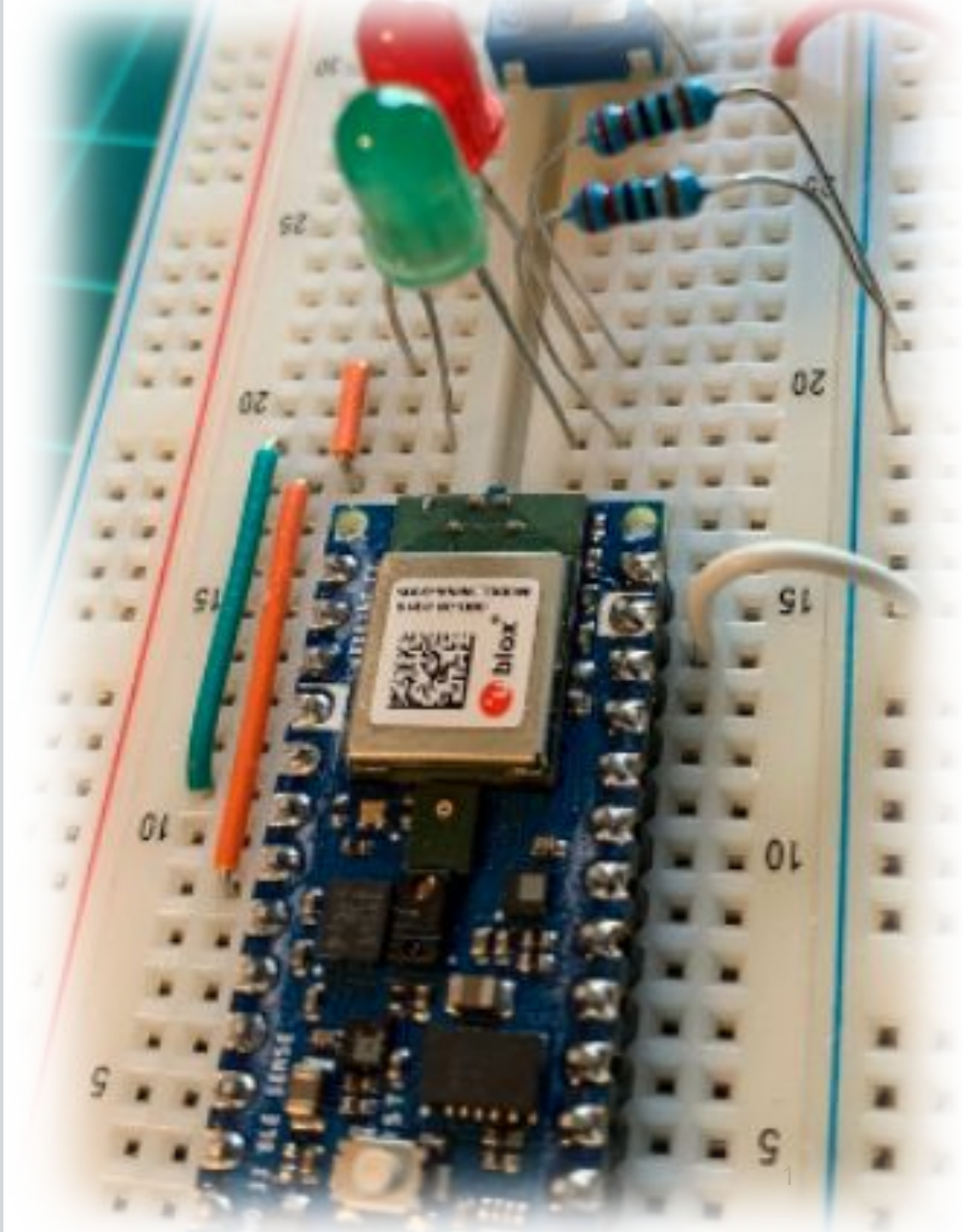
IESTI01 – TinyML

Embedded Machine Learning

27. Image Classification using Edge Impulse Studio

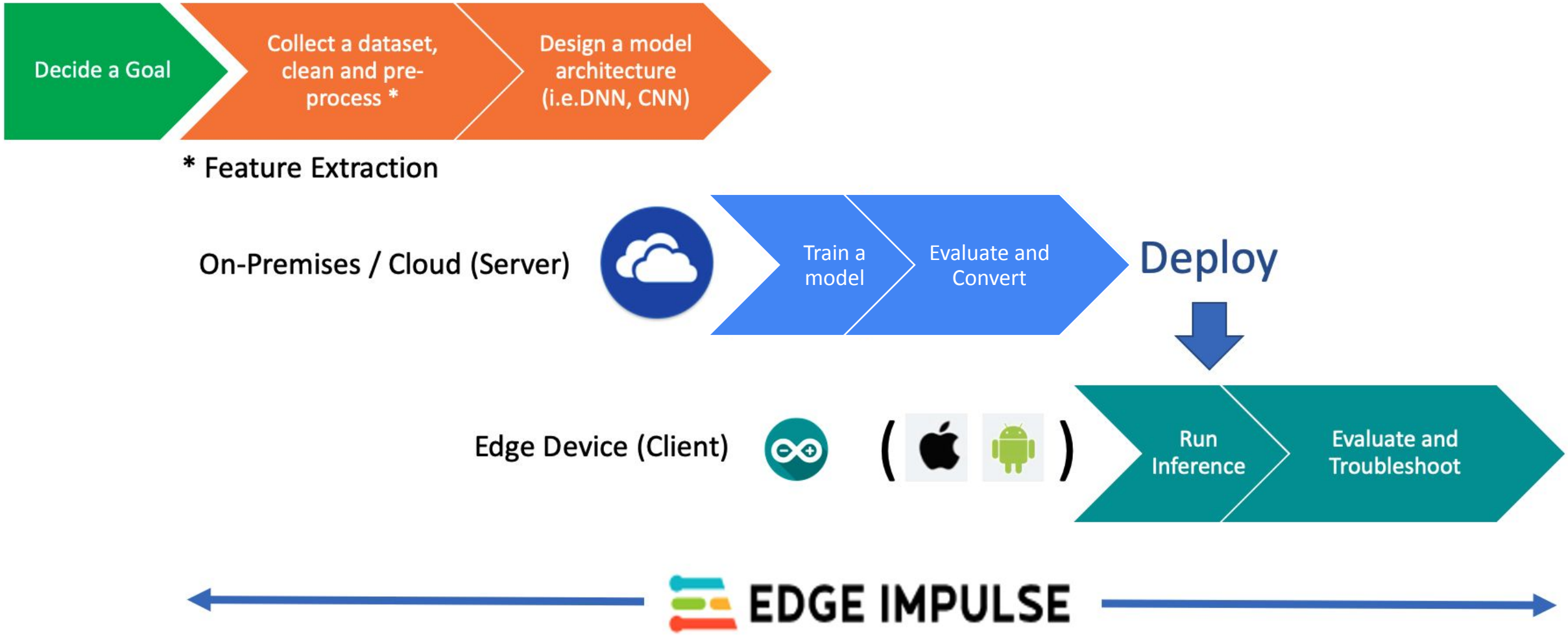


Prof. Marcelo Rovai
UNIFEI

Egg Classification Application: Design, Train, Test and Deploy

https://studio.edgeimpulse.com/public/42476/latest

Thanks to Daniel Situnayake (Edge Impulse)


Decide a Goal
Collect a dataset, clean and pre-process *
Design a model architecture (i.e.DNN, CNN)
* Feature Extraction
On-Premises / Cloud (Server)
Train a model
Evaluate and Convert
Deploy
Edge Device (Client)
Run Inference
Evaluate and Troubleshoot
EDGE IMPULSE

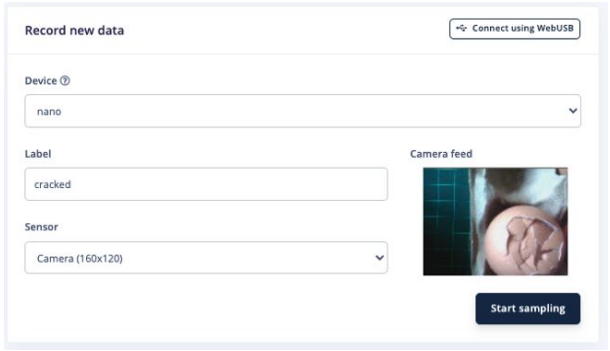
Egg Classification Project

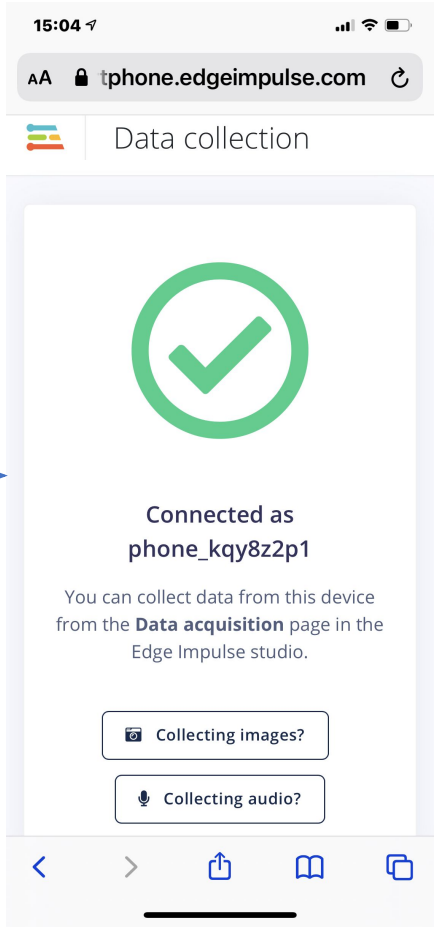
Decide a Goal

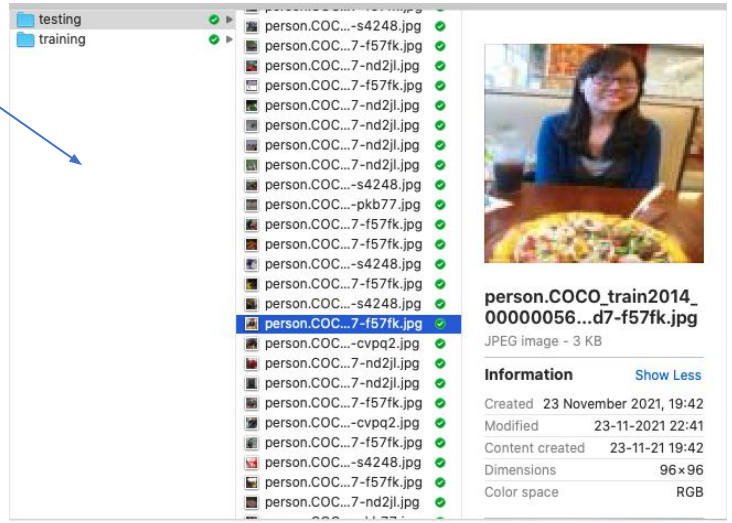
- Possible Egg Images:
  - Intact
  - Cracked
  - Empty (no egg)

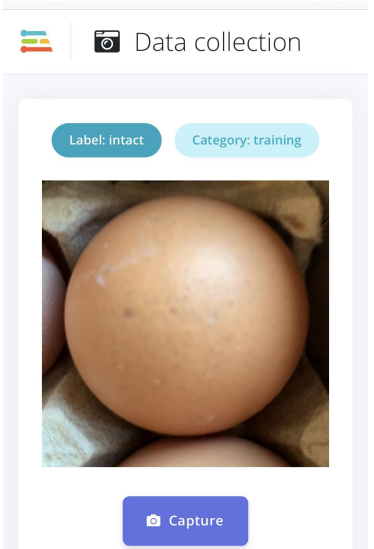
Collect a dataset, clean and pre-process *

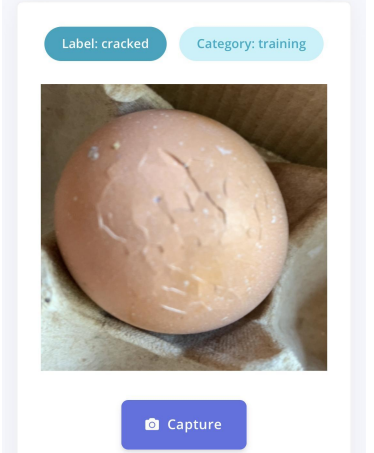
- Data Collection:
  - Arduino (on-line)
  - SmartPhone (on-line)
  - Dataset (upload file)

