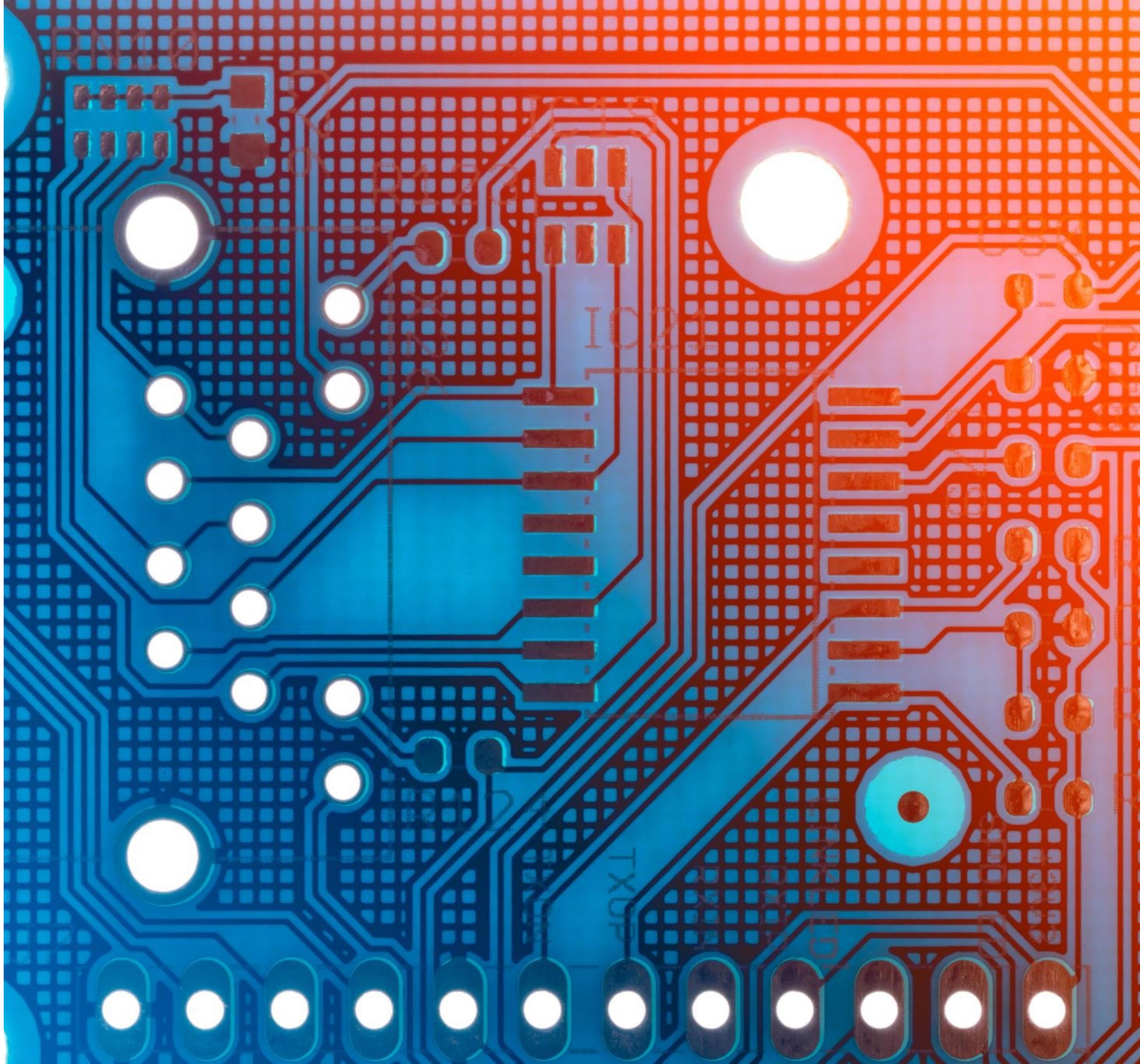
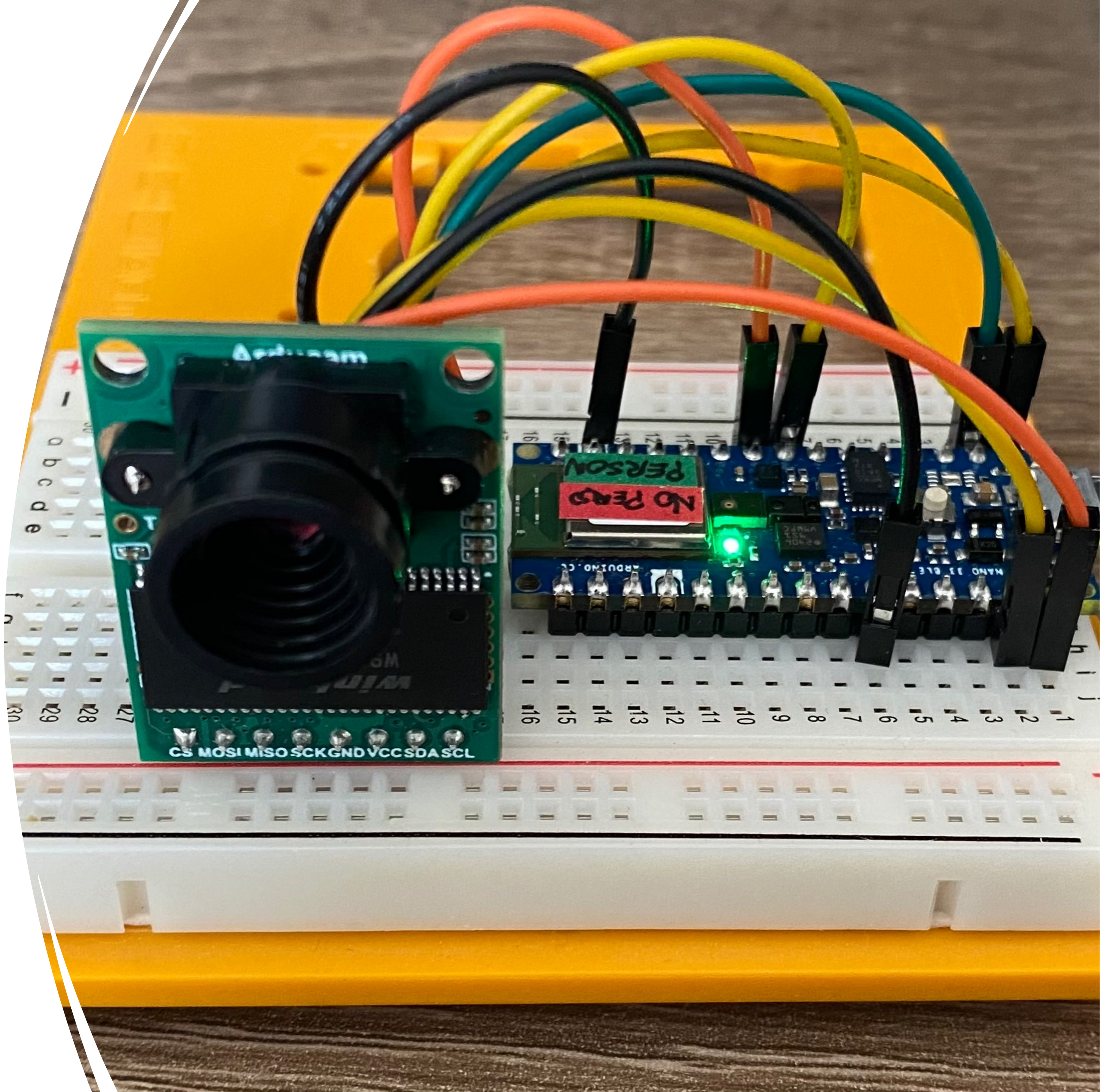
Embedded Machine Learning for Person Detection

