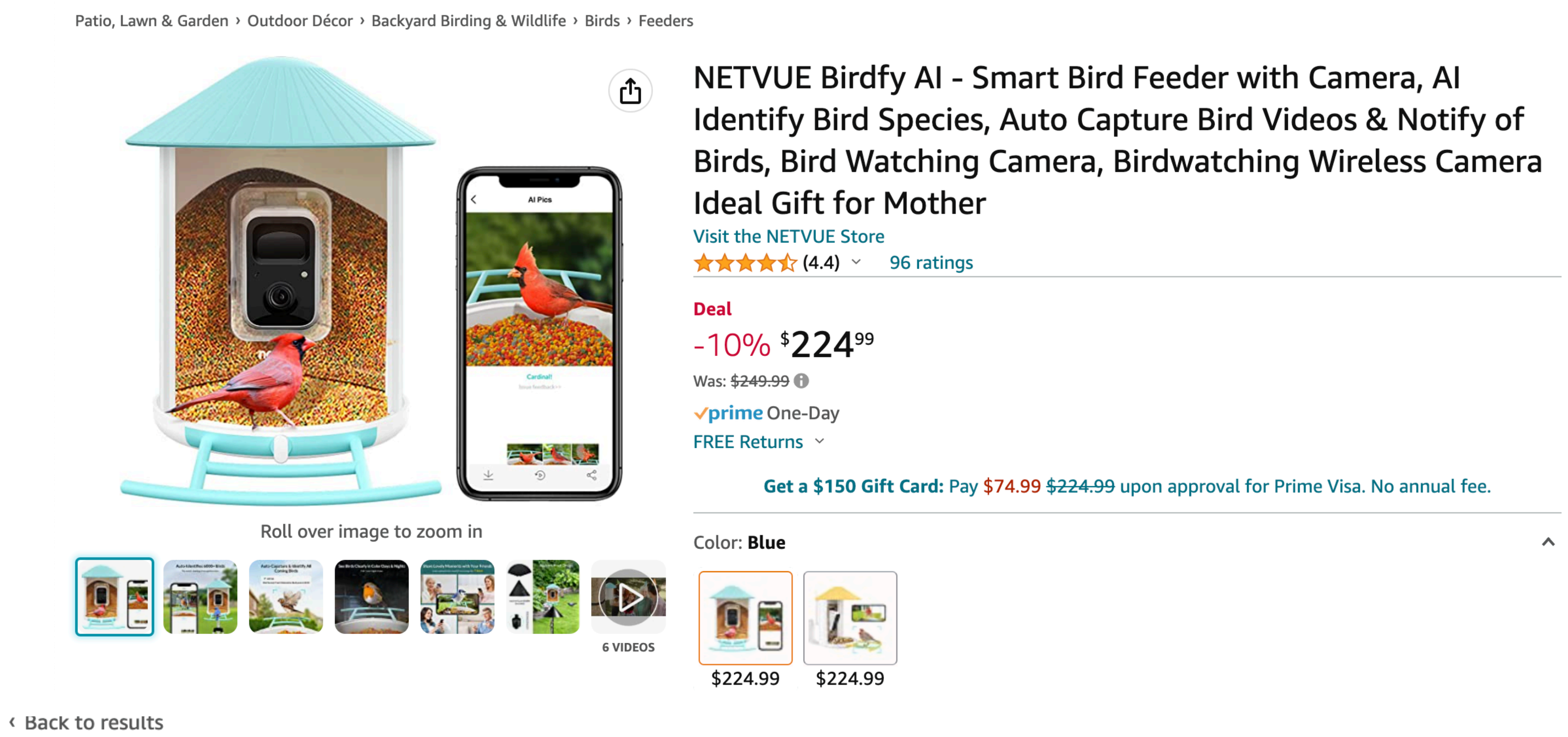
Deployment

Field Test

# FUTURE DEVELOPMENT

- Events triggered after bully bird detection
- Form factor of feeder
- Additional feeder features/automation
- Locality customization
- Season customization
- Human validation of field test results
- User testing UI/UX
- Hardware upgrades