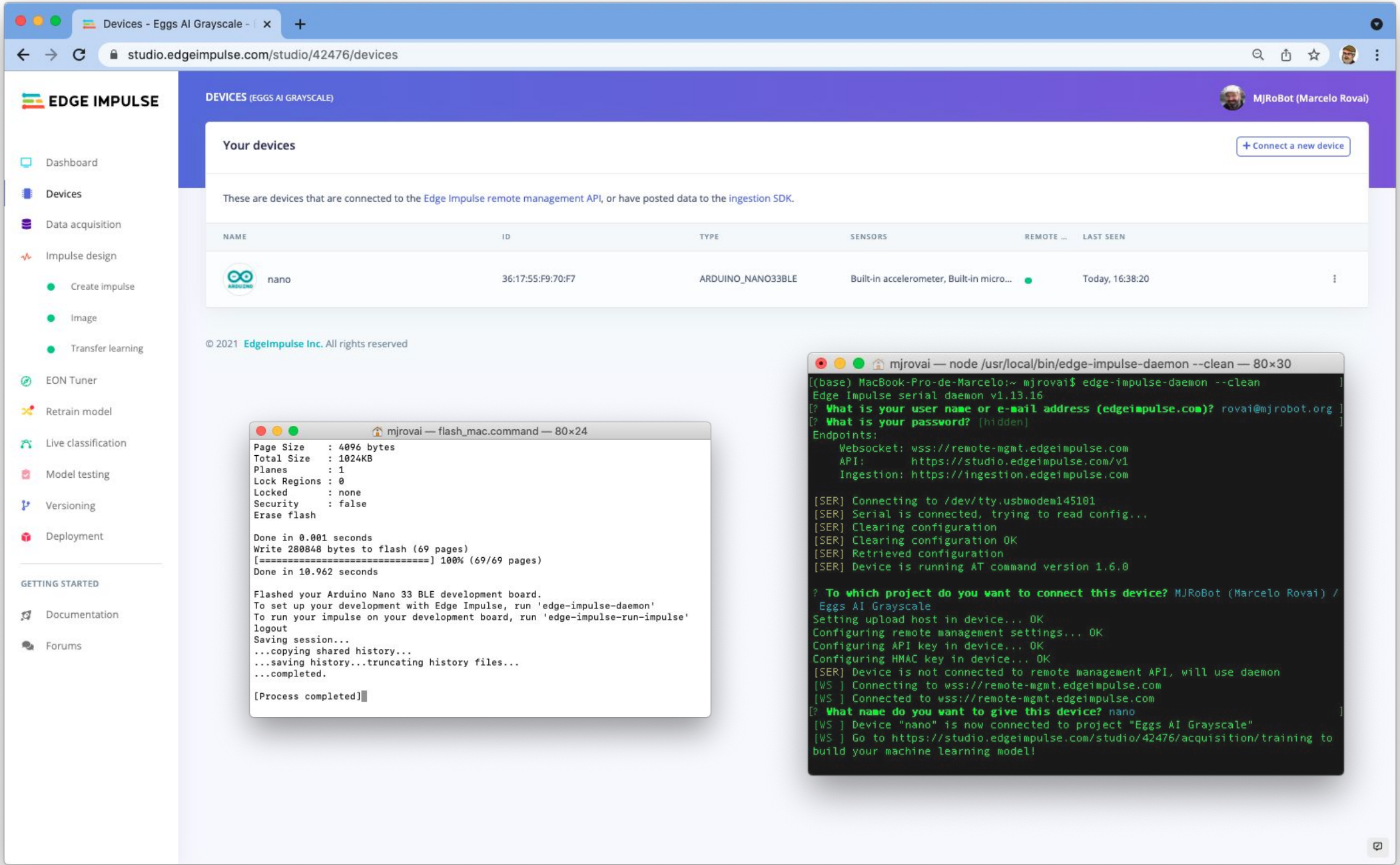


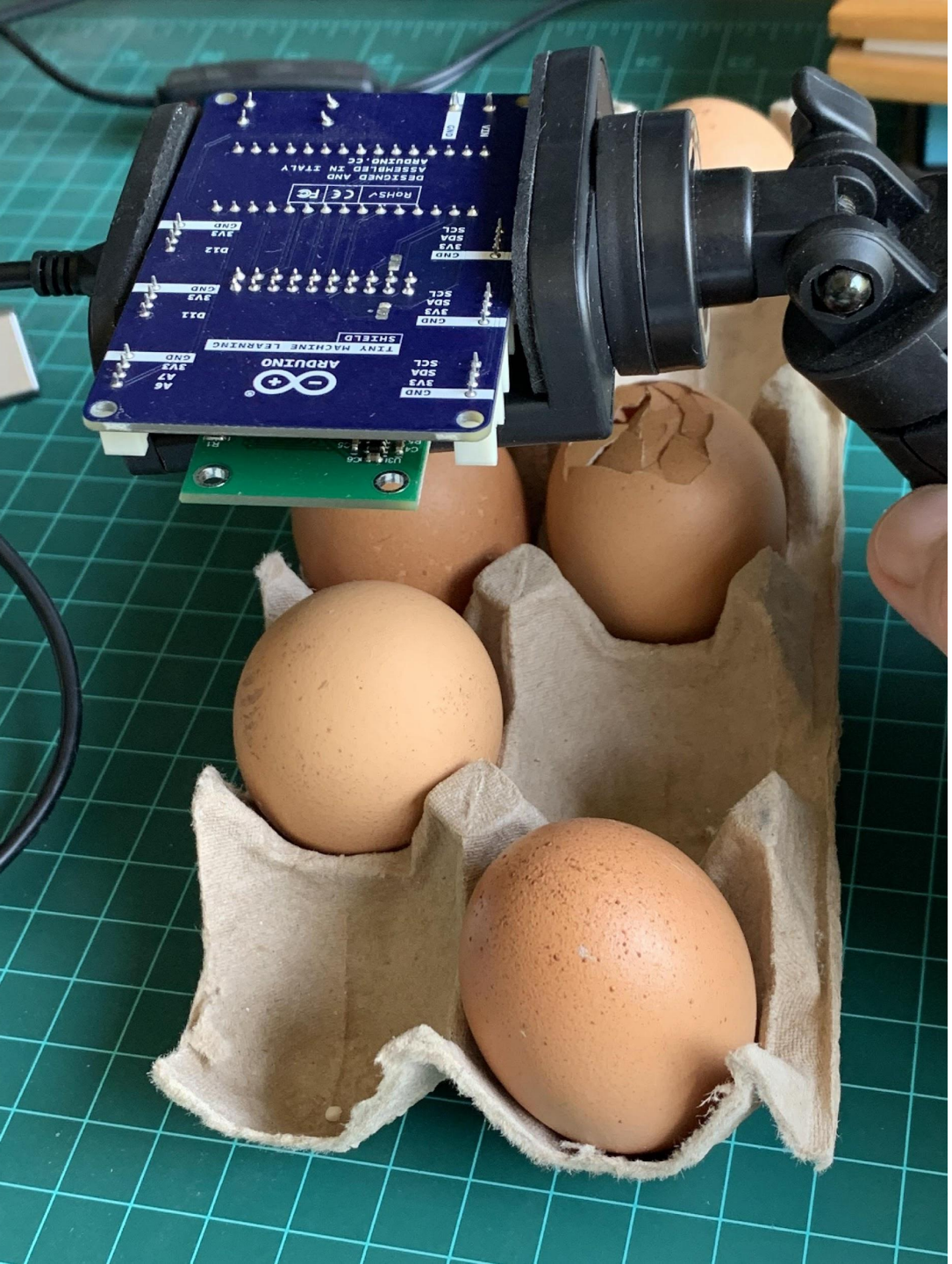









Devices - Eggs AI Grayscale -
studio.edgeimpulse.com/studio/42476/devices
EDGE IMPULSE
DEVICES (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Your devices
+ Connect a new device
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.
NAME
ID
TYPE
SENSORS
REMOTE ...
LAST SEEN
nano
36:17:55:F9:70:F7
ARDUINO_NANO33BLE
Built-in accelerometer, Built-in micro...
Today, 16:38:20
© 2021 EdgeImpulse Inc. All rights reserved
mjrovai — flash_mac.command — 80×24
Page Size : 4096 bytes
Total Size : 1024KB
Planes : 1
Lock Regions : 0
Locked : none
Security : false
Erase flash
Done in 0.001 seconds
Write 280848 bytes to flash (69 pages)
[==============================] 100% (69/69 pages)
Done in 10.962 seconds
Flashed your Arduino Nano 33 BLE development board.
To set up your development with Edge Impulse, run 'edge-impulse-daemon'
To run your impulse on your development board, run 'edge-impulse-run-impulse'
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
[Process completed]
mjrovai — node /usr/local/bin/edge-impulse-daemon --clean — 80×30
[(base) MacBook-Pro-de-Marcelo:~ mjrovai$ edge-impulse-daemon --clean
Edge Impulse serial daemon v1.13.16
[? What is your user name or e-mail address (edgeimpulse.com)? rovai@mjrobot.org]
[? What is your password? [hidden]]
Endpoints:
Websocket: wss://remote-mgmt.edgeimpulse.com
API: https://studio.edgeimpulse.com/v1
Ingestion: https://ingestion.edgeimpulse.com
[SER] Connecting to /dev/tty.usbmodem145101
[SER] Serial is connected, trying to read config...
[SER] Clearing configuration
[SER] Clearing configuration OK
[SER] Retrieved configuration
[SER] Device is running AT command version 1.6.0
? To which project do you want to connect this device? MJRoBot (Marcelo Rovai) / Eggs AI Grayscale
Setting upload host in device... OK
Configuring remote management settings... OK
Configuring API key in device... OK
Configuring HMAC key in device... OK
[SER] Device is not connected to remote management API, will use daemon
[WS] Connecting to wss://remote-mgmt.edgeimpulse.com
[WS] Connected to wss://remote-mgmt.edgeimpulse.com
[? What name do you want to give this device? nano]
[WS] Device "nano" is now connected to project "Eggs AI Grayscale"
[WS] Go to https://studio.edgeimpulse.com/studio/42476/acquisition/training to build your machine learning model!



Record new data

Connect using WebUSB

Device

nano

Label

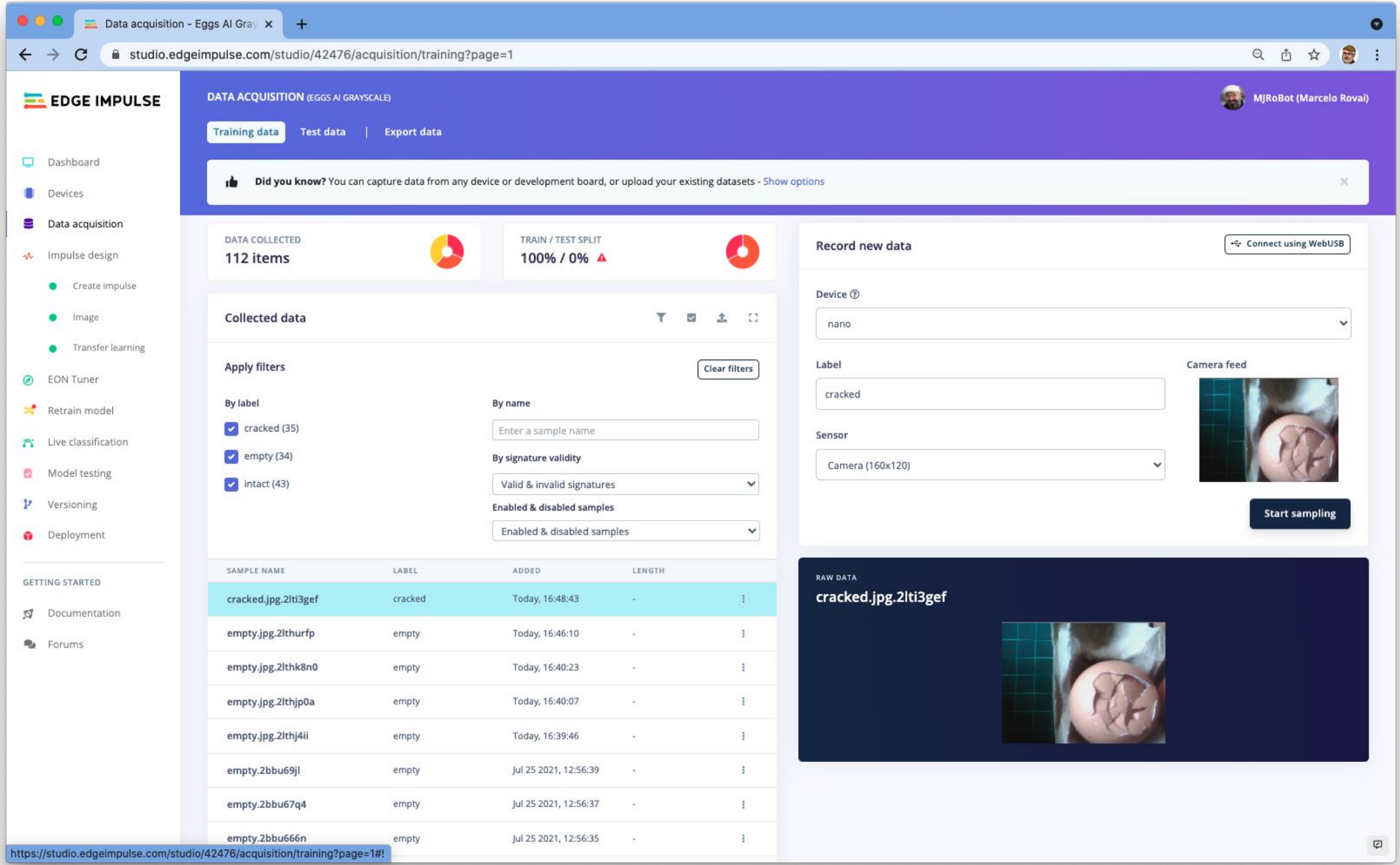
cracked

Sensor

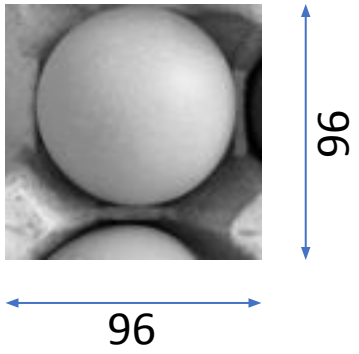
Camera (160x120)

Camera feed

Start sampling

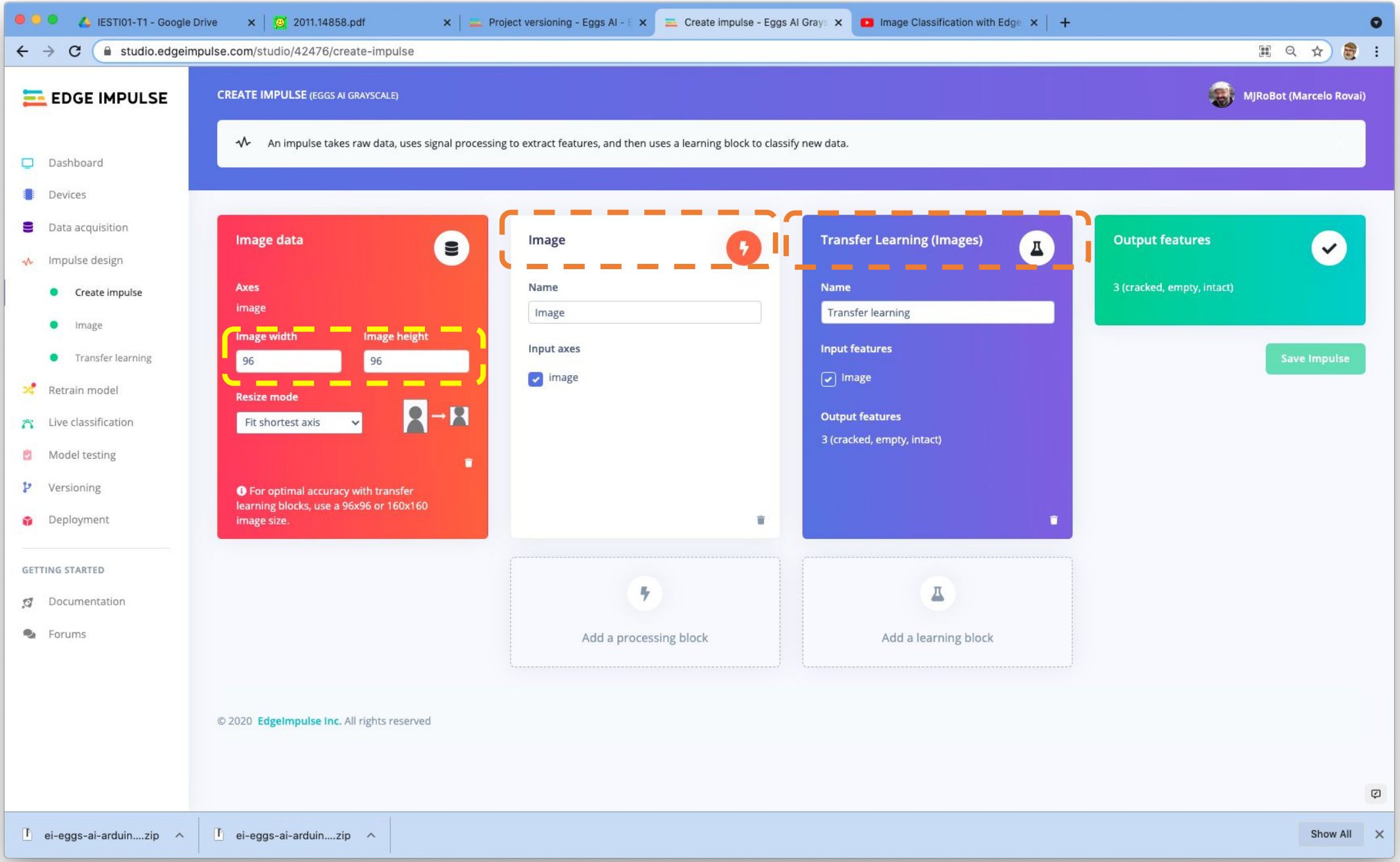

Data acquisition - Eggs AI Gray
studio.edgeimpulse.com/studio/42476/acquisition/training?page=1
EDGE IMPULSE
DATA ACQUISITION (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
Training data
Test data
Export data
Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
DATA COLLECTED
112 items
TRAIN / TEST SPLIT
100% / 0%
Collected data
Apply filters
Clear filters
By label
cracked (35)
empty (34)
intact (43)
By name
Enter a sample name
By signature validity
Valid & invalid signatures
Enabled & disabled samples
Enabled & disabled samples
SAMPLE NAME
LABEL
ADDED
LENGTH
cracked.jpg.2lti3gef cracked Today, 16:48:43 -
empty.jpg.2lthurfp empty Today, 16:46:10 -
empty.jpg.2lthk8n0 empty Today, 16:40:23 -
empty.jpg.2lthjp0a empty Today, 16:40:07 -
empty.jpg.2lthj4ii empty Today, 16:39:46 -
empty.2bbu69jl empty Jul 25 2021, 12:56:39 -
empty.2bbu67q4 empty Jul 25 2021, 12:56:37 -
empty.2bbu666n empty Jul 25 2021, 12:56:35 -
Record new data
Connect using WebUSB
Device
nano
Label
cracked
Sensor
Camera (160x120)
Camera feed
Start sampling
RAW DATA
cracked.jpg.2lti3gef
https://studio.edgeimpulse.com/studio/42476/acquisition/training?page=1#!