Presented by: Ana Ferraz



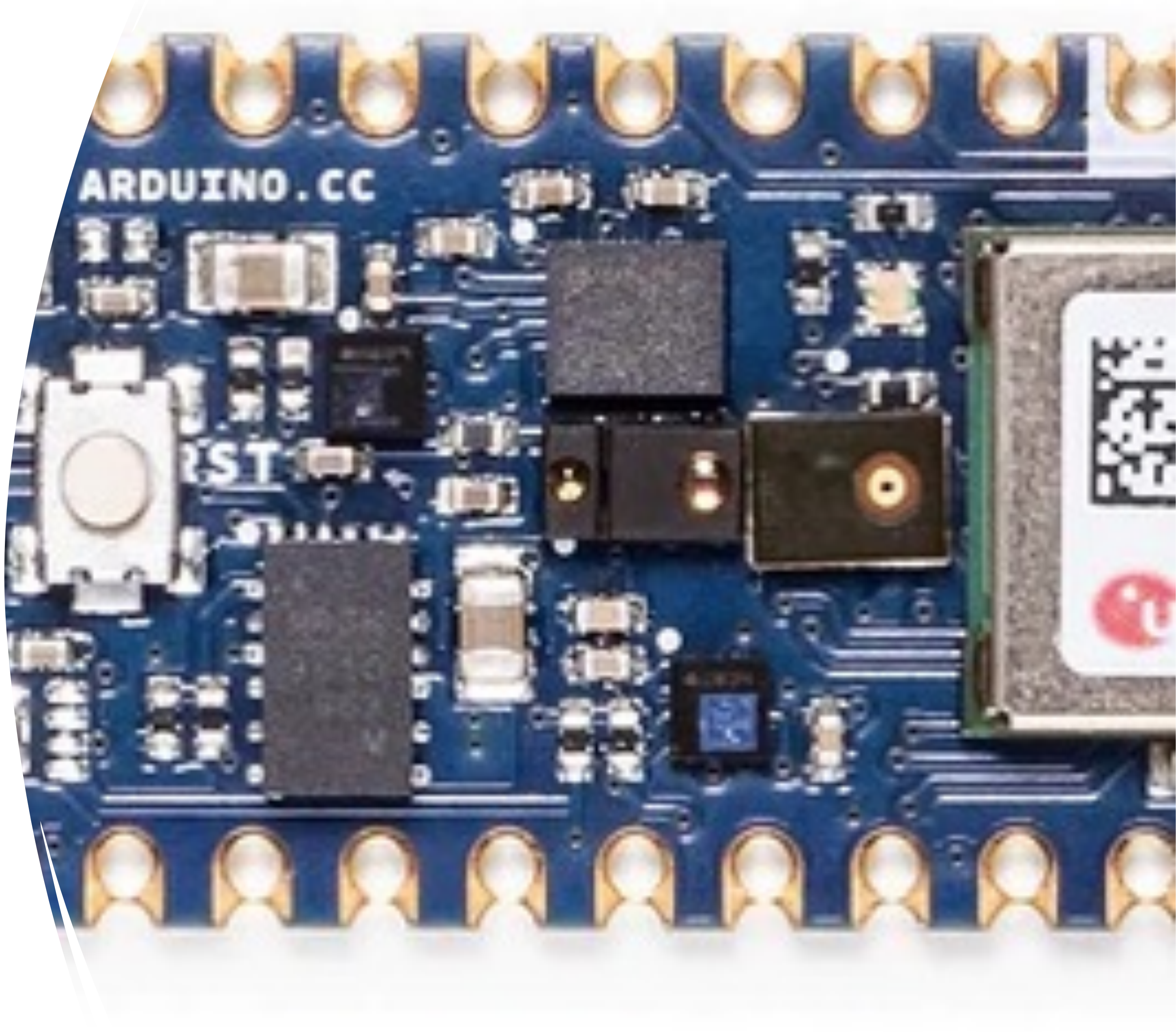
Contents

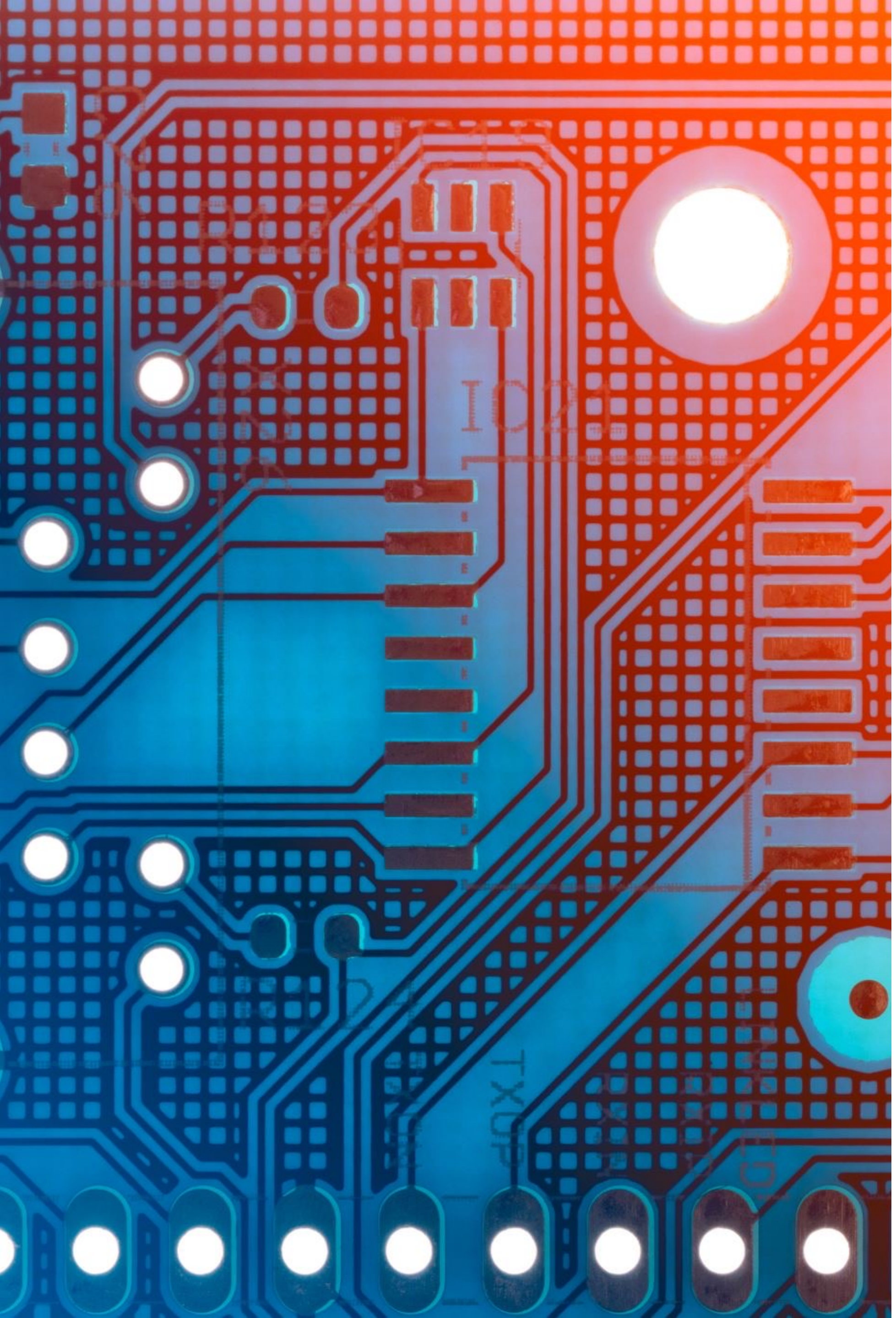
- Introduction
- Workflow
- Hardware used
- Visual Wake Words Dataset
- Application Architecture
- Person Detection
- Training a Model
- Conclusion and future work