# MARKET RESEARCH - OTHER PRODUCTS



Patio, Lawn & Garden › Outdoor Décor › Backyard Birding & Wildlife › Birds › Feeders

NETVUE Birdfy AI - Smart Bird Feeder with Camera, AI Identify Bird Species, Auto Capture Bird Videos & Notify of Birds, Bird Watching Camera, Birdwatching Wireless Camera Ideal Gift for Mother

Visit the NETVUE Store

★★★★☆ (4.4) 96 ratings

Deal

-10% $224.99

Was: ~~$249.99~~

prime One-Day

FREE Returns

Get a $150 Gift Card: Pay $74.99 ~~$224.99~~ upon approval for Prime Visa. No annual fee.

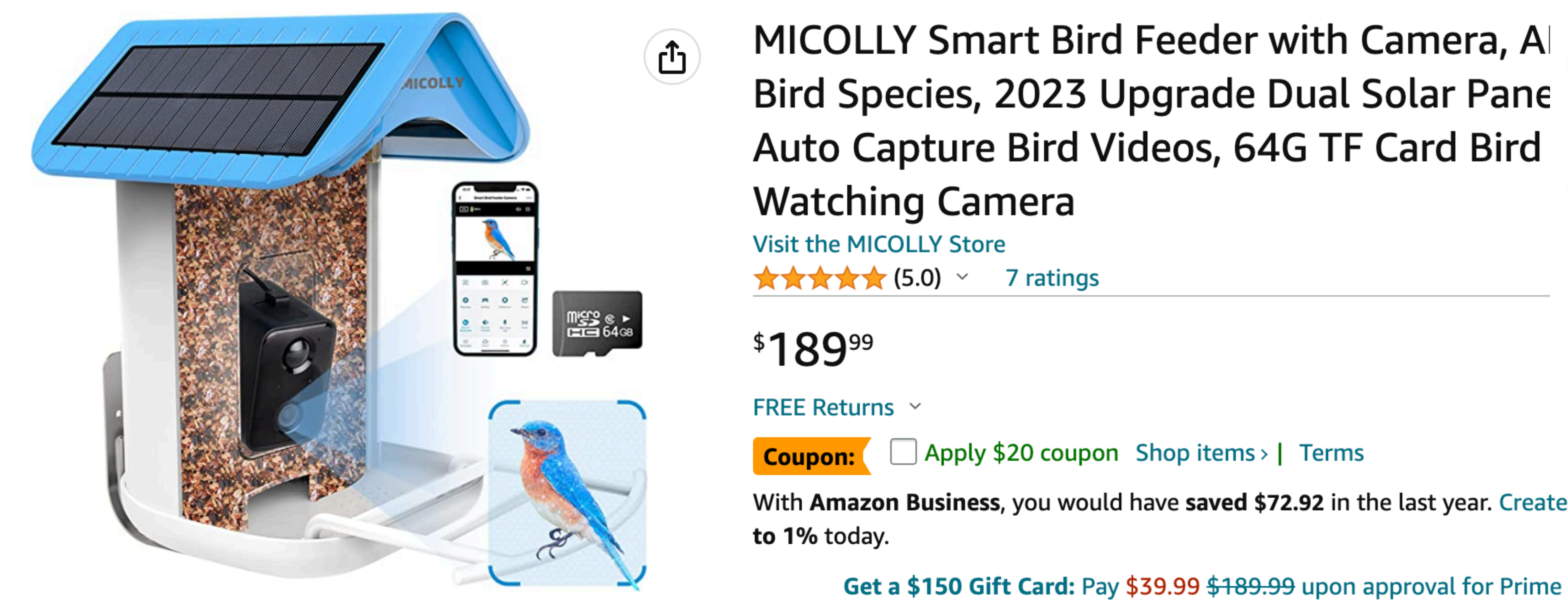
Roll over image to zoom in

6 VIDEOS

Color: Blue

$224.99 $224.99

‹ Back to results



MICOLLY Smart Bird Feeder with Camera, AI Bird Species, 2023 Upgrade Dual Solar Pane Auto Capture Bird Videos, 64G TF Card Bird Watching Camera