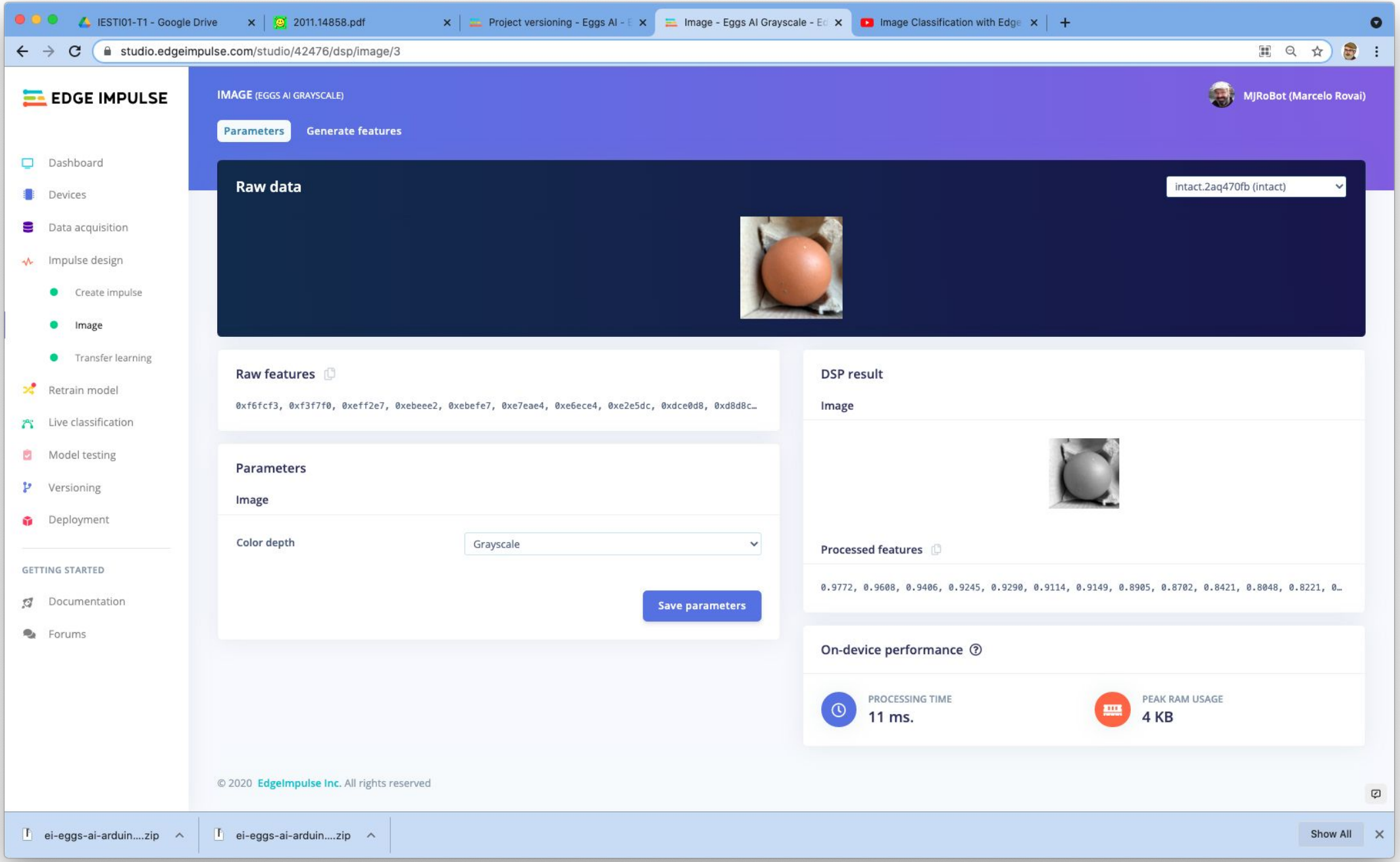
Collect a dataset, clean and pre-process *

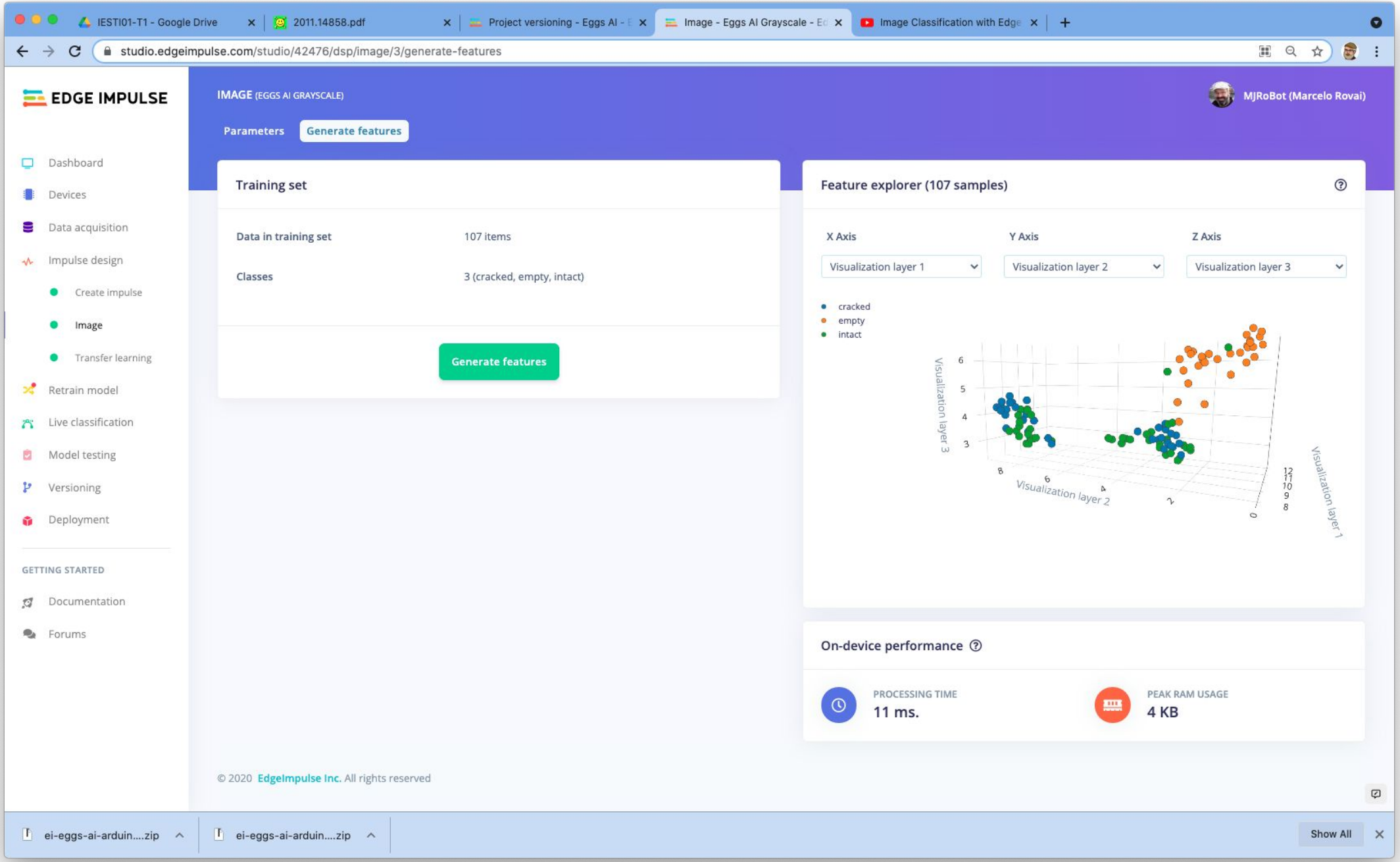


- Image Pre-Process *:
  - Convert to Grayscale
  - Re-scale 96 x 96

* During Inference, the OV7675 captures image as:
- QCIF (176 x 144)
- RGB565


IESTI01-T1 - Google Drive
2011.14858.pdf
Project versioning - Eggs AI -
Create impulse - Eggs AI Gray
Image Classification with Edge
studio.edgeimpulse.com/studio/42476/create-impulse
EDGE IMPULSE
CREATE IMPULSE (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Image data
Axes
image
Image width
96
Image height
96
Resize mode
Fit shortest axis
For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.
Image
Name
Image
Input axes
image
Transfer Learning (Images)
Name
Transfer learning
Input features
Image
Output features
3 (cracked, empty, intact)
Output features
3 (cracked, empty, intact)
Save Impulse
Add a processing block
Add a learning block
© 2020 EdgeImpulse Inc. All rights reserved
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All


IESTI01-T1 - Google Drive
2011.14858.pdf
Project versioning - Eggs AI -
Image - Eggs AI Grayscale - Ed
Image Classification with Edge
studio.edgeimpulse.com/studio/42476/dsp/image/3
EDGE IMPULSE
IMAGE (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
Parameters
Generate features
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Raw data
intact.2aq470fb (intact)
Raw features
0xf6fcf3, 0xf3f7f0, 0xeff2e7, 0xebeee2, 0xebefe7, 0xe7eae4, 0xe6ece4, 0xe2e5dc, 0xdce0d8, 0xd8d8c…
Parameters
Image
Color depth
Grayscale
Save parameters
DSP result
Image
Processed features
0.9772, 0.9608, 0.9406, 0.9245, 0.9290, 0.9114, 0.9149, 0.8905, 0.8702, 0.8421, 0.8048, 0.8221, 0…
On-device performance
PROCESSING TIME
11 ms.
PEAK RAM USAGE
4 KB
© 2020 EdgeImpulse Inc. All rights reserved
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All


IESTI01-T1 - Google Drive
2011.14858.pdf
Project versioning - Eggs AI -
Image - Eggs AI Grayscale - Ed
Image Classification with Edge
studio.edgeimpulse.com/studio/42476/dsp/image/3/generate-features
EDGE IMPULSE
IMAGE (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
Parameters
Generate features
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Training set
Data in training set
107 items
Classes
3 (cracked, empty, intact)
Generate features
Feature explorer (107 samples)
X Axis
Y Axis
Z Axis
Visualization layer 1
Visualization layer 2
Visualization layer 3
cracked
empty
intact
Visualization layer 3
Visualization layer 2
Visualization layer 1
On-device performance
PROCESSING TIME
11 ms.
PEAK RAM USAGE
4 KB
© 2020 EdgeImpulse Inc. All rights reserved
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All

Design a model architecture (i.e.DNN, CNN)

Model

MobileNetV1 96x96 0.25

A pre-trained multi-layer convolutional network designed to efficiently classify images. Uses around 105.9K RAM and 301.6K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Image Size

MobileNetV1 96x96 0.2

Uses around 83.1K RAM and 218.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

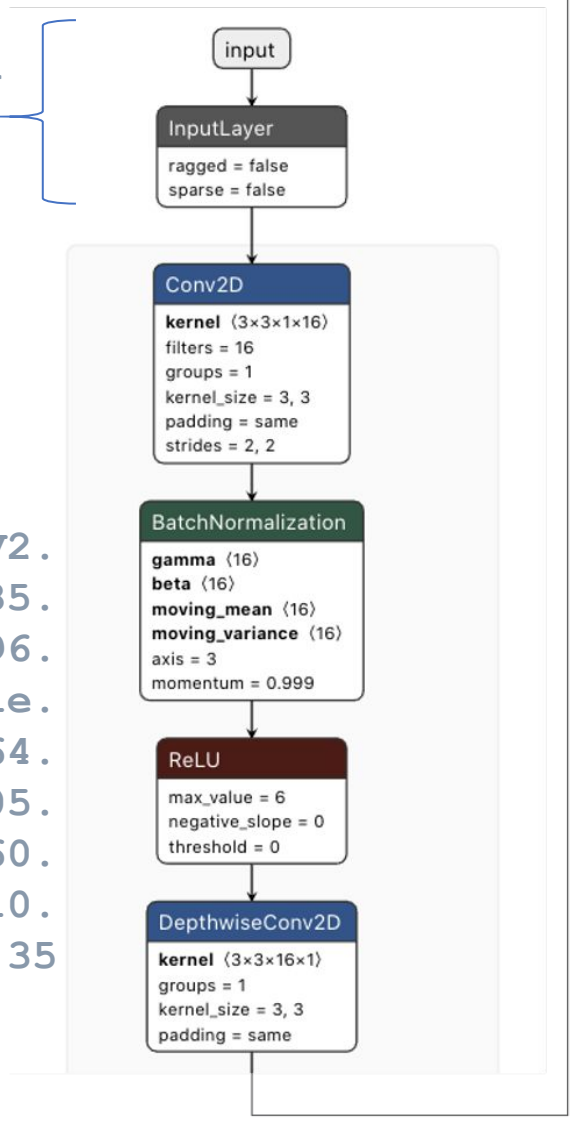
Alpha

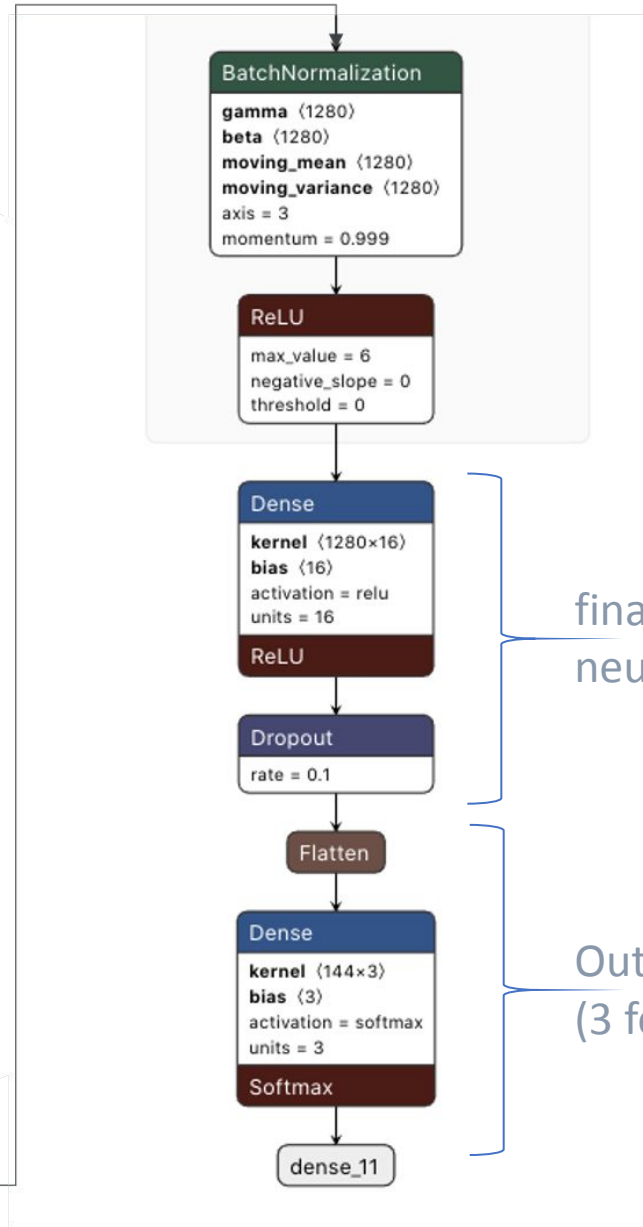
MobileNetV1 96x96 0.1

Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

MobileNetV2 96x96 0.35

Input layer
(9,216 features)

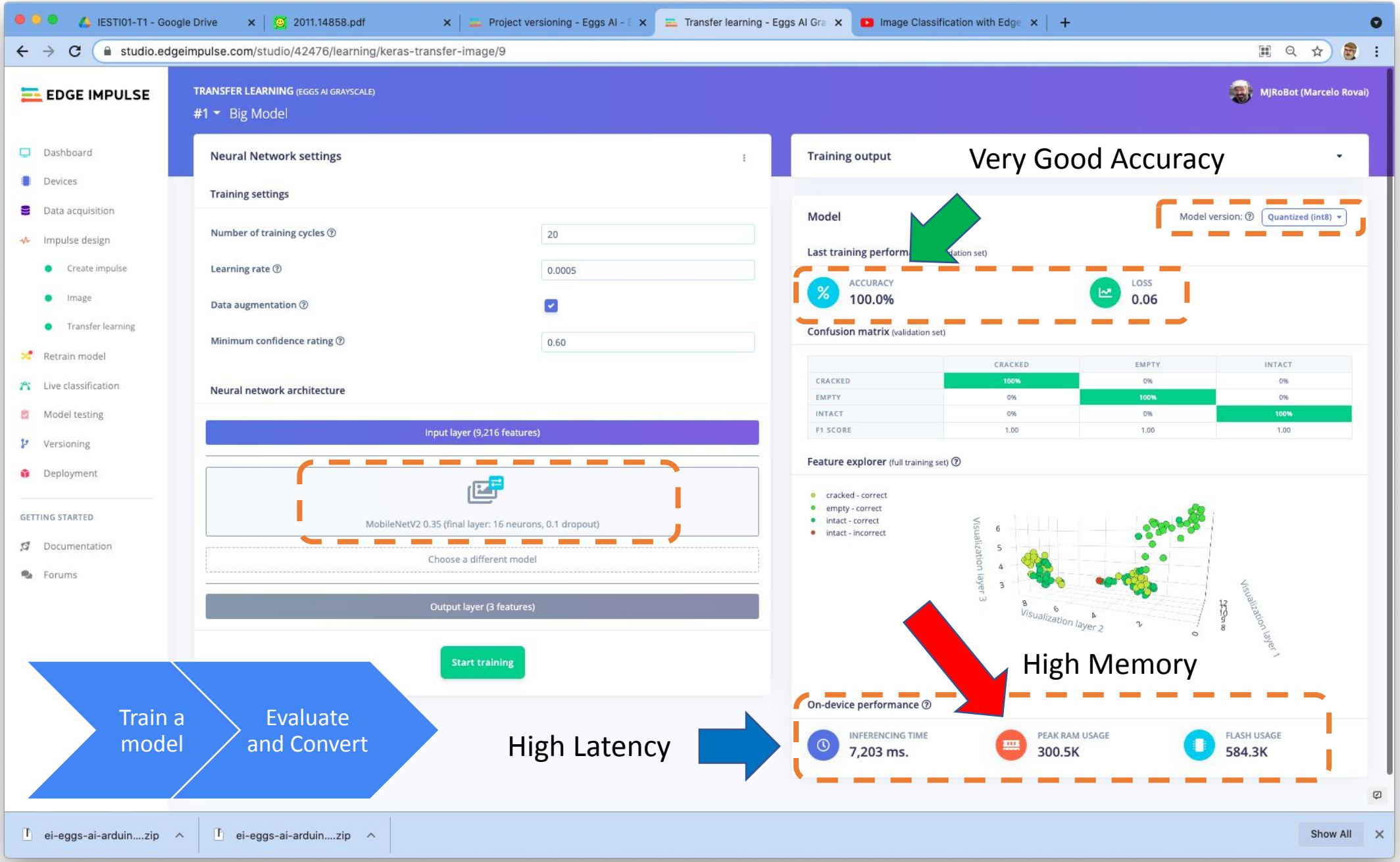




final layer: 16 neurons

Output layer
(3 features)