Introduction

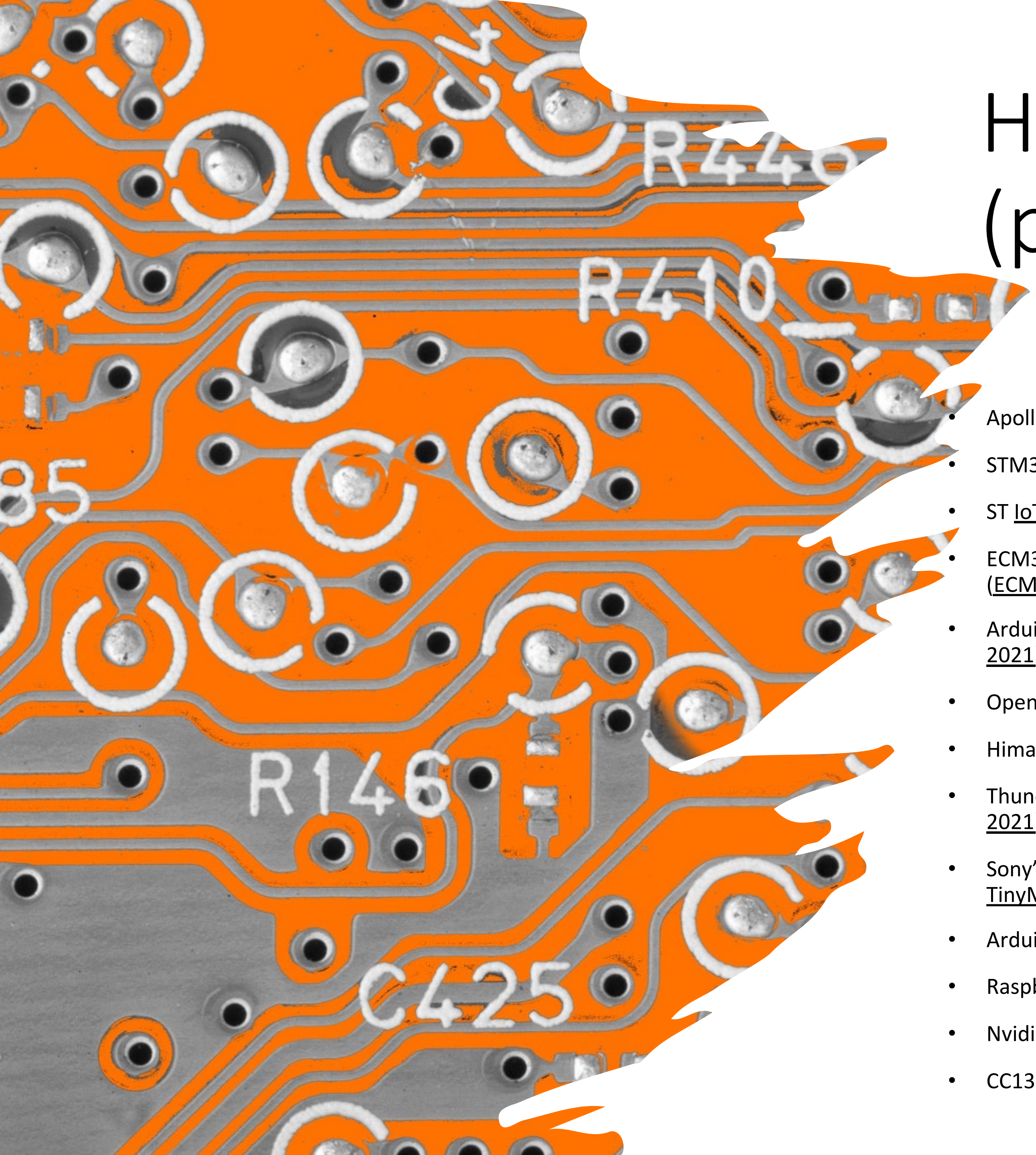
- TinyML – Review previous seminar
- ML at the embedded edge devices
- Embedded devices have serious constraints
- Various sensors built-in or connected
- Recent - new field





Deep Learning Workflow

- Decide the goal
- Collect a Dataset
- Design a Model Architecture
- Train the model
- Convert the Model
- Run Inference
- Evaluate and Troubleshoot

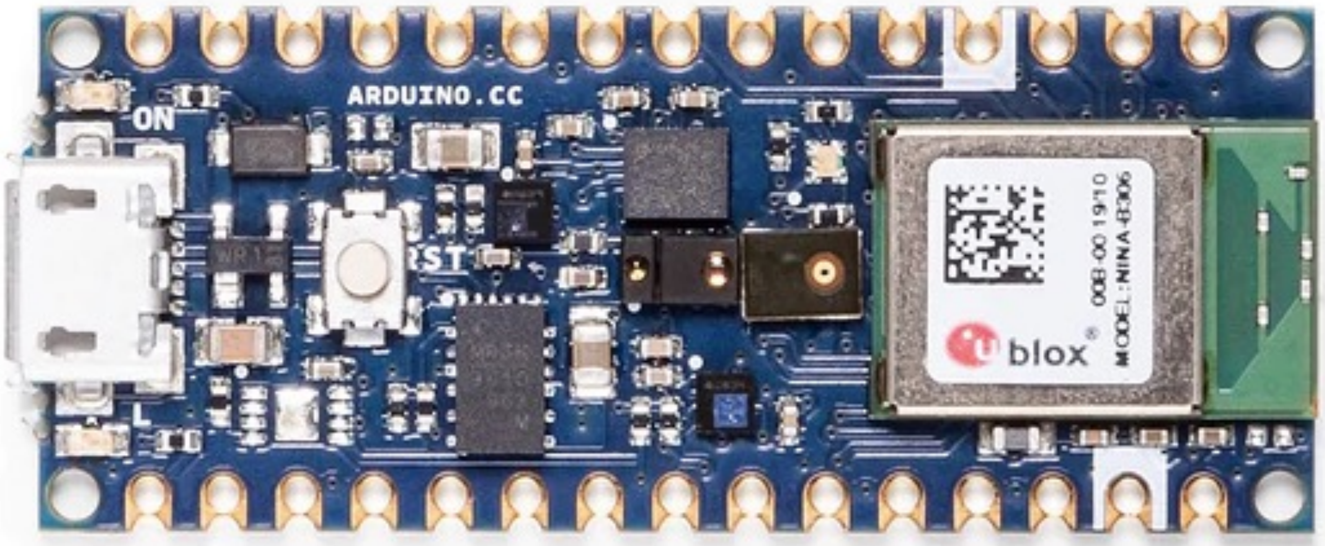


Hardware (previous presentation)

- Apollo3 (Apollo3, 2021),
- STM32F Discovery (STM32F, 2021),
- ST IoT Discovery (ST IoT Discovery, 2021),
- ECM3532 AI Sensor Neuro sensor processor (NSP) (ECM3532, 2021),
- Arduino Nano 33 BLE Sense (Arduino Nano 33, 2021),
- OpenMV Cam H7 Plus (OpenMV, 2021),
- Himax EW-I Plus (Himax, 2021),
- Thunderboard Sense 2 (Thunderboard Sense 2, 2021),
- Sony's Spresense TinyML Board (Sony's Spresense TinyML Board, 2021),
- Arduino Portenta H7 (Arduino Portenta H7, 2021),
- Raspberry Pi 4B (Raspberry Pi 4B, 2021),
- Nvidia Jetson Nano (Nvidia Jetson Nano, 2021),
- CC1352P Launchpad (CC1352P Launchpad, 2021),
- ESP-EYE (ESP-EYE, 2021),
- GAP8 (GAP8, 2021),
- GAP9 (GAP9, 2021),
- AI-deck 1.1 (AI-deck 1.1, 2021),
- Seeed Wio Terminal (Seeed Wio Terminal, 2021),
- Agora Product Development Kit (Agora Product Development Kit, 2021),
- Pico4ML BLE (Pico4ML BLE, 2021),
- MKR Video 4000 (MKR Video 4000, 2021),
- Nicla Sense ME (Nicla Sense ME, 2021),
- Nordic Semi nRF52840 DK (Nordic Semi nRF52840 DK, 2021),
- Nordic Semi Thingy:91 (Nordic Semi Thingy:91, 2021),
- XCore.ai (XCore.ai, 2021),
- FRDM-K64F (FRDM-K64F, 2021).