Visit the MICOLLY Store

★★★★★ (5.0) 7 ratings

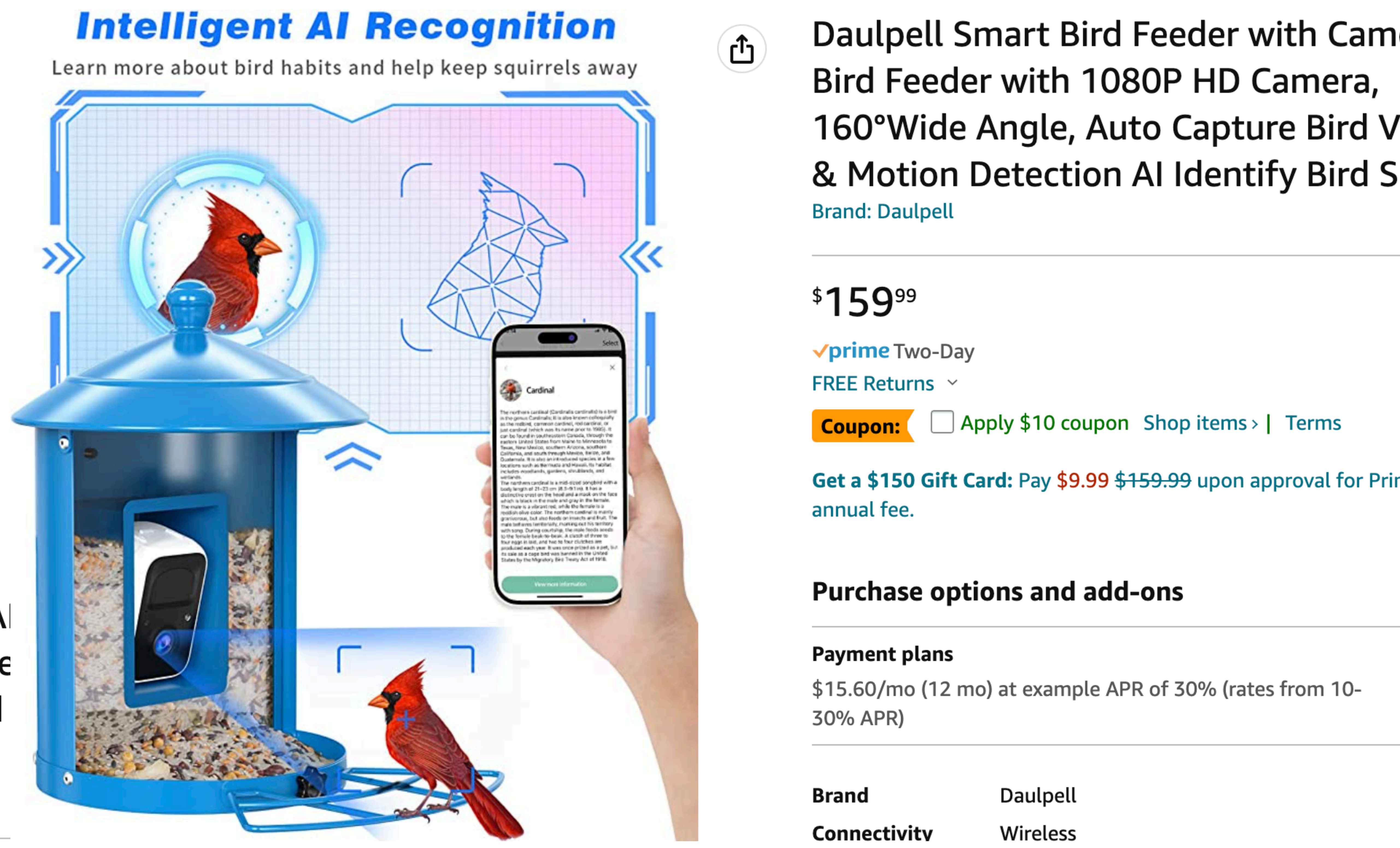
$189.99

FREE Returns

Coupon: □ Apply $20 coupon Shop items › | Terms

With **Amazon Business**, you would have **saved $72.92** in the last year. Create a free account and **save up to 1%** today.