`MobileNetV2.`
`0_35.`
`96x96.`
`grayscale.`
`bsize_64.`
`lr_0_005.`
`epoch_260.`
`val_loss_3.10.`
`val_accuracy_0.35`


studio.edgeimpulse.com/studio/42476/learning/keras-transfer-image/9
EDGE IMPULSE
TRANSFER LEARNING (EGGS AI GRAYSCALE)
#1 Big Model
MJRoBot (Marcelo Rovai)
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Neural Network settings
Training settings
Number of training cycles
20
Learning rate
0.0005
Data augmentation
Minimum confidence rating
0.60
Neural network architecture
Input layer (9,216 features)
MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)
Choose a different model
Output layer (3 features)
Start training
Training output
Very Good Accuracy
Model
Model version:
Quantized (int8)
ACCURACY
100.0%
LOSS
0.06
Confusion matrix (validation set)
CRACKED
EMPTY
INTACT
F1 SCORE
Feature explorer (full training set)
cracked - correct
empty - correct
intact - correct
intact - incorrect
Visualization layer 1
Visualization layer 2
Visualization layer 3
High Memory
On-device performance
INFERENCING TIME
7,203 ms.
PEAK RAM USAGE
300.5K
FLASH USAGE
584.3K
High Latency
Train a model
Evaluate and Convert
ei-eggs-ai-arduin....zip
Show All

Benchmark

Arduino Nano 33 BLE Sense Cortex M4F 64 MHz

INFERENCING TIME	PEAK RAM USAGE	FLASH USAGE
7,203 ms.	300.5K	584.3K

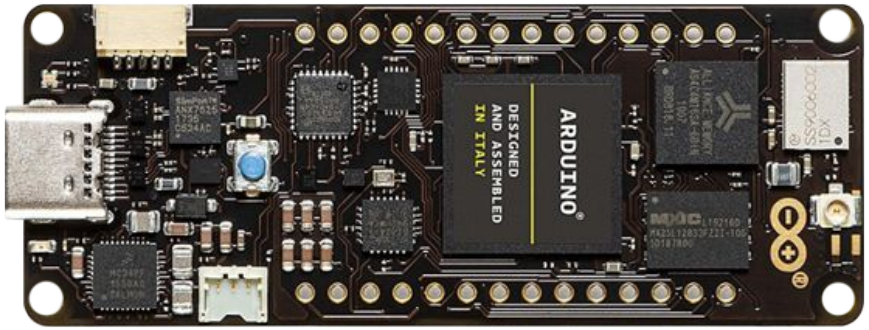
Problem -> High Latency and Nano only has 256KB SRAM

Cortex-M7 216 MHz

INFERENCING TIME	PEAK RAM USAGE	FLASH USAGE
855 ms.	300.5K	584.3K

Intel Core i9 2.4GHz

INFERENCING TIME	PEAK RAM USAGE	FLASH USAGE
68 ms.	300.5K	584.3K



Portenta H7

- Portenta form factor
- STM32H747
- Dual core
  - Arm Cortex M7@480MHz
  - Arm Cortex M4@240MHz
- up to 64 MByte SDRAM
- up to 128 MByte QSPI Flash
- Common Criteria Crypto
- WiFi b/g/n 65Mbps + BT 5.1
- 100Mbit Ethernet
- CAN
- USB-C with DisplayPort output