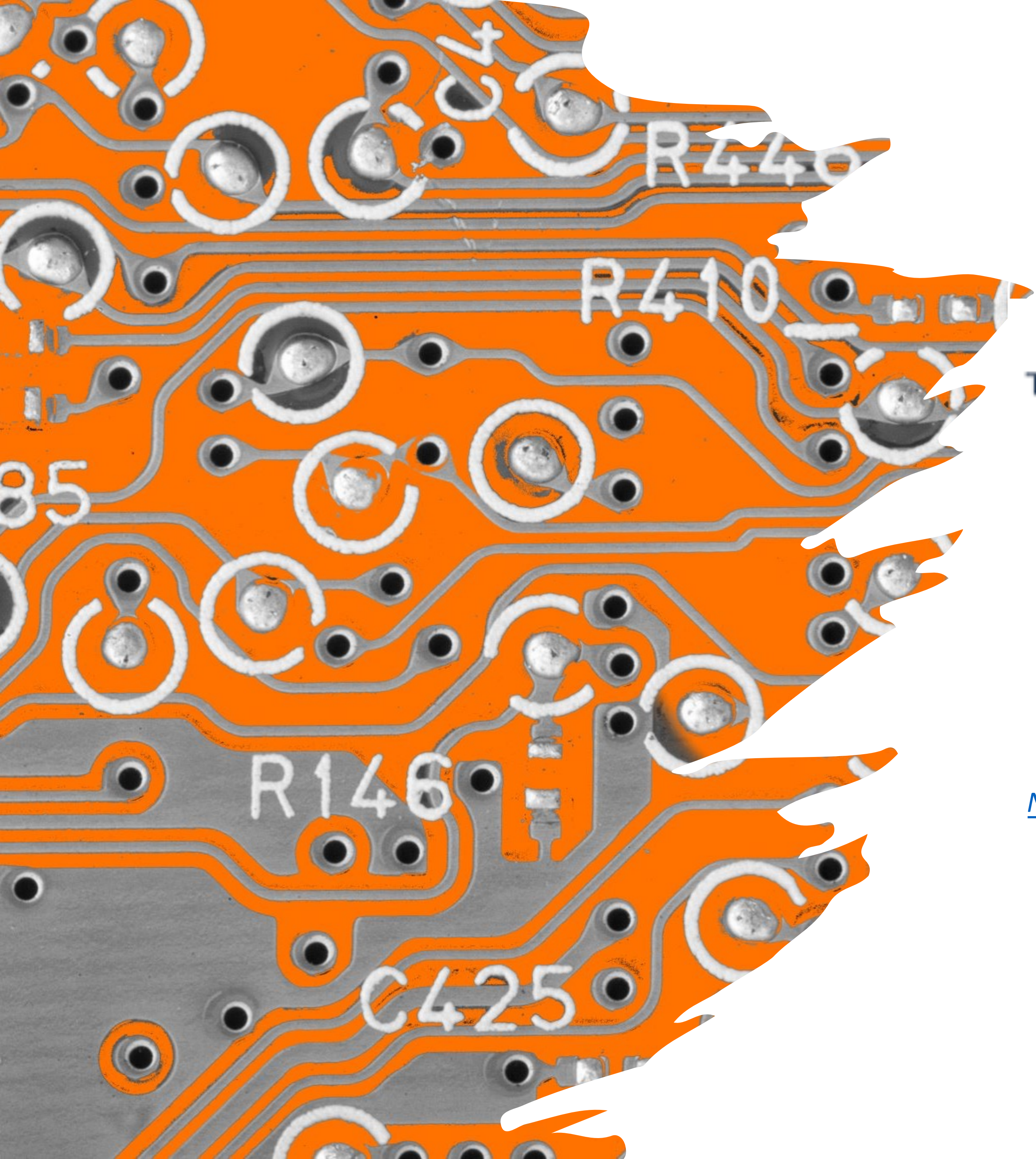
Hardware Board

- Arduino Nano 33 BLE Sense
  - 9 axis inertial sensor
  - Humidity and Temperature
  - Barometric
  - Microphone
  - Gesture
  - Proximity, light color, intensity
- 32-bit ARM® Cortex®-M4 CPU
- 64MHz
- 1MB program memory
- SRAM 256KB



Hardware Camera

- Can be used in Arduino, Raspberry Pi, etc.
- 2 megapixels image
- SPI interface for the sensor configuration
- Output format: FAW, YUV, RGB, JPEG



Software and Libraries (previous presentation)

TensorFlow Lite (TFL)

uTensor

Edge Impulse

NanoEdge AI Studio

PyTorch Mobile

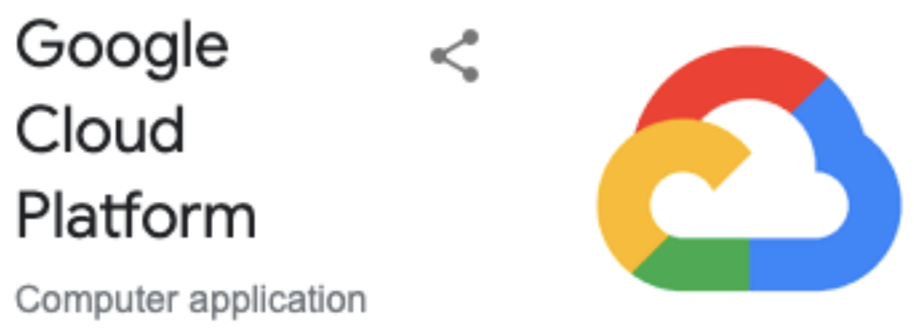
Embedded Learning Library (ELL)

STM32Cube.AI

µTVM: MicroTVM

Software Used

- Google Cloud Platform
- TensorFlow Lite – Training and Conversion
- Arduino – upload to hardware



Google Cloud Platform

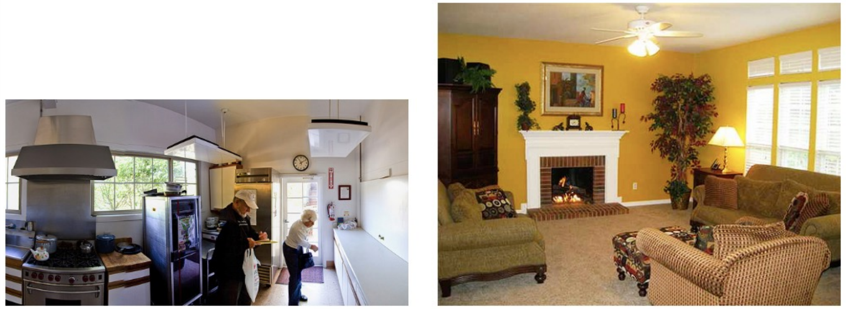
Computer application

TensorFlow Lite

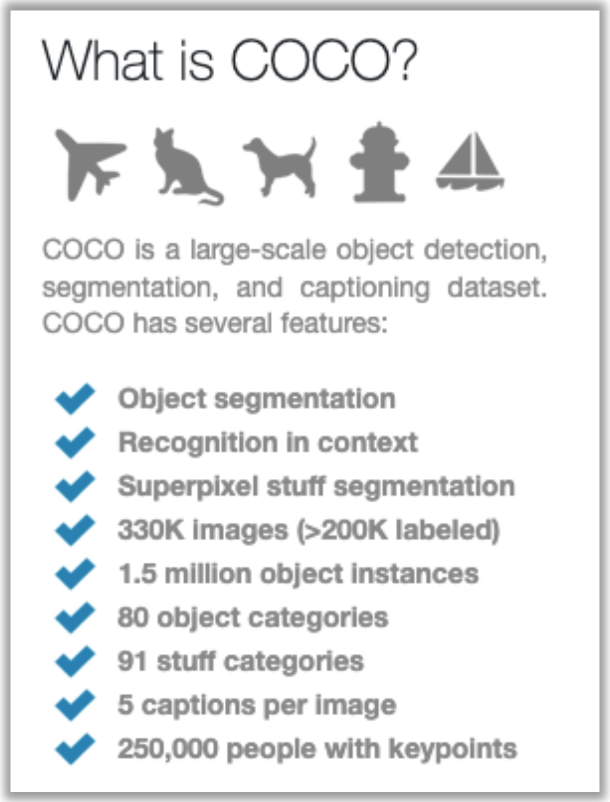
ARDUINO

Dataset Used

- Visual Wake Words [3]
- Re-labeling COCO dataset
  - Label 1 – has at least one object bounding box
  - Label 2 – doesn’t have the object bounding box
- Small bounding boxes (<0.5%) excluded



(a) ‘Person’ (b) ‘Not-person’



What is COCO?