Get a $150 Gift Card: Pay $39.99 ~~$189.99~~ upon approval for Prime Visa. No annual fee.



Intelligent AI Recognition
Learn more about bird habits and help keep squirrels away
Cardinal

Daulpell Smart Bird Feeder with Cam Bird Feeder with 1080P HD Camera, 160°Wide Angle, Auto Capture Bird V & Motion Detection AI Identify Bird S

Brand: Daulpell

$159.99

prime Two-Day

FREE Returns

Coupon: □ Apply $10 coupon Shop items › | Terms

Get a $150 Gift Card: Pay $9.99 ~~$159.99~~ upon approval for Pri annual fee.

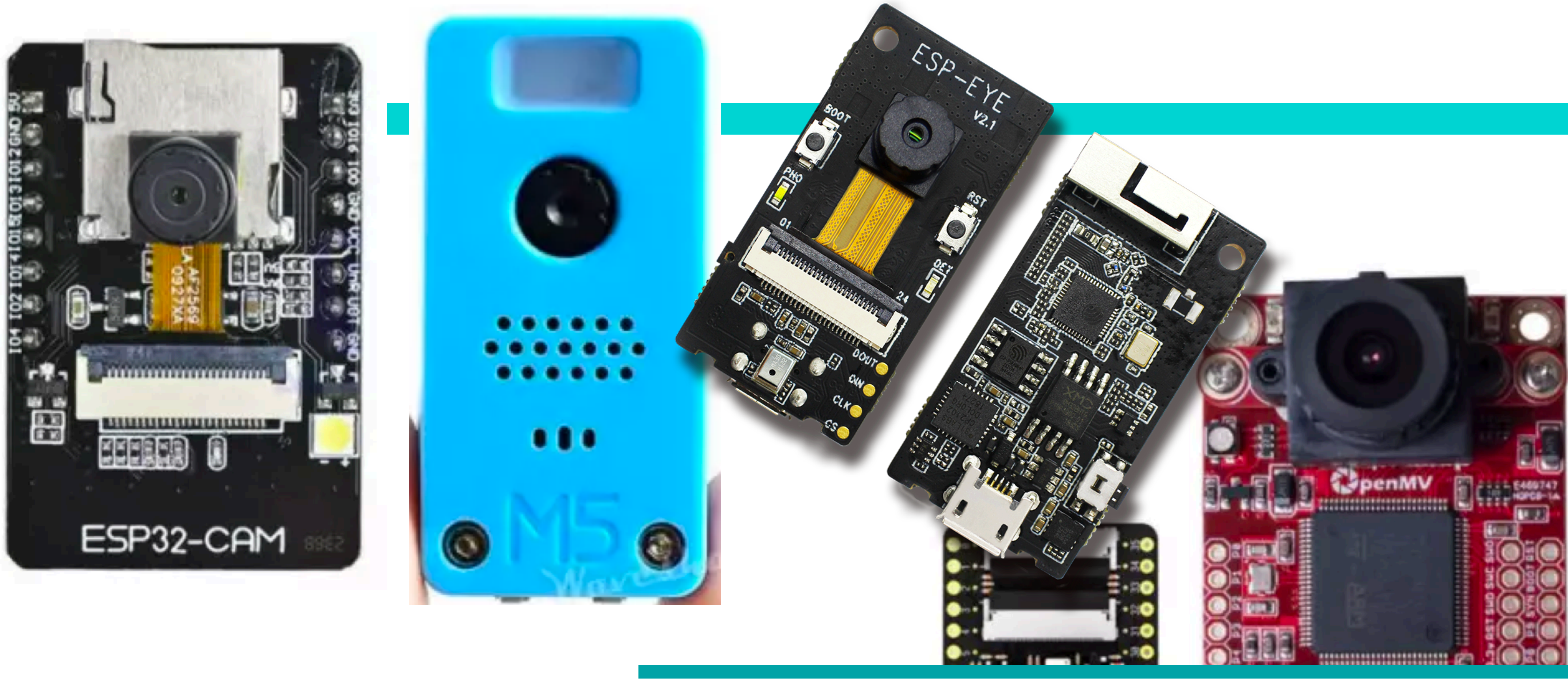
**Purchase options and add-ons**

**Payment plans**

$15.60/mo (12 mo) at example APR of 30% (rates from 10-30% APR)