life.augmented

13:05
◂ Camera

AA ne.edgeimpulse.com

Classifier

Classify

	CRACKED	EMPTY	INTACT
1	0.01	**0.99**	0.00

Switch to data collection mode

13:05
◂ Camera

AA ne.edgeimpulse.com

Classifier

Classify

	CRACKED	EMPTY	INTACT
2	0.01	0.00	**0.99**
1	0.01	**0.99**	0.00

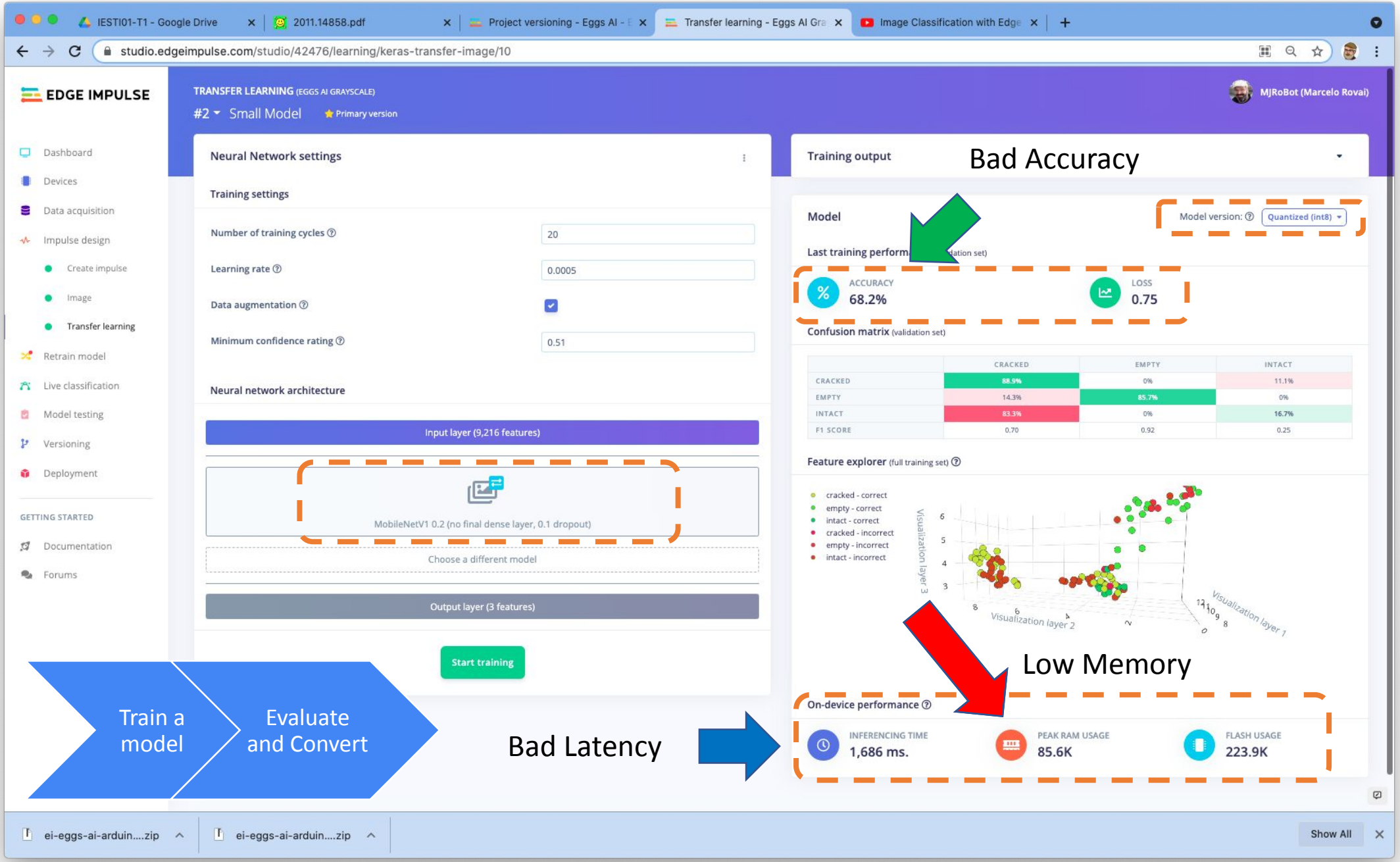
13:07
◂ Camera

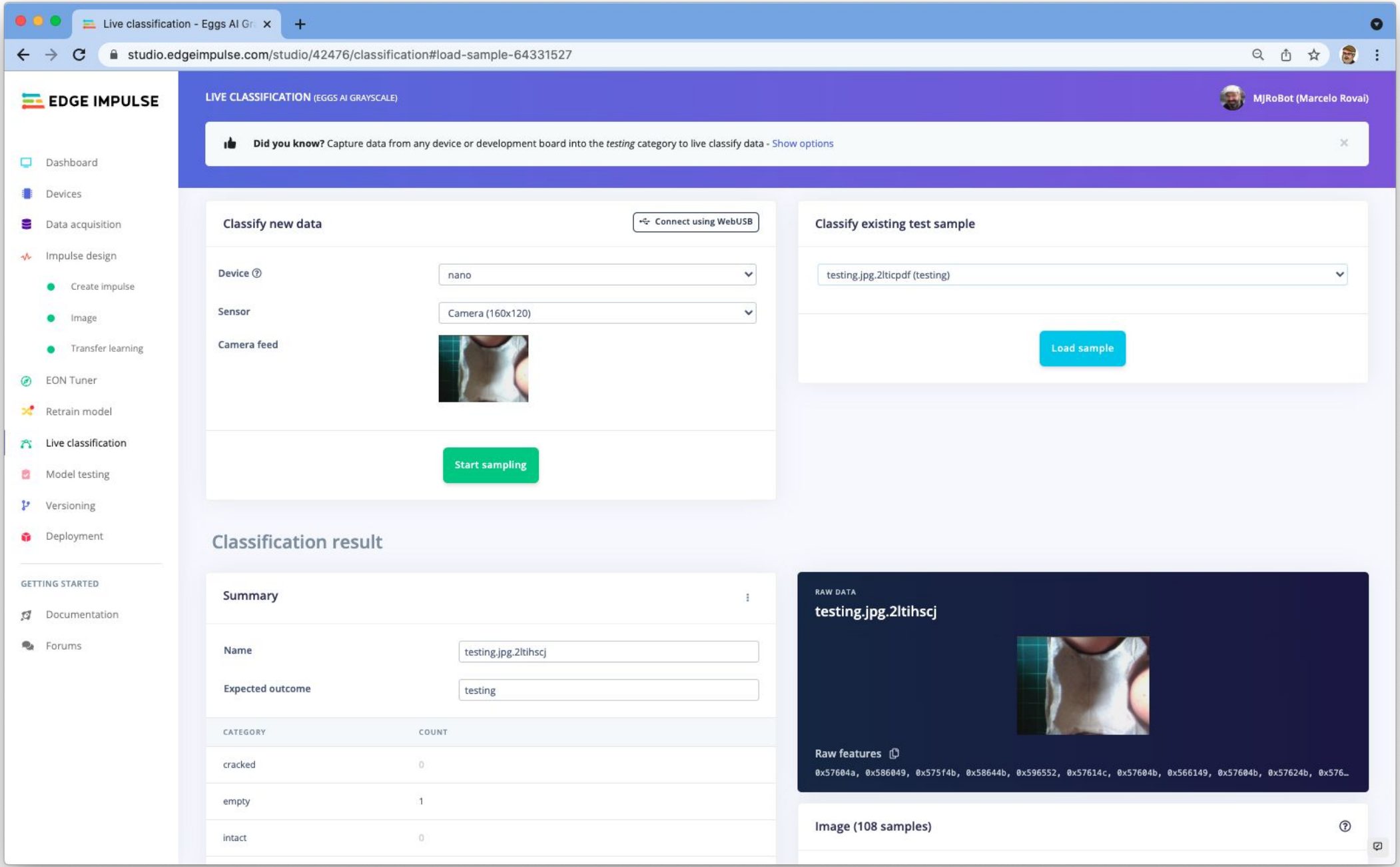
AA ne.edgeimpulse.com

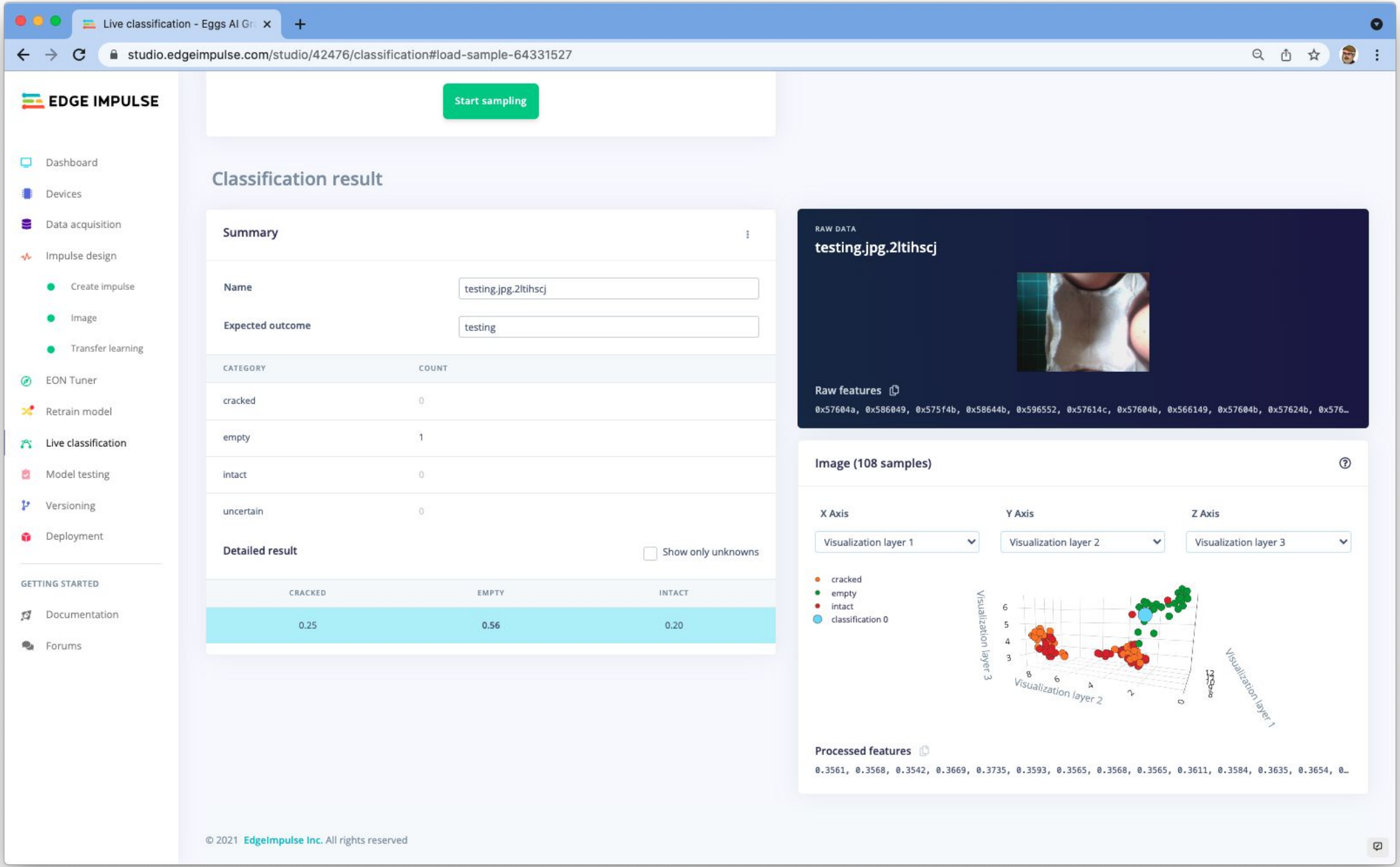
Classifier

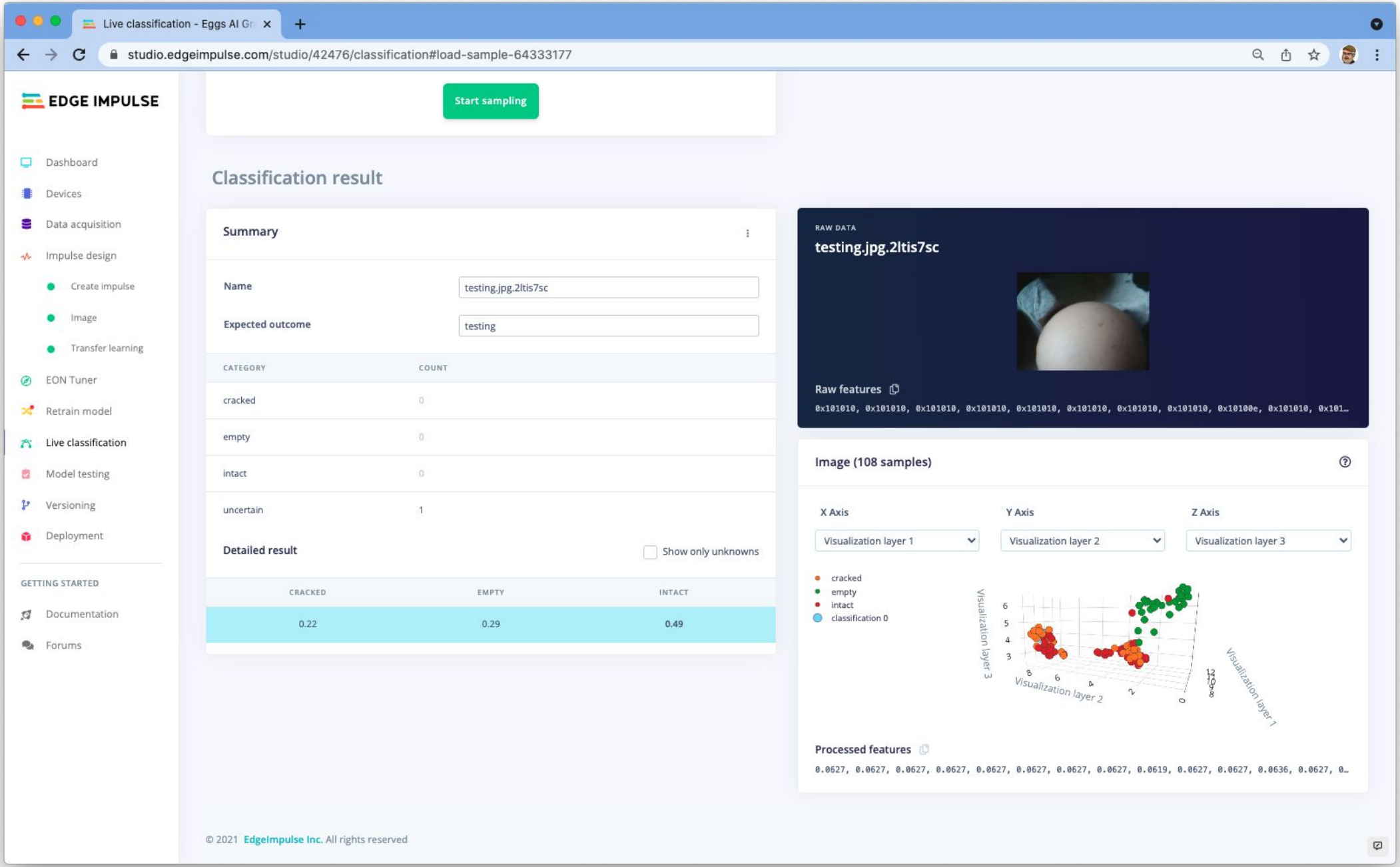
Classify

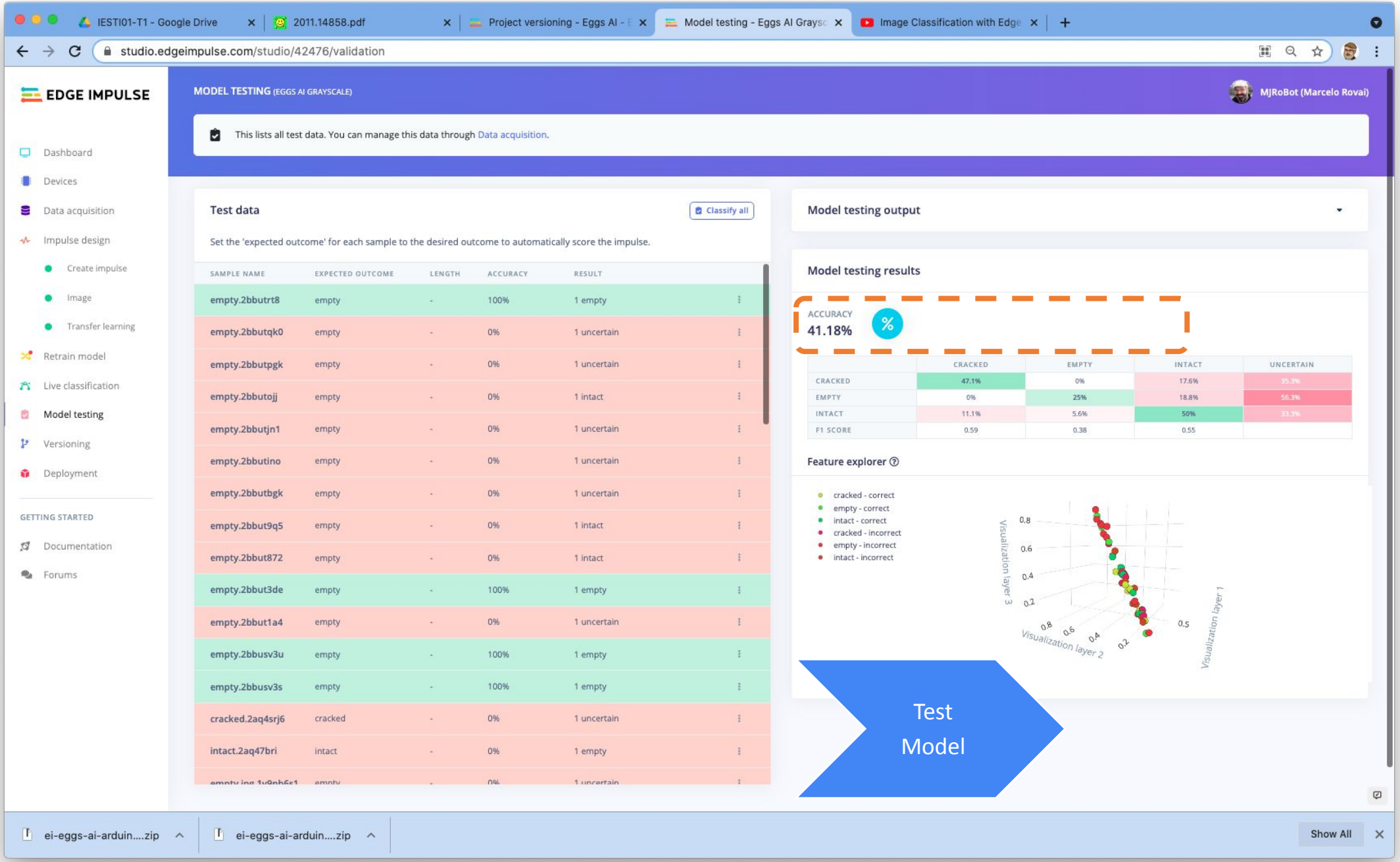
	CRACKED	EMPTY	INTACT
11	**0.93**	0.07	0.00
10	0.38	0.24	**0.38**

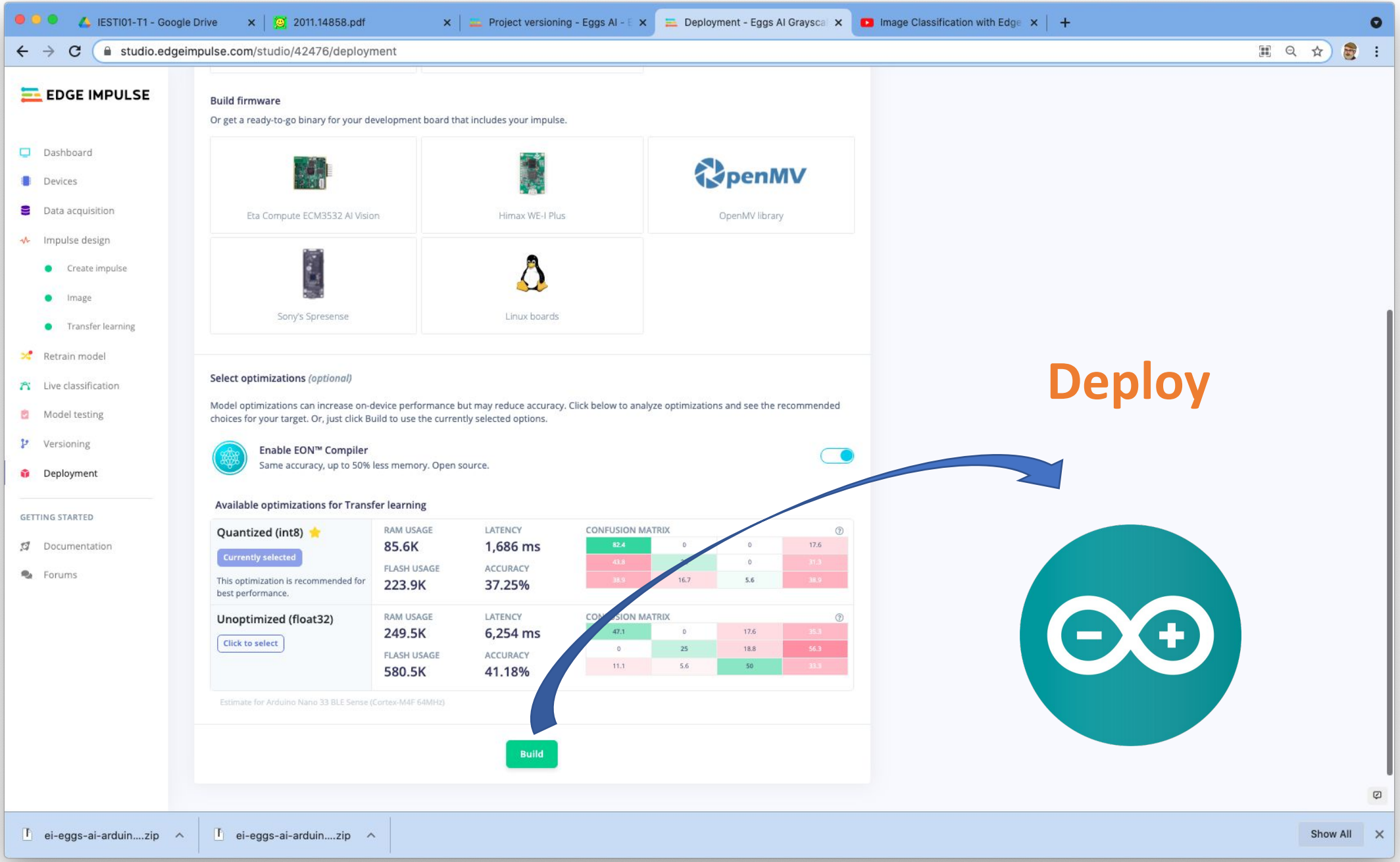

studio.edgeimpulse.com/studio/42476/learning/keras-transfer-image/10
EDGE IMPULSE
TRANSFER LEARNING (EGGS AI GRAYSCALE)
#2 Small Model
Primary version
MJRoBot (Marcelo Rovai)
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Neural Network settings
Training settings
Number of training cycles
20
Learning rate
0.0005
Data augmentation
Minimum confidence rating
0.51
Neural network architecture
Input layer (9,216 features)
MobileNetV1 0.2 (no final dense layer, 0.1 dropout)
Choose a different model
Output layer (3 features)
Start training
Training output
Bad Accuracy
Model
Model version:
Quantized (int8)
ACCURACY
68.2%
LOSS
0.75
Confusion matrix (validation set)
CRACKED
EMPTY
INTACT
F1 SCORE
Feature explorer (full training set)
cracked - correct
empty - correct
intact - correct
cracked - incorrect
empty - incorrect
intact - incorrect
Visualization layer 1
Visualization layer 2
Visualization layer 3
Low Memory
On-device performance
INFERENCING TIME
1,686 ms.
PEAK RAM USAGE
85.6K
FLASH USAGE
223.9K
Train a model
Evaluate and Convert
Bad Latency
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All


Live classification - Eggs AI Gr
studio.edgeimpulse.com/studio/42476/classification#load-sample-64331527
EDGE IMPULSE
LIVE CLASSIFICATION (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
Did you know? Capture data from any device or development board into the testing category to live classify data - Show options
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Classify new data
Connect using WebUSB
Device
nano
Sensor
Camera (160x120)
Camera feed
Start sampling
Classify existing test sample
testing.jpg.2lticpdf (testing)
Load sample
Classification result
Summary
Name
testing.jpg.2ltihscj
Expected outcome
testing
CATEGORY
COUNT
cracked
0
empty
1
intact
0
RAW DATA
testing.jpg.2ltihscj
Raw features
0x57604a, 0x586049, 0x575f4b, 0x58644b, 0x596552, 0x57614c, 0x57604b, 0x566149, 0x57604b, 0x57624b, 0x576…
Image (108 samples)


Live classification - Eggs AI Gr
studio.edgeimpulse.com/studio/42476/classification#load-sample-64331527
EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Start sampling
Classification result
Summary
Name
testing.jpg.2ltihscj
Expected outcome
testing
CATEGORY
COUNT
cracked
0
empty
1
intact
0
uncertain
0
Detailed result
Show only unknowns
CRACKED
EMPTY
INTACT
0.25
0.56
0.20
RAW DATA
testing.jpg.2ltihscj
Raw features
0x57604a, 0x586049, 0x575f4b, 0x58644b, 0x596552, 0x57614c, 0x57604b, 0x566149, 0x57604b, 0x57624b, 0x576…
Image (108 samples)
X Axis
Y Axis
Z Axis
Visualization layer 1
Visualization layer 2
Visualization layer 3
cracked
empty
intact
classification 0
Visualization layer 3
Visualization layer 2
Visualization layer 1
Processed features
0.3561, 0.3568, 0.3542, 0.3669, 0.3735, 0.3593, 0.3565, 0.3568, 0.3565, 0.3611, 0.3584, 0.3635, 0.3654, 0…
© 2021 EdgeImpulse Inc. All rights reserved


Live classification - Eggs AI Gr
studio.edgeimpulse.com/studio/42476/classification#load-sample-64333177
EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Start sampling
Classification result
Summary
Name
testing.jpg.2ltis7sc
Expected outcome
testing
CATEGORY
COUNT
cracked
0
empty
0
intact
0
uncertain
1
Detailed result
Show only unknowns
CRACKED
EMPTY
INTACT
0.22
0.29
0.49
RAW DATA
testing.jpg.2ltis7sc
Raw features
0x101010, 0x101010, 0x101010, 0x101010, 0x101010, 0x101010, 0x101010, 0x101010, 0x10100e, 0x101010, 0x101…
Image (108 samples)
X Axis
Y Axis
Z Axis
Visualization layer 1
Visualization layer 2
Visualization layer 3
cracked
empty
intact
classification 0
Visualization layer 3
Visualization layer 2
Visualization layer 1
Processed features
0.0627, 0.0627, 0.0627, 0.0627, 0.0627, 0.0627, 0.0627, 0.0627, 0.0619, 0.0627, 0.0627, 0.0636, 0.0627, 0…
© 2021 EdgeImpulse Inc. All rights reserved