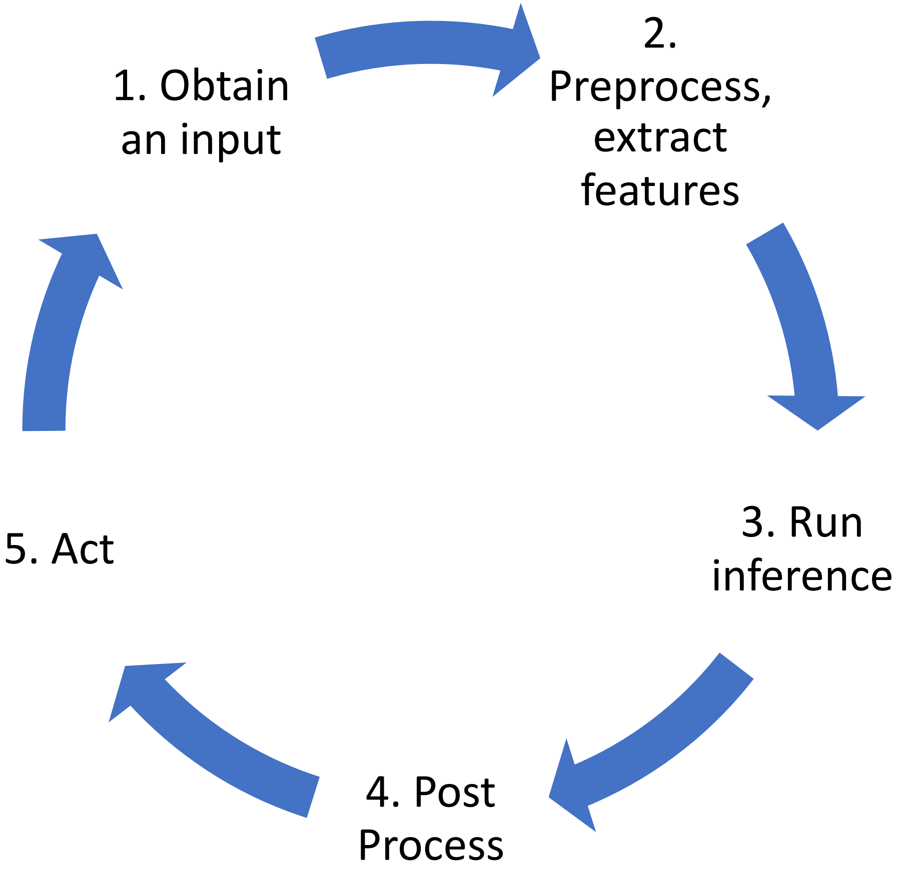
COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

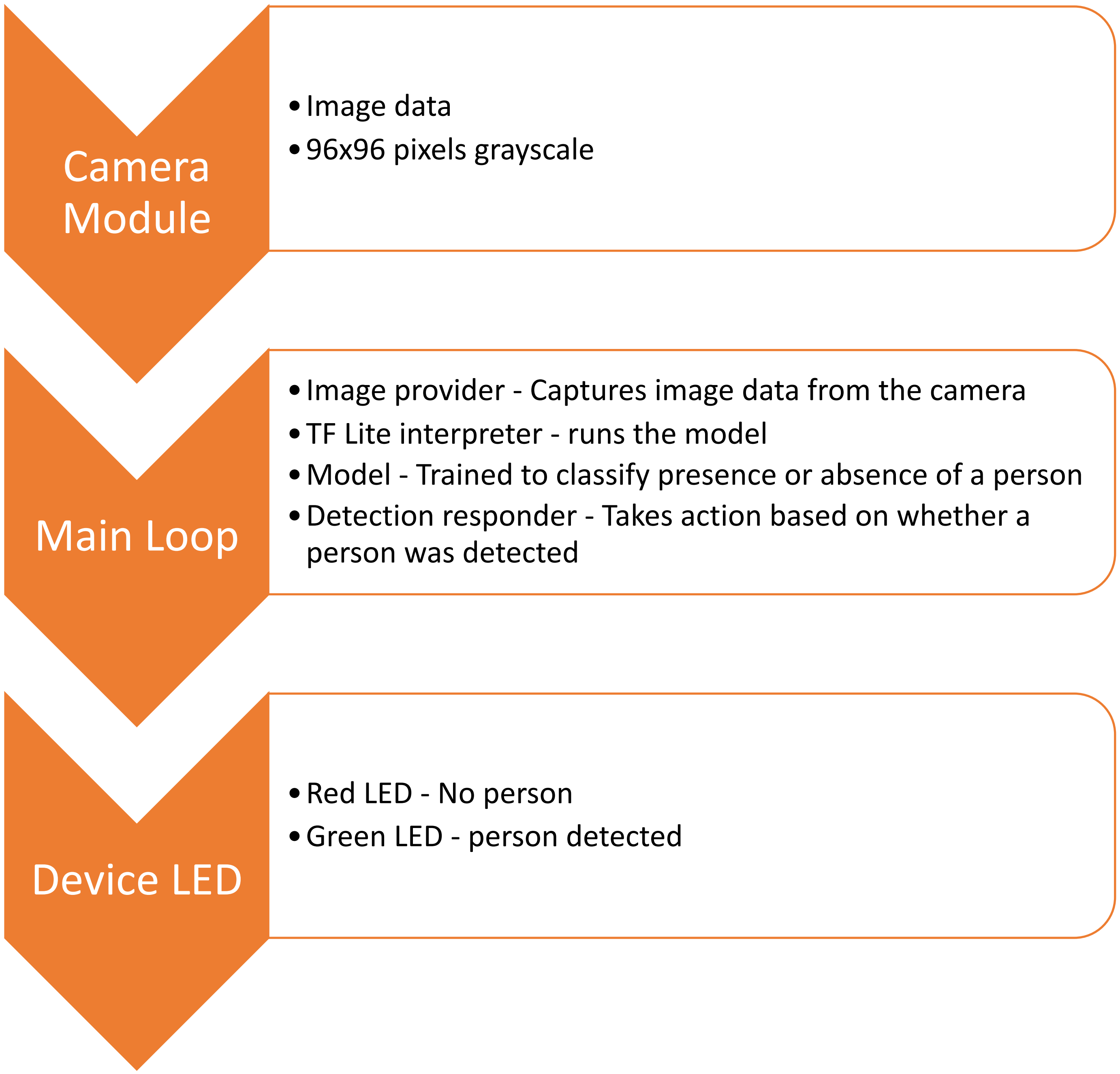
[4] https://cocodataset.org/#home

Application Architecture

- Obtain an input
- Preprocess the input to extract features
- Run inference
- Post process the model’s output
- Use resulting information to act



Structure Person Detection Application



Camera Module

- Image data
- 96x96 pixels grayscale

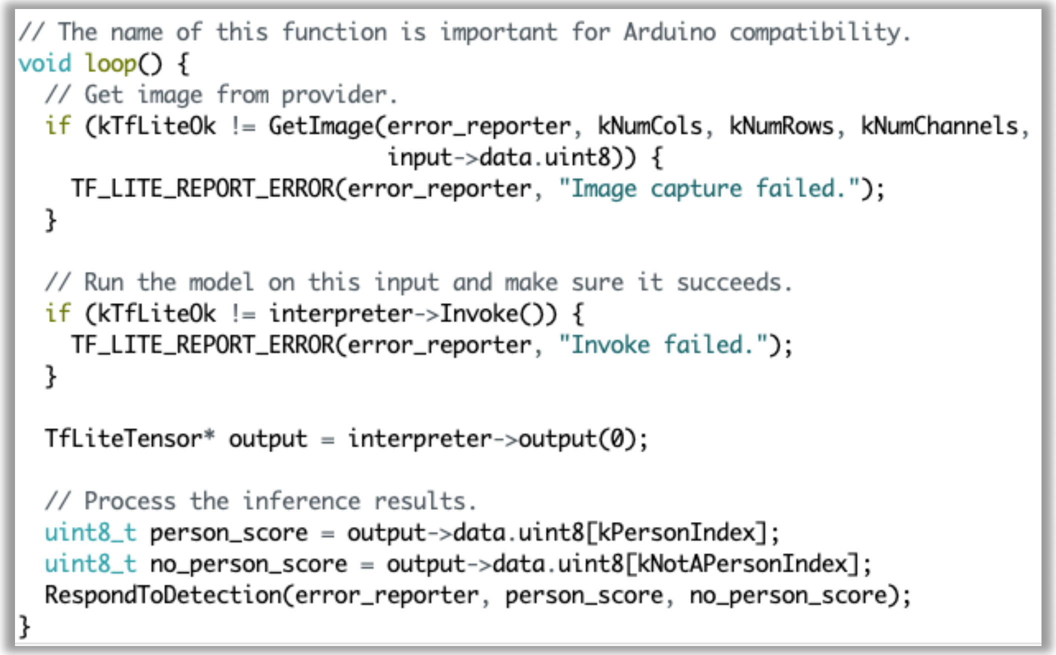
Main Loop

- Image provider - Captures image data from the camera
- TF Lite interpreter - runs the model
- Model - Trained to classify presence or absence of a person
- Detection responder - Takes action based on whether a person was detected