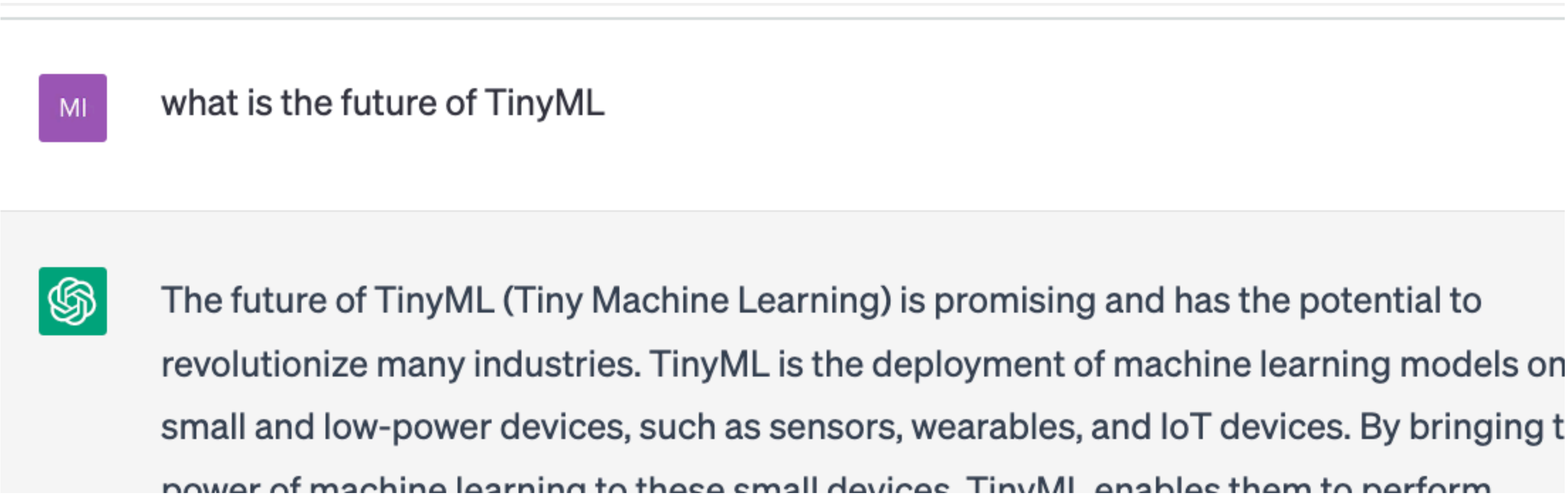
| | |
|---|---|
| **Brand** | Daulpell |
| **Connectivity** | Wireless |

# DIY FLEXIBILITY



```
/* Edge Impulse Arduino examples
 * Copyright (c) 2022 EdgeImpulse Inc.
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */
```

# FUTURE OF TINYML



what is the future of TinyML

The future of TinyML (Tiny Machine Learning) is promising and has the potential to revolutionize many industries. TinyML is the deployment of machine learning models on small and low-power devices, such as sensors, wearables, and IoT devices. By bringing t power of machine learning to these small devices, TinyML enables them to perform

# FURTHER LEARNING

Tensorflow Lite for Microcontrollers: https://www.tensorflow.org/lite/microcontrollers

MIT Sloan School of Business ML Overview: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Hymel et al., “Edge Impulse: An MLOps Platform for Tiny Machine Learning” 2023, https://arxiv.org/abs/2212.03332

IBM, “What are convolutional neural networks?” https://www.ibm.com/topics/convolutional-neural-networks

# Q & A

# DEMO