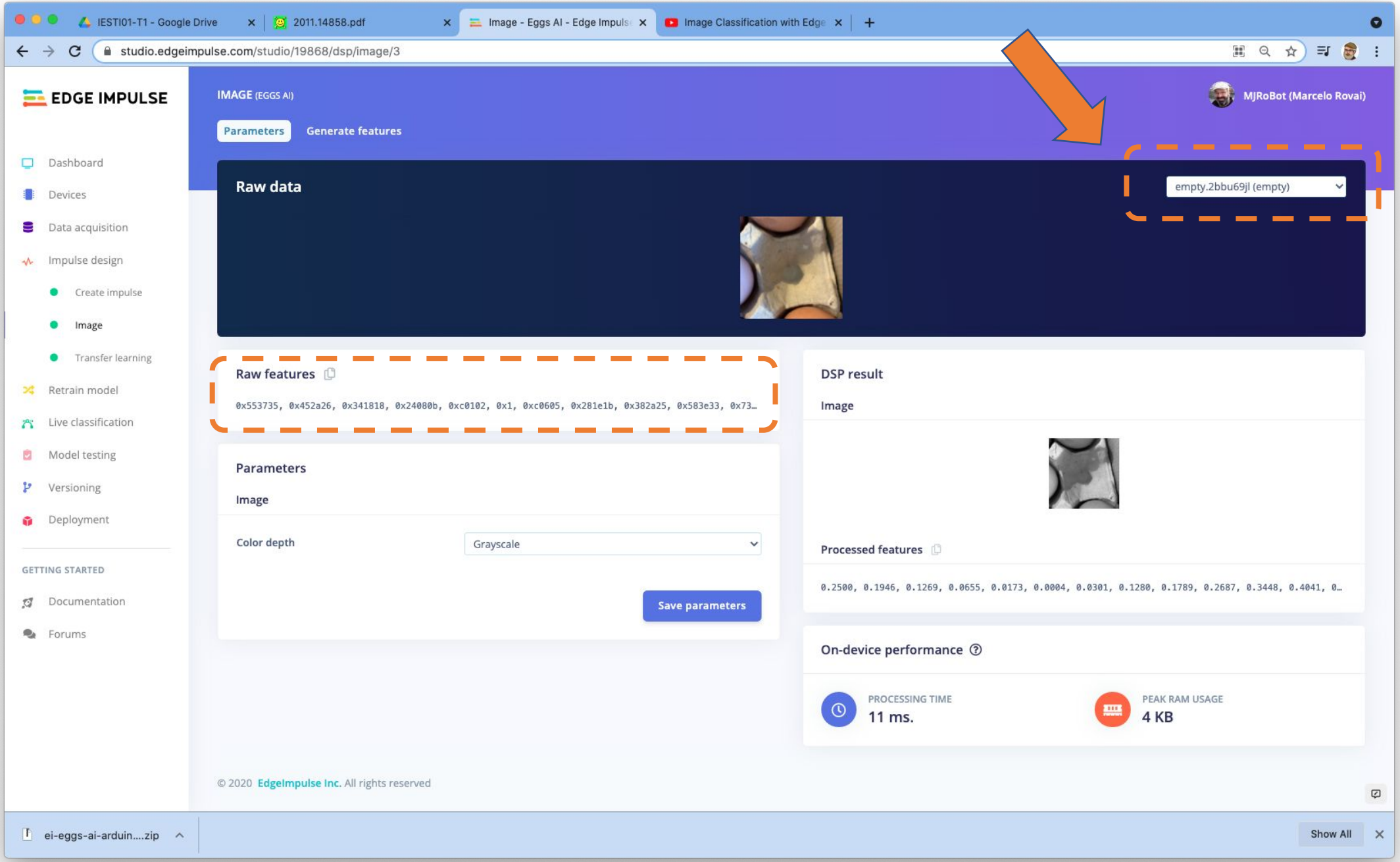

IESTI01-T1 - Google Drive
2011.14858.pdf
Project versioning - Eggs AI -
Model testing - Eggs AI Grays
Image Classification with Edge
studio.edgeimpulse.com/studio/42476/validation
EDGE IMPULSE
MODEL TESTING (EGGS AI GRAYSCALE)
MJRoBot (Marcelo Rovai)
This lists all test data. You can manage this data through Data acquisition.
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Test data
Classify all
Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.
SAMPLE NAME
EXPECTED OUTCOME
LENGTH
ACCURACY
RESULT
empty.2bbutrt8 empty - 100% 1 empty
empty.2bbutqk0 empty - 0% 1 uncertain
empty.2bbutpgk empty - 0% 1 uncertain
empty.2bbutojj empty - 0% 1 intact
empty.2bbutjn1 empty - 0% 1 uncertain
empty.2bbutino empty - 0% 1 uncertain
empty.2bbutbgk empty - 0% 1 uncertain
empty.2bbut9q5 empty - 0% 1 intact
empty.2bbut872 empty - 0% 1 intact
empty.2bbut3de empty - 100% 1 empty
empty.2bbut1a4 empty - 0% 1 uncertain
empty.2bbusv3u empty - 100% 1 empty
empty.2bbusv3s empty - 100% 1 empty
cracked.2aq4srj6 cracked - 0% 1 uncertain
intact.2aq47bri intact - 0% 1 empty
Model testing output
Model testing results
ACCURACY
41.18%
CRACKED EMPTY INTACT UNCERTAIN
CRACKED 47.1% 0% 17.6% 35.3%
EMPTY 0% 25% 18.8% 56.3%
INTACT 11.1% 5.6% 50% 33.3%
F1 SCORE 0.59 0.38 0.55
Feature explorer
cracked - correct
empty - correct
intact - correct
cracked - incorrect
empty - incorrect
intact - incorrect
Visualization layer 1
Visualization layer 2
Visualization layer 3
Test
Model
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All


IESTI01-T1 - Google Drive
2011.14858.pdf
Project versioning - Eggs AI -
Deployment - Eggs AI Grayscal
Image Classification with Edge
studio.edgeimpulse.com/studio/42476/deployment
EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Build firmware
Or get a ready-to-go binary for your development board that includes your impulse.
Eta Compute ECM3532 AI Vision
Himax WE-I Plus
OpenMV library
Sony's Spresense
Linux boards
Select optimizations (optional)
Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.
Enable EON™ Compiler
Same accuracy, up to 50% less memory. Open source.
Available optimizations for Transfer learning
Quantized (int8)
Currently selected
This optimization is recommended for best performance.
RAM USAGE
85.6K
LATENCY
1,686 ms
FLASH USAGE
223.9K
ACCURACY
37.25%
CONFUSION MATRIX
Unoptimized (float32)
Click to select
RAM USAGE
249.5K
LATENCY
6,254 ms
FLASH USAGE
580.5K
ACCURACY
41.18%
Estimate for Arduino Nano 33 BLE Sense (Cortex-M4F 64MHz)
Build
Deploy
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All

Egg Classification using Transfer Learning Model
Code Walkthrough!

static_buffer.ino
EI_Image_Classifier_Eggs.ino

Thanks to Jan Jongboom (Edge Impulse) and Jeremy Ellis (@rocksetta)


IESTI01-T1 - Google Drive
2011.14858.pdf
Image - Eggs AI - Edge Impuls
Image Classification with Edge
studio.edgeimpulse.com/studio/19868/dsp/image/3
EDGE IMPULSE
IMAGE (EGGS AI)
MJRoBot (Marcelo Rovai)
Parameters
Generate features
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Raw data
empty.2bbu69jl (empty)
Raw features
0x553735, 0x452a26, 0x341818, 0x24080b, 0xc0102, 0x1, 0xc0605, 0x281e1b, 0x382a25, 0x583e33, 0x73…
Parameters
Image
Color depth
Grayscale
Save parameters
DSP result
Image
Processed features
0.2500, 0.1946, 0.1269, 0.0655, 0.0173, 0.0004, 0.0301, 0.1280, 0.1789, 0.2687, 0.3448, 0.4041, 0…
On-device performance
PROCESSING TIME
11 ms.
PEAK RAM USAGE
4 KB
© 2020 EdgeImpulse Inc. All rights reserved
ei-eggs-ai-arduin....zip
Show All

Testing Model inference

(static_buffer.ino)

static_buffer | Arduino 1.8.15

static_buffer §

```
 * 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.
 */

/* Includes ---------------------------------------------------------------- */
#include <Eggs_AI_inferencing.h>

static const float features[] = {
    // copy raw features here (for example from the 'Live classification' page)
    // see https://docs.edgeimpulse.com/docs/running-your-impulse-arduino
    0xf6fcf3, 0xf3f7f0, 0xeff2e7, 0xebeee2, 0xebefe7, 0xe7eae4, 0xe6ece4, 0xe2e
};

/**
 * @brief      Copy raw feature data in out_ptr
 *             Function called by inference library
 *
 * @param[in]  offset   The offset
 * @param[in]  length   The length
 * @param      out_ptr  The out pointer
 *
 * @return     0
 */
int raw_feature_get_data(size_t offset, size_t length, float *out_ptr) {
    memcpy(out_ptr, features + offset, length * sizeof(float));
    return 0;
}
```

29 Arduino Nano 33 BLE on /dev/cu.usbmodem144301

/dev/cu.usbmodem144301

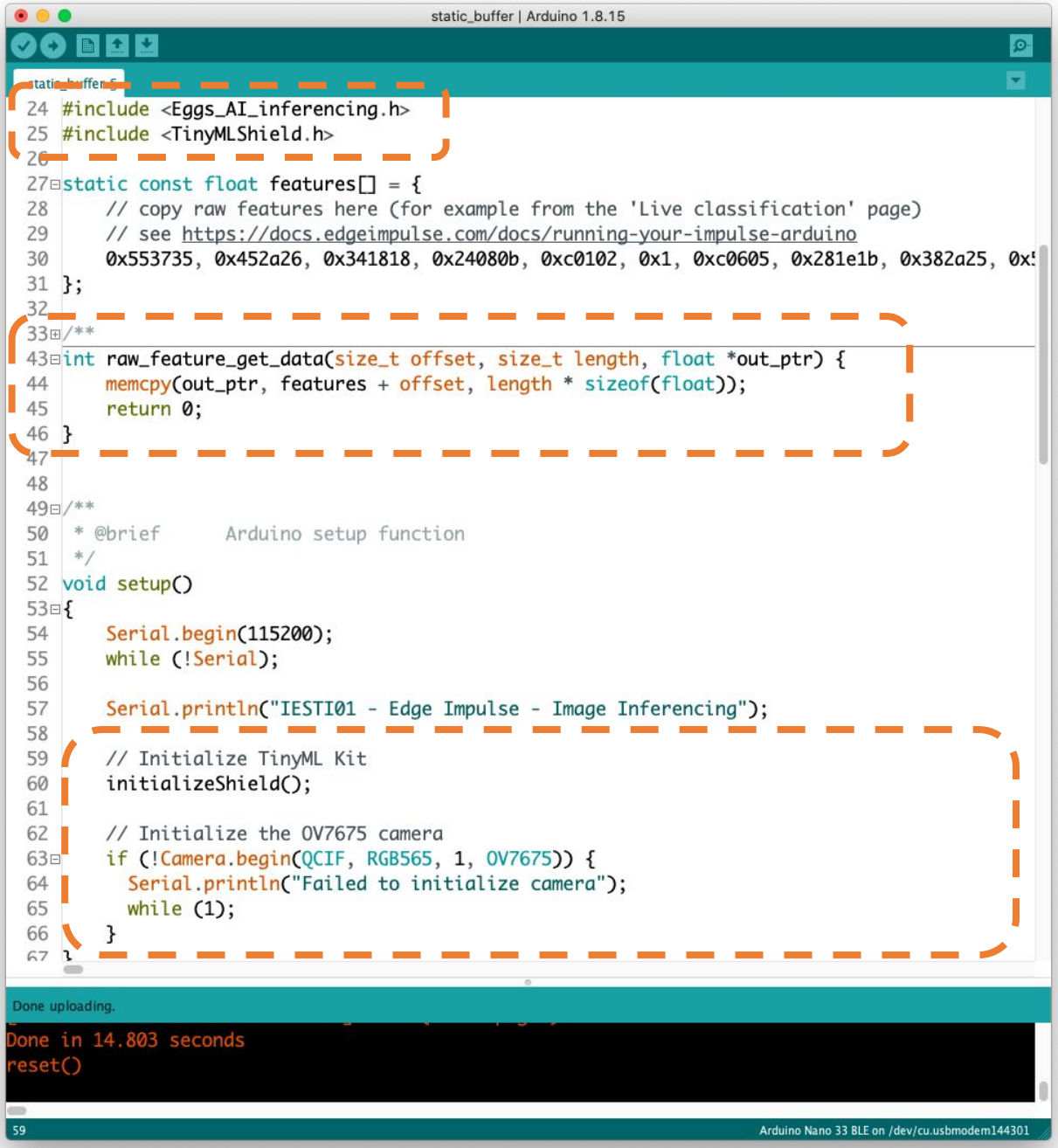
```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 8 ms., Classification: 694 ms., Anomaly: 0 ms.):
[0.27344, 0.54297, 0.18359]
    cracked: 0.27344
    empty: 0.54297
    intact: 0.18359
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 8 ms., Classification: 694 ms., Anomaly: 0 ms.):
[0.27344, 0.54297, 0.18359]
    cracked: 0.27344
    empty: 0.54297
    intact: 0.18359
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 8 ms., Classification: 694 ms., Anomaly: 0 ms.):
[0.27344, 0.54297, 0.18359]
    cracked: 0.27344
    empty: 0.54297
    intact: 0.18359
```

Send

Autoscroll | Show timestamp | Both NL & CR | 115200 baud | Clear output

Testing Model inference and Camera initialization

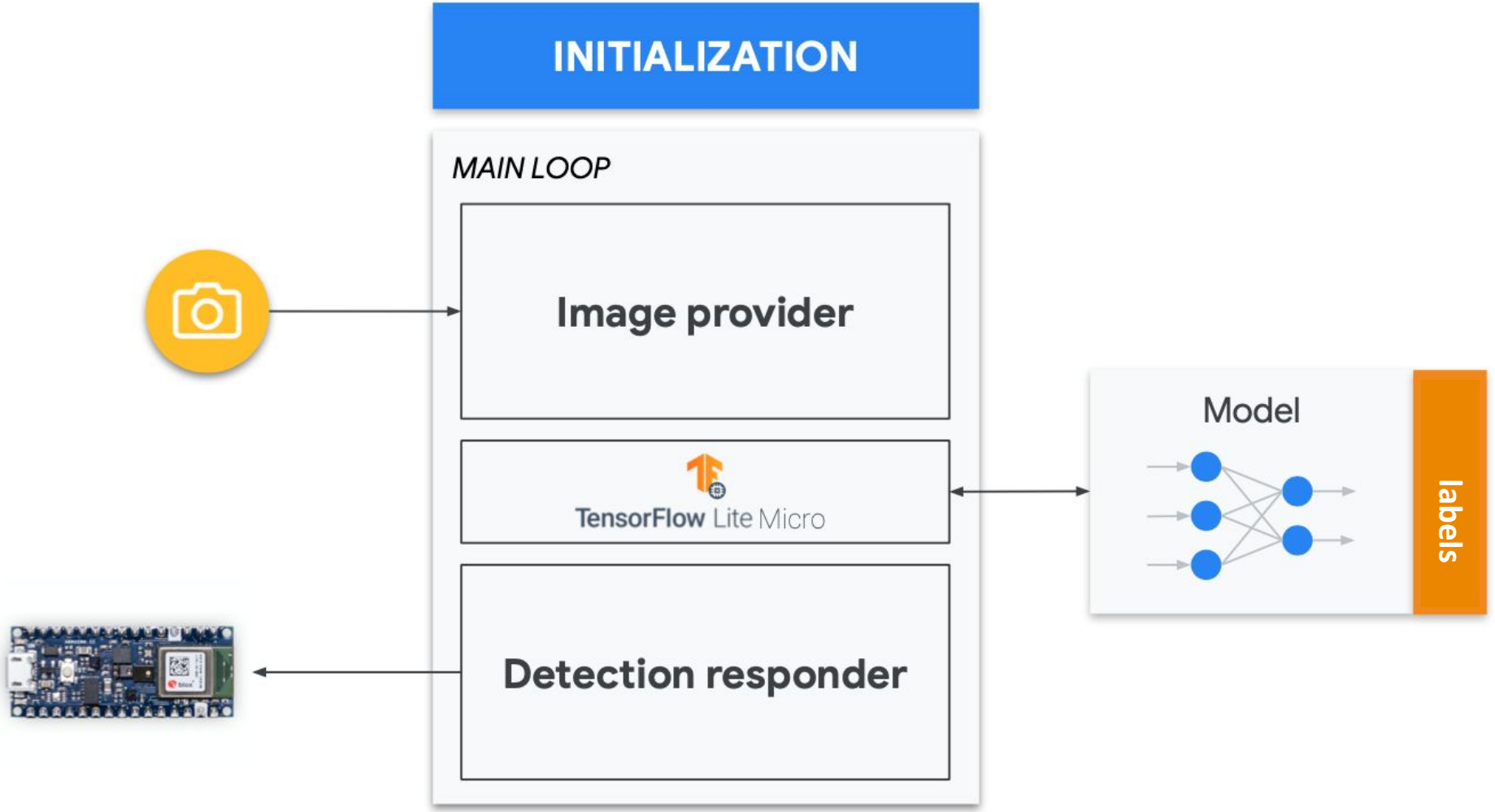
(static_buffer.ino)