Device LED

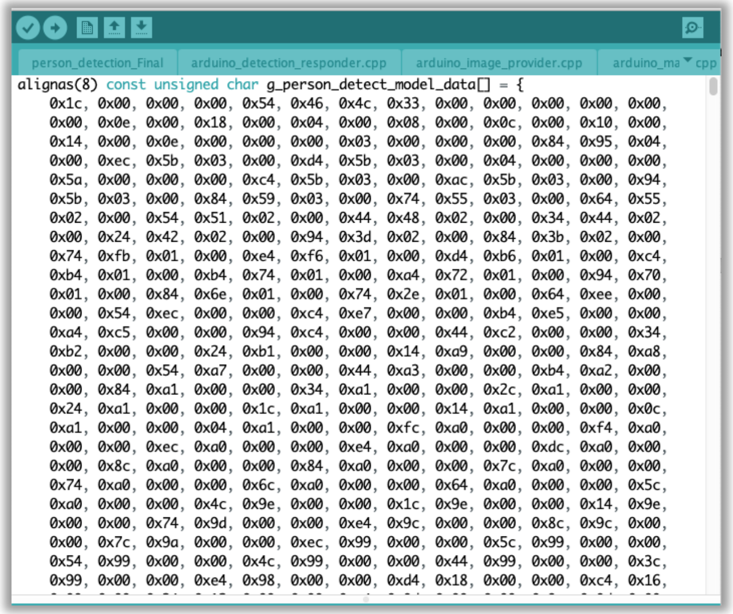
- Red LED - No person
- Green LED - person detected

Main routines



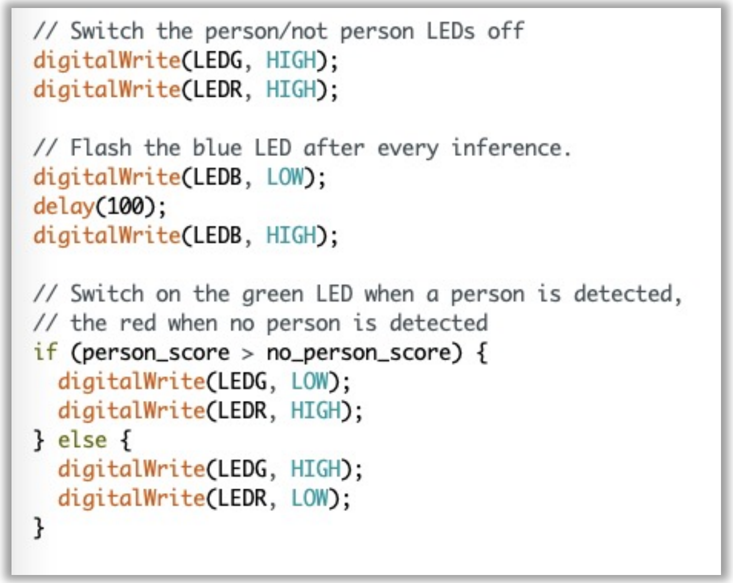
```
// The name of this function is important for Arduino compatibility.
void loop() {
  // Get image from provider.
  if (kTfLiteOk != GetImage(error_reporter, kNumCols, kNumRows, kNumChannels,
                            input->data.uint8)) {
    TF_LITE_REPORT_ERROR(error_reporter, "Image capture failed.");
  }

  // Run the model on this input and make sure it succeeds.
  if (kTfLiteOk != interpreter->Invoke()) {
    TF_LITE_REPORT_ERROR(error_reporter, "Invoke failed.");
  }

  TfLiteTensor* output = interpreter->output(0);

  // Process the inference results.
  uint8_t person_score = output->data.uint8[kPersonIndex];
  uint8_t no_person_score = output->data.uint8[kNotAPersonIndex];
  RespondToDetection(error_reporter, person_score, no_person_score);
}
```

Main Loop



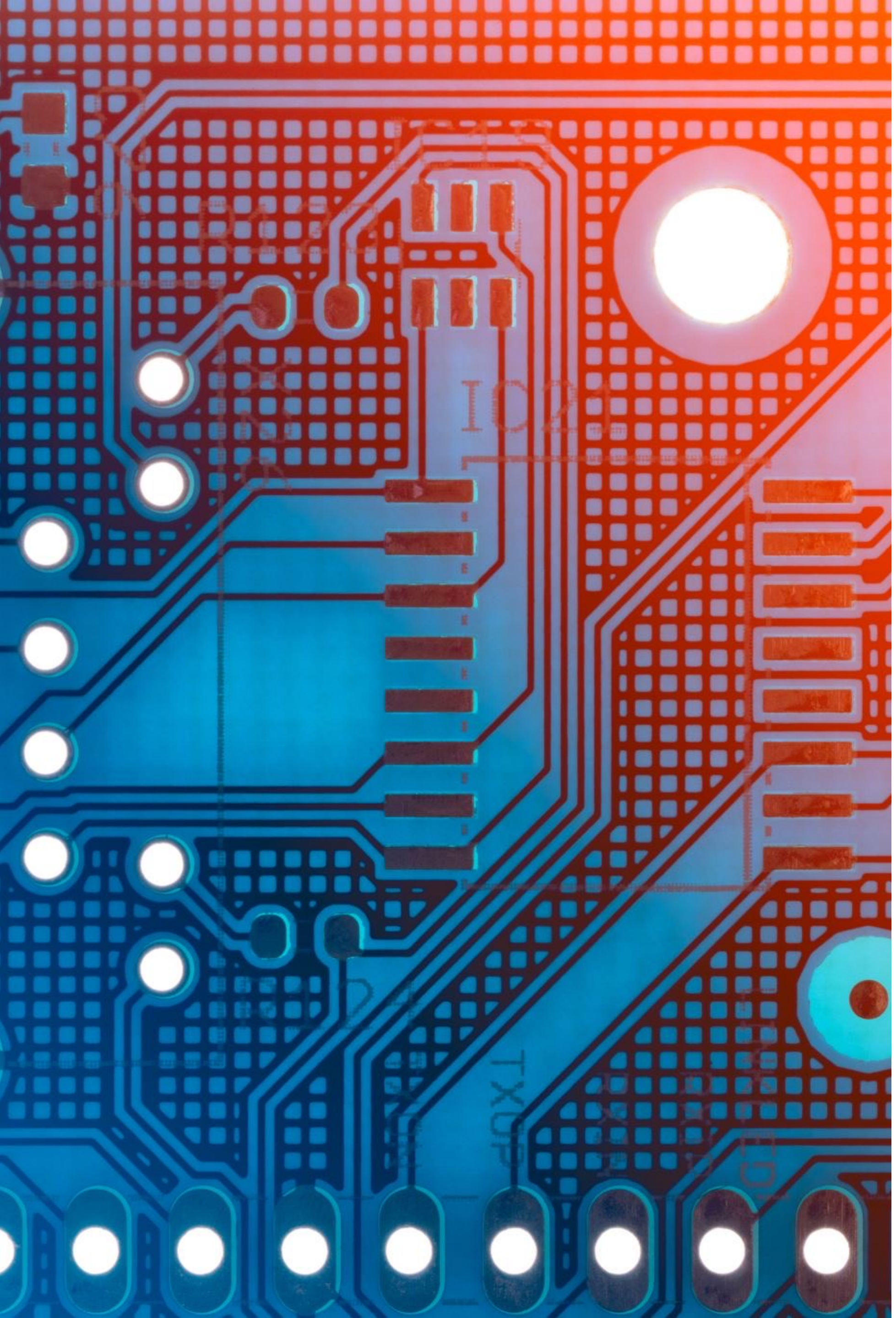
```
alignas(8) const unsigned char g_person_detect_model_data[] = {
    0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x0e, 0x00, 0x18, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00, 0x10, 0x00,
    0x14, 0x00, 0x0e, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x84, 0x95, 0x04,
    0x00, 0xec, 0x5b, 0x03, 0x00, 0xd4, 0x5b, 0x03, 0x00, 0x04, 0x00, 0x00, 0x00,
    0x5a, 0x00, 0x00, 0x00, 0xc4, 0x5b, 0x03, 0x00, 0xac, 0x5b, 0x03, 0x00, 0x94,
    0x5b, 0x03, 0x00, 0x84, 0x59, 0x03, 0x00, 0x74, 0x55, 0x03, 0x00, 0x64, 0x55,
    0x02, 0x00, 0x54, 0x51, 0x02, 0x00, 0x44, 0x48, 0x02, 0x00, 0x34, 0x44, 0x02,
    0x00, 0x24, 0x42, 0x02, 0x00, 0x94, 0x3d, 0x02, 0x00, 0x84, 0x3b, 0x02, 0x00,
    0x74, 0xfb, 0x01, 0x00, 0xe4, 0xf6, 0x01, 0x00, 0xd4, 0xb6, 0x01, 0x00, 0xc4,
    0xb4, 0x01, 0x00, 0xb4, 0x74, 0x01, 0x00, 0xa4, 0x72, 0x01, 0x00, 0x94, 0x70,
    0x01, 0x00, 0x84, 0x6e, 0x01, 0x00, 0x74, 0x2e, 0x01, 0x00, 0x64, 0xee, 0x00,
    0x00, 0x54, 0xec, 0x00, 0x00, 0xc4, 0xe7, 0x00, 0x00, 0xb4, 0xe5, 0x00, 0x00,
    0xa4, 0xc5, 0x00, 0x00, 0x94, 0xc4, 0x00, 0x00, 0x44, 0xc2, 0x00, 0x00, 0x34,
    0xb2, 0x00, 0x00, 0x24, 0xb1, 0x00, 0x00, 0x14, 0xa9, 0x00, 0x00, 0x84, 0xa8,
    0x00, 0x00, 0x54, 0xa7, 0x00, 0x00, 0x44, 0xa3, 0x00, 0x00, 0xb4, 0xa2, 0x00,
    0x00, 0x84, 0xa1, 0x00, 0x00, 0x34, 0xa1, 0x00, 0x00, 0x2c, 0xa1, 0x00, 0x00,
    0x24, 0xa1, 0x00, 0x00, 0x1c, 0xa1, 0x00, 0x00, 0x14, 0xa1, 0x00, 0x00, 0x0c,
    0xa1, 0x00, 0x00, 0x04, 0xa1, 0x00, 0x00, 0xfc, 0xa0, 0x00, 0x00, 0xf4, 0xa0,
    0x00, 0x00, 0xec, 0xa0, 0x00, 0x00, 0xe4, 0xa0, 0x00, 0x00, 0xdc, 0xa0, 0x00,
    0x00, 0x8c, 0xa0, 0x00, 0x00, 0x84, 0xa0, 0x00, 0x00, 0x7c, 0xa0, 0x00, 0x00,
    0x74, 0xa0, 0x00, 0x00, 0x6c, 0xa0, 0x00, 0x00, 0x64, 0xa0, 0x00, 0x00, 0x5c,
    0xa0, 0x00, 0x00, 0x4c, 0x9e, 0x00, 0x00, 0x1c, 0x9e, 0x00, 0x00, 0x14, 0x9e,
    0x00, 0x00, 0x74, 0x9d, 0x00, 0x00, 0xe4, 0x9c, 0x00, 0x00, 0x8c, 0x9c, 0x00,
    0x00, 0x7c, 0x9a, 0x00, 0x00, 0xec, 0x99, 0x00, 0x00, 0x5c, 0x99, 0x00, 0x00,
    0x54, 0x99, 0x00, 0x00, 0x4c, 0x99, 0x00, 0x00, 0x44, 0x99, 0x00, 0x00, 0x3c,
    0x99, 0x00, 0x00, 0xe4, 0x98, 0x00, 0x00, 0xd4, 0x18, 0x00, 0x00, 0xc4, 0x16,
```