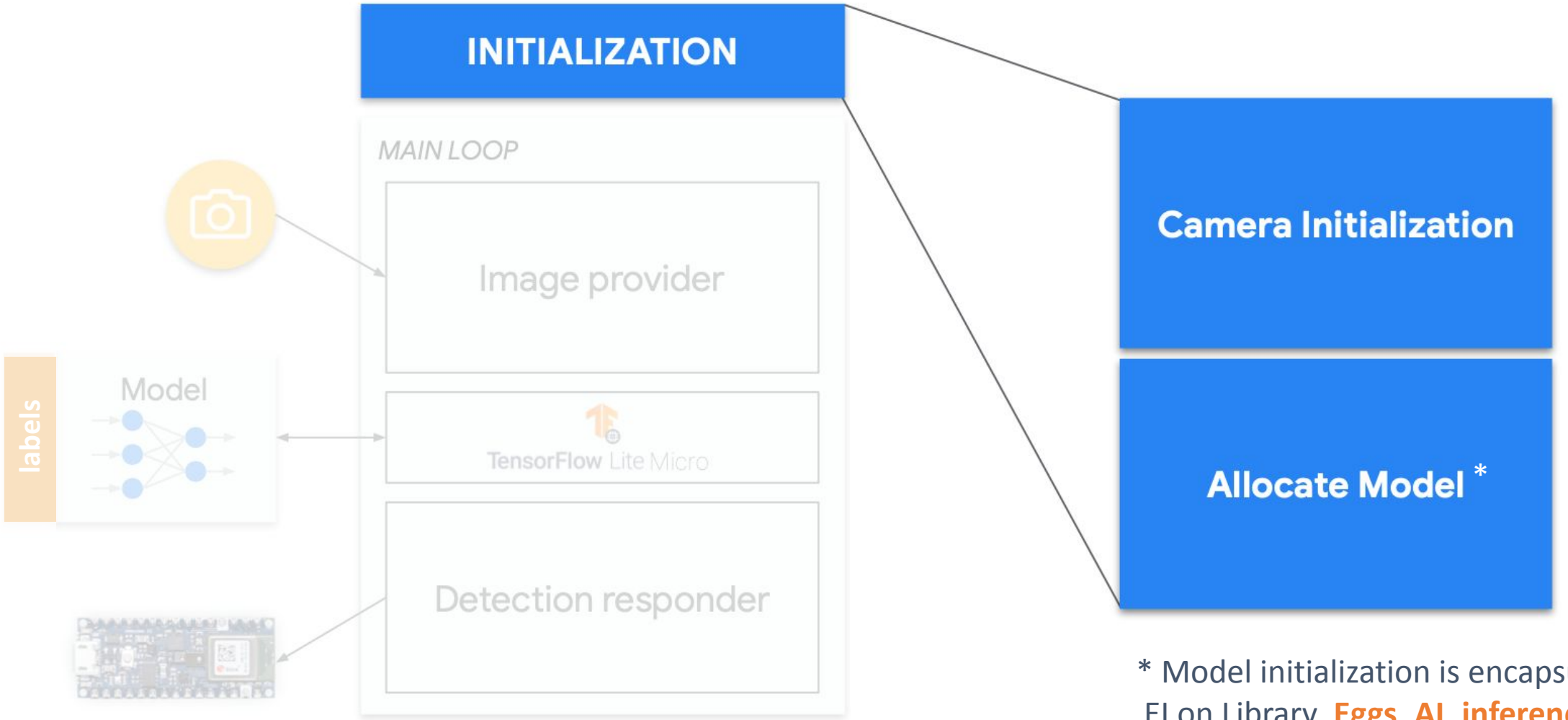
Function to Copy raw feature data in out_ptr
This function is called by inference library

Initialize Kit and Camera

Egg Classification Components (EI_Image_Classifier_Eggs.ino)



Initialization



* Model initialization is encapsulated by EI on Library **Eggs_AI_inferencing.h**

Including main libraries for inference and Camera handling

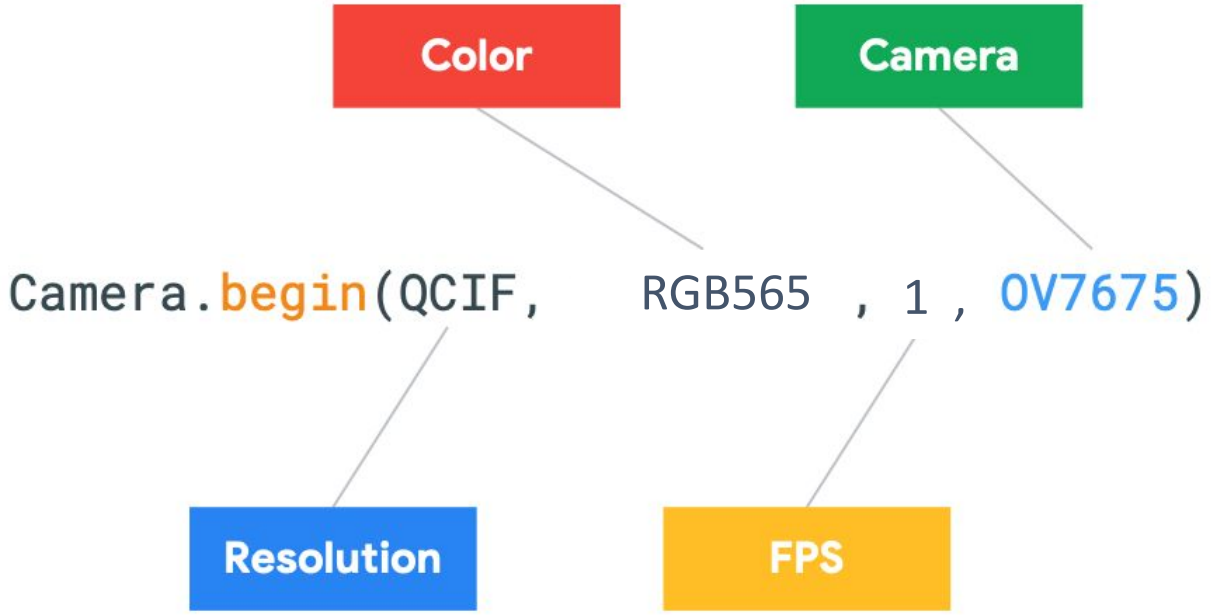
Main Image definitions

Auxiliary functions for image format conversion

Initialization

Camera Initialization

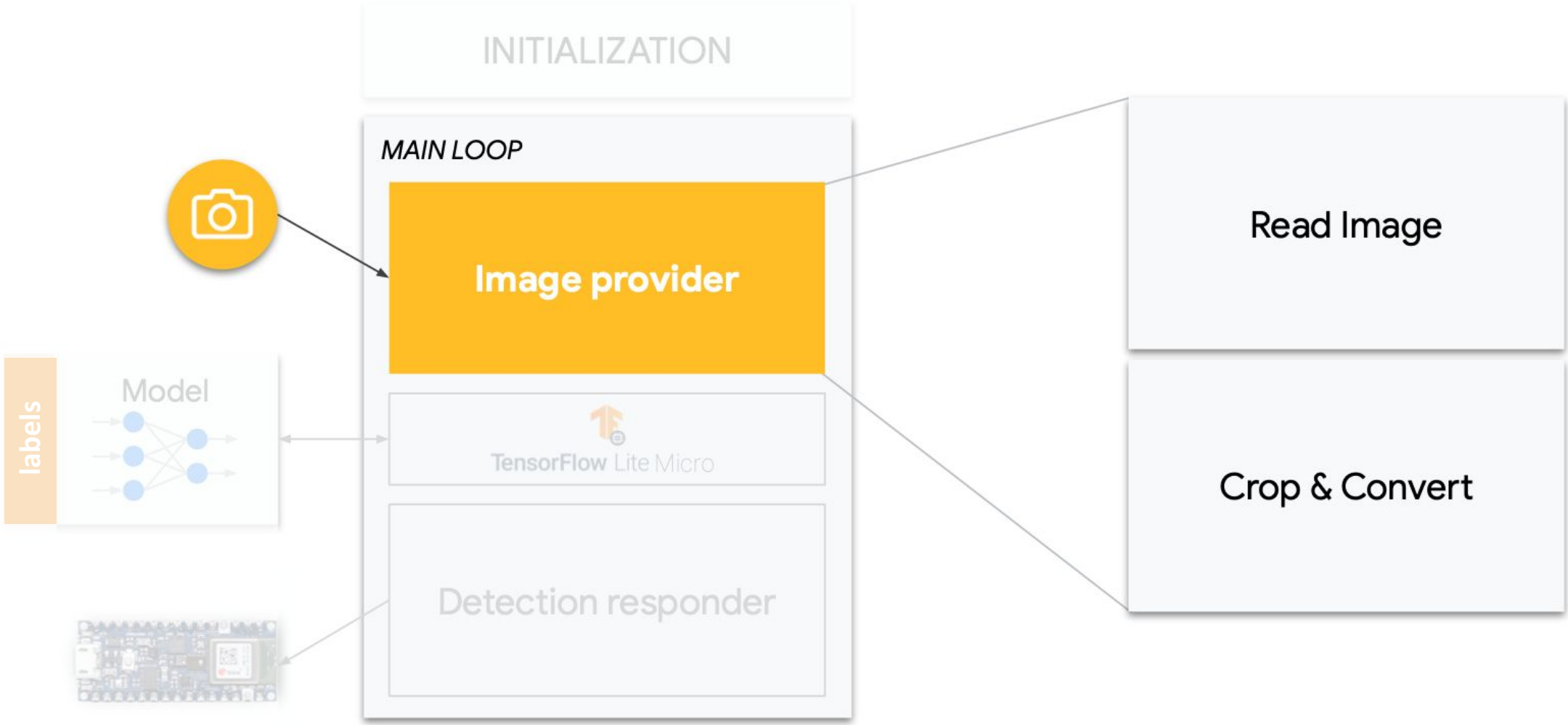
Allocate Model



Initialize Serial and display main inference settings as Image height/width, TFLM Arena size, etc.

Initialize TinyML Kit and Camera

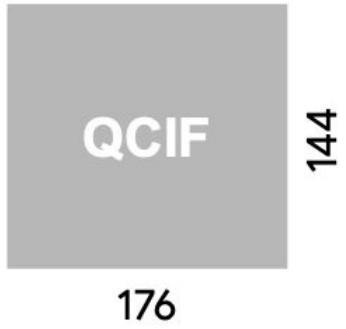
Pre-processing



Pre-processing

Read Image

Crop & Convert



```
// Get image from Camera
Camera.readFrame((uint8_t*)frame_buffer);

// Set up pointer to look after data, crop it and convert it to RGB888
signal_t signal;
signal.total_length = CUTOUT_COLS * CUTOUT_ROWS;
signal.get_data = &cutout_get_data;
```

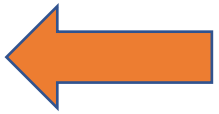
EI_Image_Classifier_Eggs | Arduino 1.8.15

EI_Image_Classifier_Eggs §

```
65
66 int cutout_get_data(size_t offset, size_t length, float *out_ptr) {
67     // offset and length naturally operate on the *cutout*,
68     // so we need to cut it out from the real framebuffer
69     size_t bytes_left = length;
70     size_t out_ptr_ix = 0;
71
72     // read byte for byte
73     while (bytes_left != 0) {
74         // find location of the byte in the cutout
75         size_t cutout_row = floor(offset / CUTOUT_COLS);
76         size_t cutout_col = offset - (cutout_row * CUTOUT_COLS);
77
78         // then read the value from the real frame buffer
79         size_t frame_buffer_row = cutout_row + cutout_row_start;
80         size_t frame_buffer_col = cutout_col + cutout_col_start;
81
82         // grab the value and convert to r/g/b
83         uint16_t pixelTemp = frame_buffer[(frame_buffer_row * FRAME_BUFFER_COLS) + frame_buffer_col];
84
85         // This line needed to switch big and little endians
86         uint16_t pixel = (pixelTemp>>8) | (pixelTemp<<8);
87
88         uint8_t r, g, b;
89         r565_to_rgb(pixel, &r, &g, &b);
90
91         // then convert to out_ptr format
92         float pixel_f = (r << 16) + (g << 8) + b;
93         //float pixel_f = (r << 16) | (g << 8) | b;
94         out_ptr[out_ptr_ix] = pixel_f;
95
96         // and go to the next pixel
97         out_ptr_ix++;
98         offset++;
99         bytes_left--;
100    }
101    return 0;
102 }
103
```

```
writeBuffer(scr_addr=0x34, dst_addr=0x53000, size=0x1000)
[==============================] 100% (84/84 pages)
Done in 13.373 seconds
reset()
```

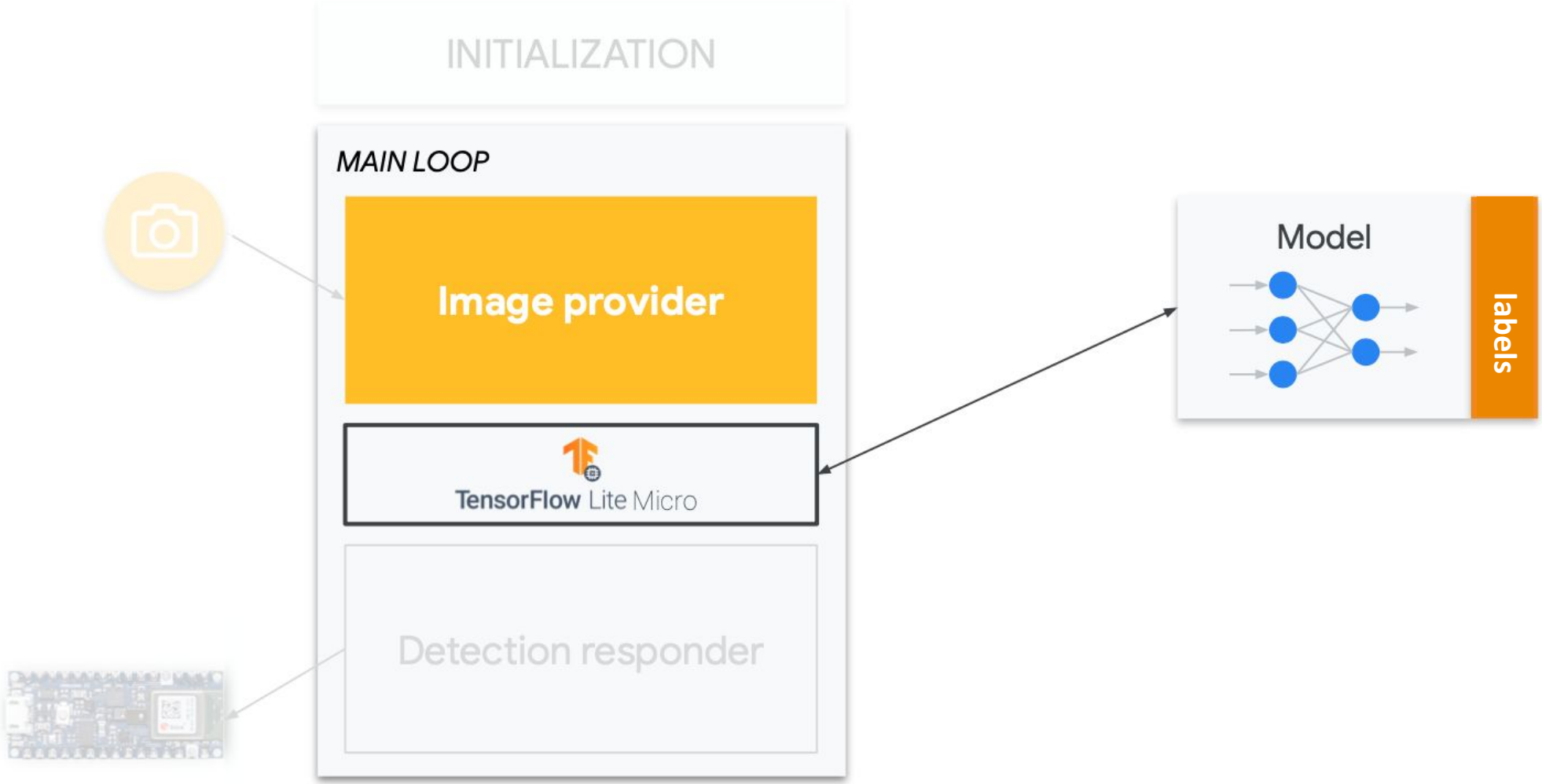
65 Arduino Nano 33 BLE on /dev/cu.usbmodem144301



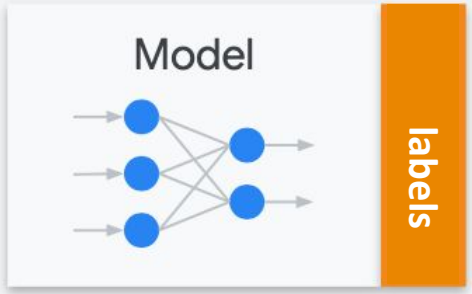
This function is called by the classifier (during loop), to get image data (frame buffer)

NOTE: There is not a separate copy of the cutout here, so we'll read from the frame buffer dynamically