Model data array

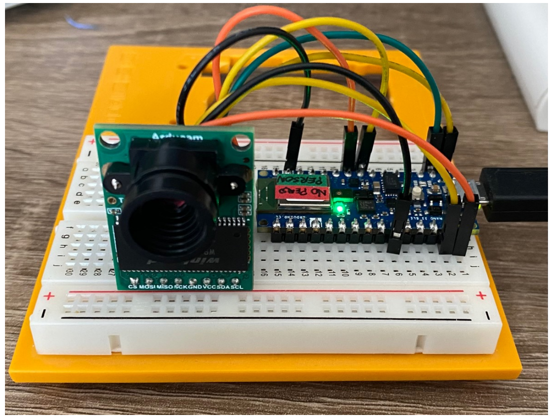


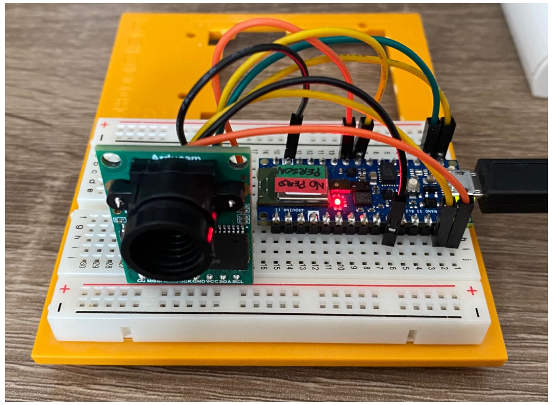
```
// Switch the person/not person LEDs off
digitalWrite(LEDG, HIGH);
digitalWrite(LEDR, HIGH);

// Flash the blue LED after every inference.
digitalWrite(LEDB, LOW);
delay(100);
digitalWrite(LEDB, HIGH);

// Switch on the green LED when a person is detected,
// the red when no person is detected
if (person_score > no_person_score) {
  digitalWrite(LEDG, LOW);
  digitalWrite(LEDR, HIGH);
} else {
  digitalWrite(LEDG, HIGH);
  digitalWrite(LEDR, LOW);
}
```

Detection responder



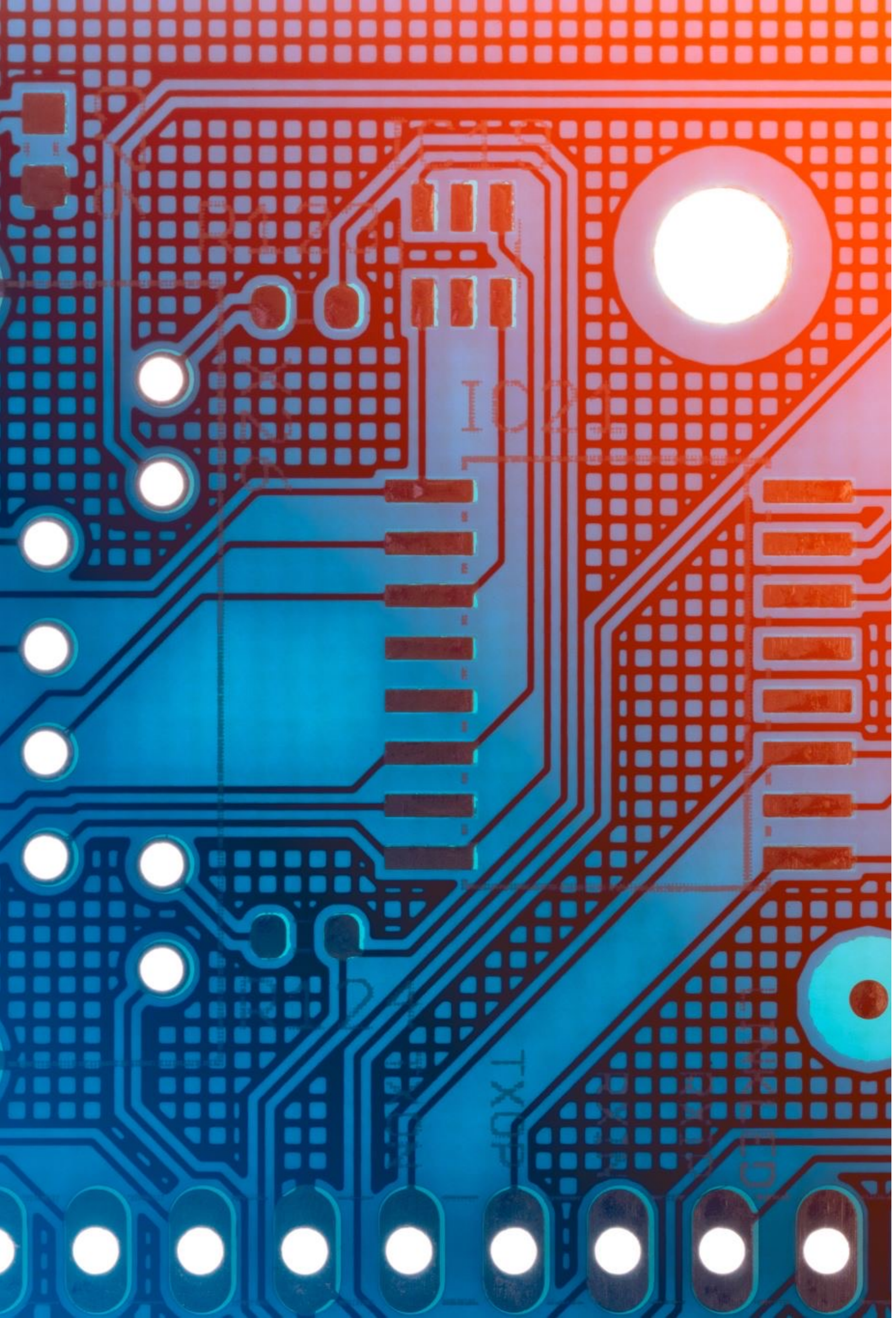
Deploying to Microcontrollers





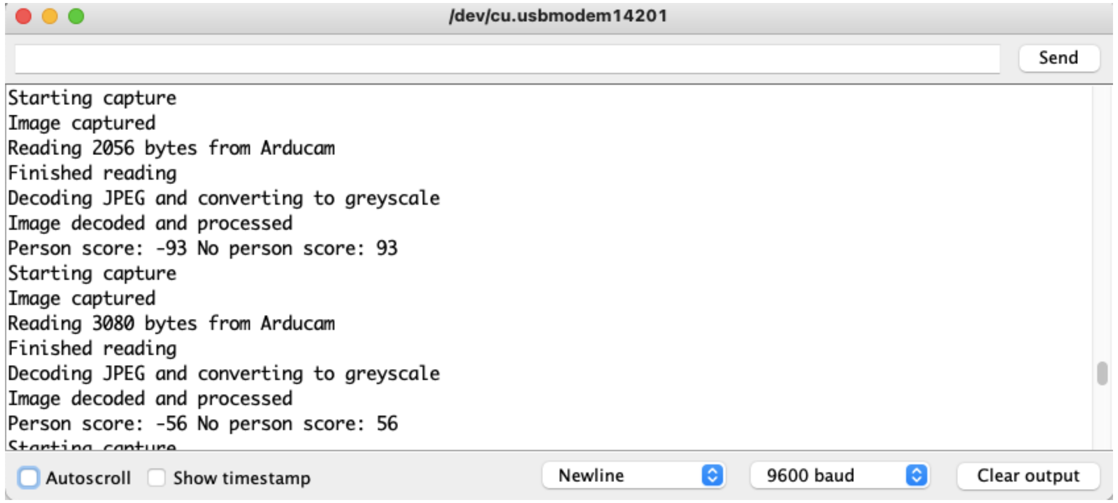
Camera e Board connections

Camera Pin	Arduino Board Pin
CS	D7
MOSI	D11
MISO	D12
SCK	D13
GND	GND
VCC	3.3V
SDA	A4
SCL	A5



Deploying to Microcontrollers

Person Score	No Person Score	Explanation
-82	+82	High confidence in No Person Score
+49	-50	High confidence in Person Score
-28	+28	Slight confidence in Person Score
+28	-28	Slight confidence in No Person Score

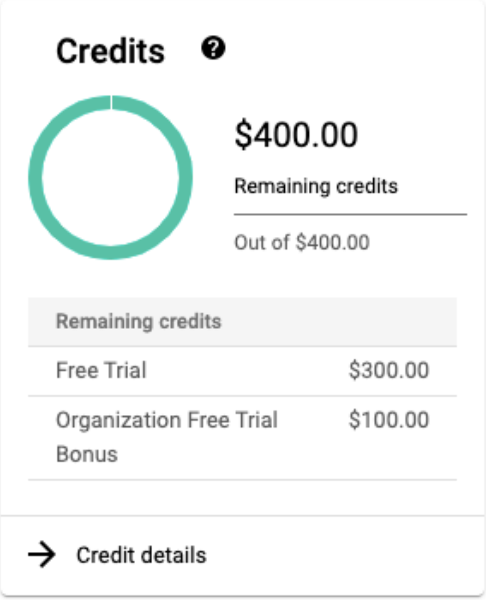


```
Starting capture
Image captured
Reading 2056 bytes from Arducam
Finished reading
Decoding JPEG and converting to greyscale
Image decoded and processed
Person score: -93 No person score: 93
Starting capture
Image captured
Reading 3080 bytes from Arducam
Finished reading
Decoding JPEG and converting to greyscale
Image decoded and processed
Person score: -56 No person score: 56
Starting capture
```

Arduino Project Hub link

Google Cloud Platform

- Picking a Machine
- Google Cloud Platform Instance
- Training the model for other categories



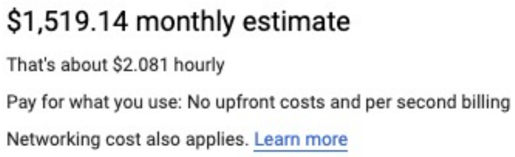
Credits

$400.00

Remaining credits

Out of $400.00

Remaining credits	
Free Trial	$300.00
Organization Free Trial Bonus	$100.00

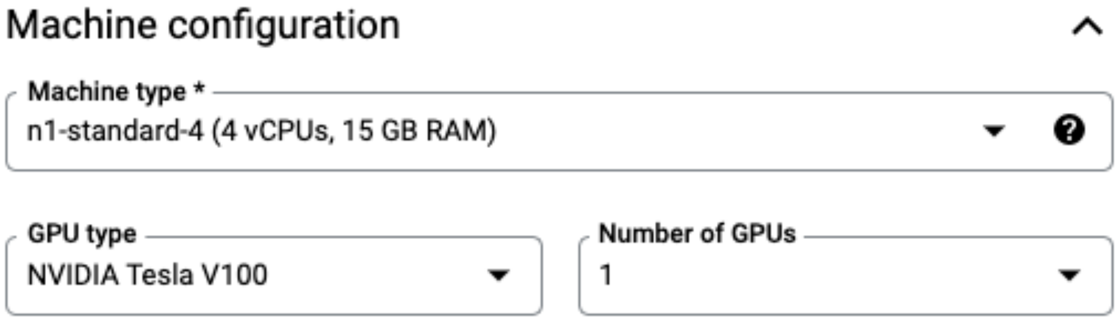


$1,519.14 monthly estimate

That's about $2.081 hourly

Pay for what you use: No upfront costs and per second billing

Networking cost also applies. Learn more



Machine configuration

Machine type *: n1-standard-4 (4 vCPUs, 15 GB RAM)

GPU type: NVIDIA Tesla V100

Number of GPUs: 1

Exporting to TensorFlow Lite

Series of commands:

- Exporting to a GraphDef Protobuf File
- Freezing the Weights
- Quantizing and Converting to TensorFlow Lite
- Converting to a C Source File

Conclusion

- TensorFlow Lite broadens the reach of ML by enabling the transfer of deep learning models into tiny embedded systems.
- The TinyML process of training simplified models in the cloud, converting the files and uploading into the embedded device poses different challenges than traditional ML.
- The hardware/software/libraries compatibility, code compilation, driver updates are also added challenges to TinyML systems.
- Trade accuracy and size of the model

References

[1] P. Ray, "A review on TinyML: State-of-the-art and prospects," Journal of King Saud University, vol. 34, no. 4, pp. 1595-1623, 2022.

[2] P. Warden and D. Situnayake, TinyML - Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly, 2019.

[3] A. Chowdhery, P. Warden, J. Shlens, A. Howard and R. Rhodes, "Visual Wake Words Dataset," Google Research, 2019.

[4] "COCO - Common Objects in Context," [Online]. Available: https://cocodataset.org/#home. [Accessed May 2022].

[5] Arduino Store, "Arduino Nano 33 BLE Sense with headers," [Online]. Available: https://oreil.ly/6qlMD. [Accessed April 2022].

[6] Arducam, "Arducam Mini OV2640 2MP," [Online]. Available: https://oreil.ly/LAwhb. [Accessed April 2022].

[7] "tf-slim," [Online]. Available: https://github.com/google-research/tf-slim. [Accessed May 2022].

[8] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang and P. Warden, "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems," in MLSys Conference, San Jose, 2021.

[9] D. L. Dutta and S. Bharali, "Tiny ML Meets IoT: A Comprehensive Survey," Internet of Things, 2021.