Interpreter + Model

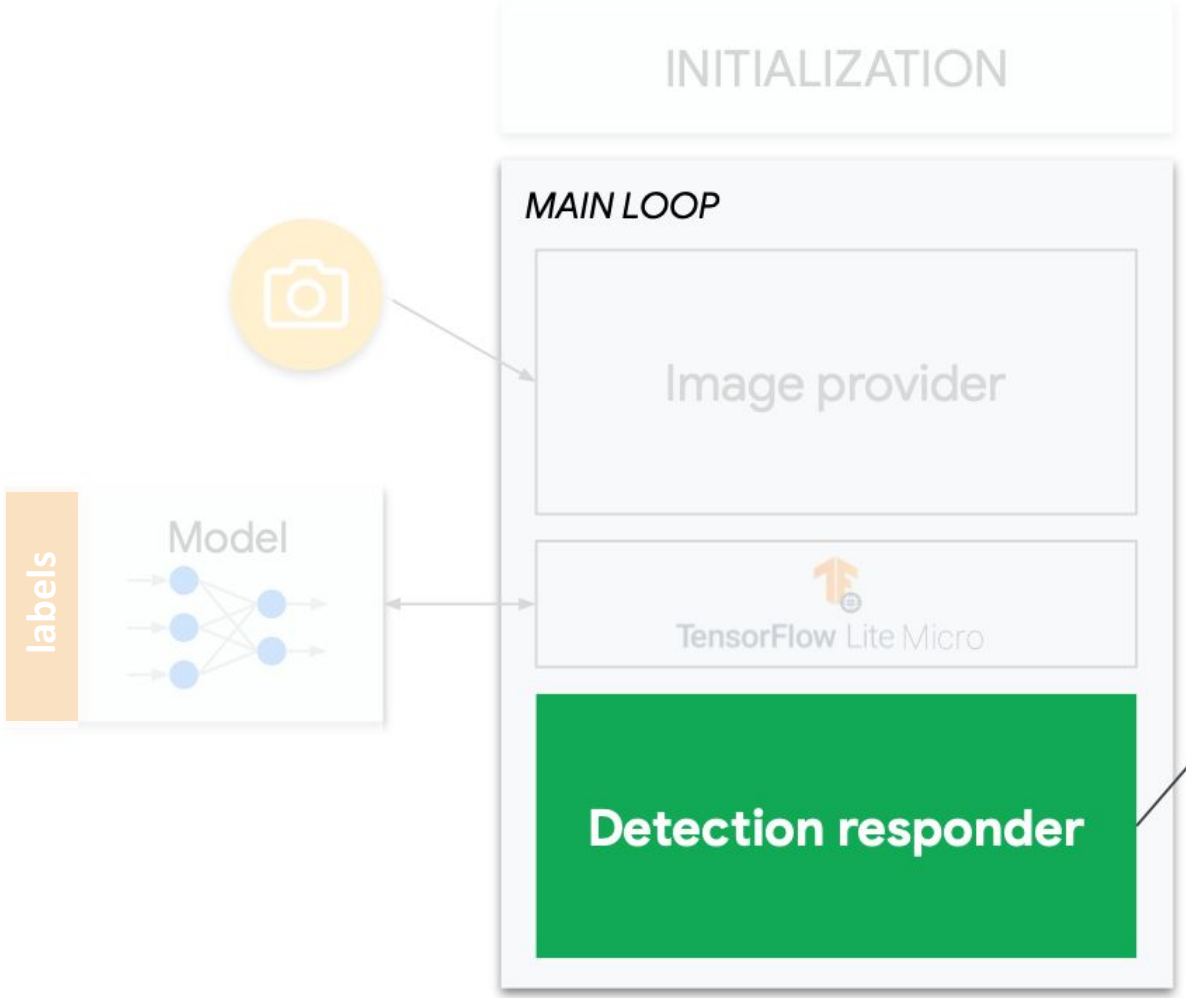


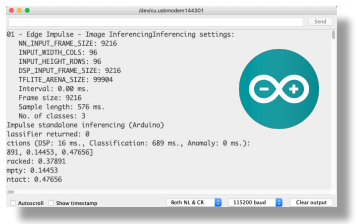
Interpreter + Model




```
// invoke the impulse
EI_IMPULSE_ERROR res = run_classifier(&signal, &result, false /* debug */);
ei_printf("run_classifier returned: %d\n", res);

if (res != 0) return;
```

Post-processing



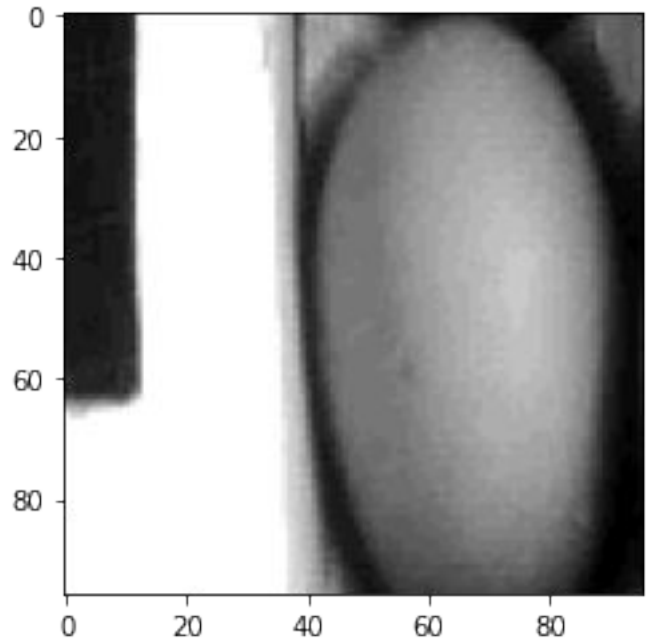


Detection responder

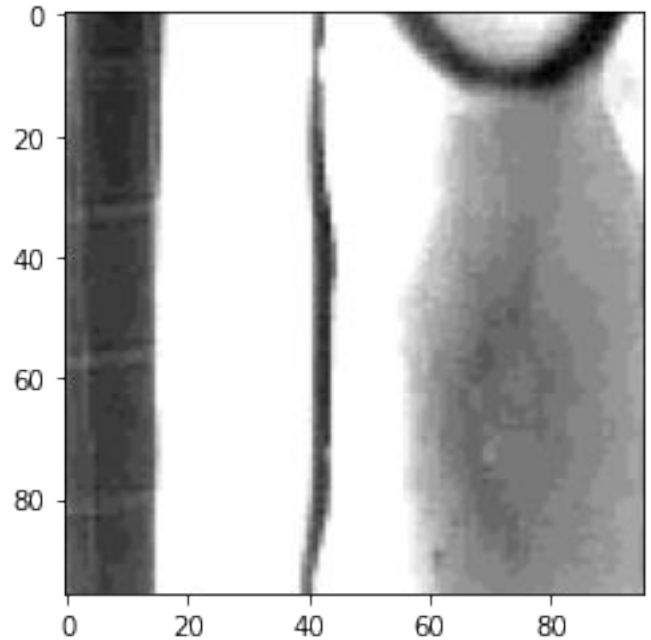
```
// print the predictions
ei_printf("Predictions ");
ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",
    result.timing.dsp, result.timing.classification, result.timing.anomaly);
ei_printf(": \n");
ei_printf("[");
for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
    ei_printf("%.5f", result.classification[ix].value);
```

```
/*
* The code portion bolow is used to show what image the camera is seem
* (for test only)
*/
     Serial.println();
     for (size_t ix = 0; ix < signal.total_length; ix++) {
         float value[1];
         signal.get_data(ix, 1, value);
         ei_printf("0x%06x", (int)value[0]);
         if (ix != signal.total_length - 1) {
             ei_printf(", ");
         }
     }
```

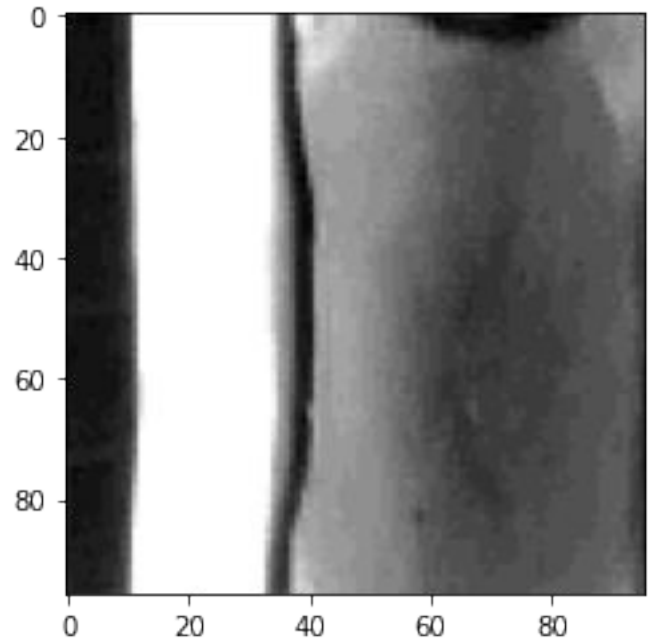
```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 16 ms., Classification: 690 ms., Anomaly: 0 ms.):
[0.42969, 0.35156, 0.21875]
    cracked: 0.42969
    empty: 0.35156
    intact: 0.21875

0x203838, 0x203838, 0x203c38, 0x203c30, 0x203c38, 0x203c38, 0x203430, 0x203830,
```



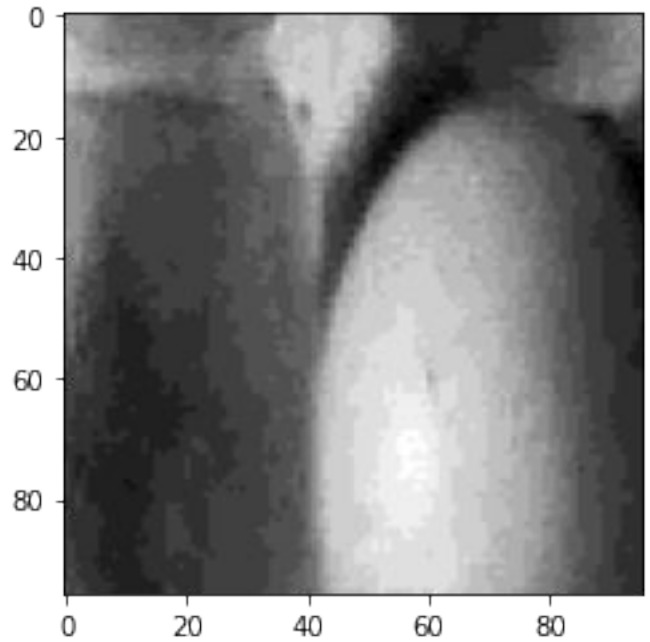
```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 16 ms., Classification: 690 ms., Anomaly: 0 ms.):
[0.22266, 0.39062, 0.38281]
    cracked: 0.22266
    empty: 0.39062
    intact: 0.38281

0xc0f0f0, 0xb0f0c8, 0x98d0c0, 0x90c8a8, 0x88bca0, 0x88b898, 0x80b490, 0x78b488,
```



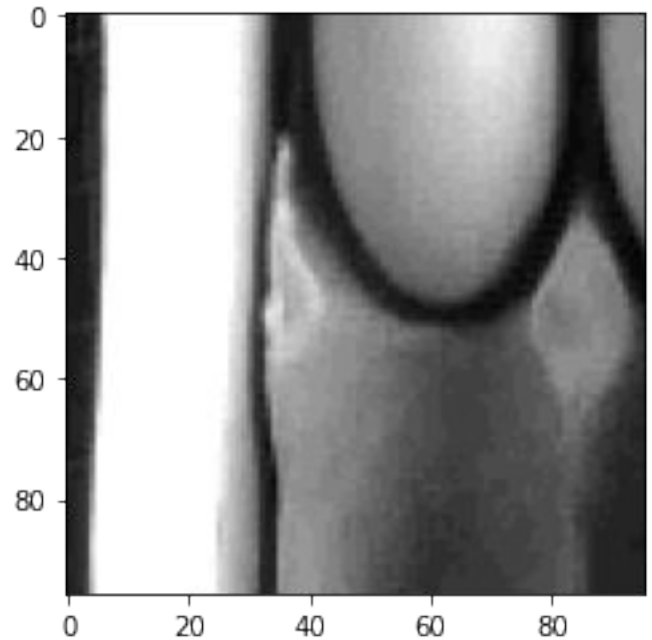
```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 16 ms., Classification: 690 ms., Anomaly: 0 ms.):
[0.30469, 0.45703, 0.23438]
    cracked: 0.30469
    empty: 0.45703
    intact: 0.23438

0x387048, 0x387048, 0x307048, 0x307048, 0x306c40, 0x306440, 0x386840, 0x386c40, (
```



```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 16 ms., Classification: 690 ms., Anomaly: 0 ms.):
[0.41797, 0.30469, 0.27734]
    cracked: 0.41797
    empty: 0.30469
    intact: 0.27734

0x709080, 0x709080, 0x709080, 0x709080, 0x709078, 0x688878, 0x688470, 0x608068,
```



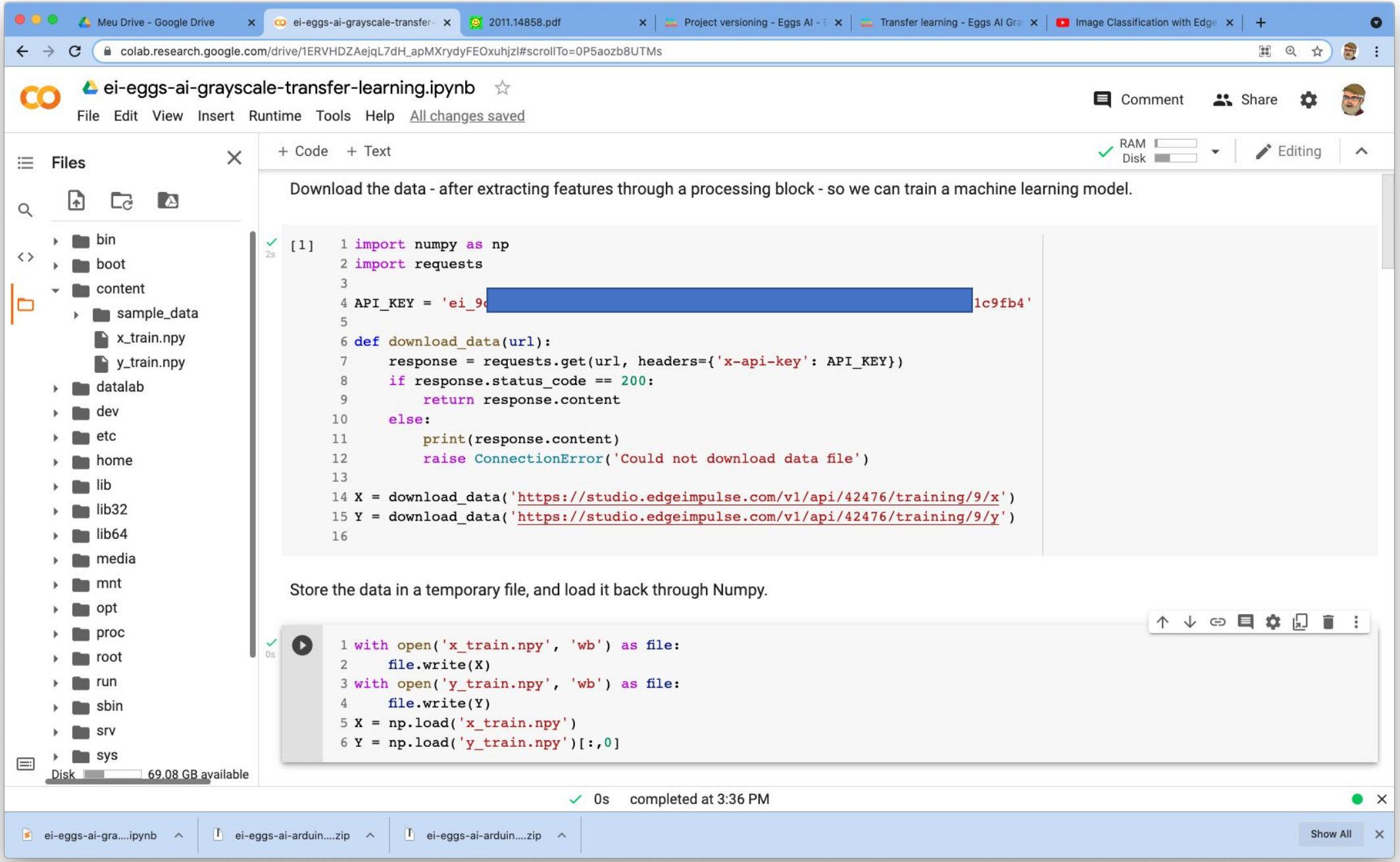
```
Edge Impulse standalone inferencing (Arduino)
run_classifier returned: 0
Predictions (DSP: 16 ms., Classification: 690 ms., Anomaly: 0 ms.):
[0.17969, 0.60156, 0.21875]
    cracked: 0.17969
    empty: 0.60156
    intact: 0.21875

0x305858, 0x285458, 0x407060, 0x386468, 0x385858, 0x485c60, 0xb8bc78, 0xf0f0f0,
```



Eggs Classification using Transfer Learning

Coding under the hood!

ei-eggs-ai-grayscale-transfer-learning.ipynb


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=0P5aozb8UTMs
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
bin
boot
content
sample_data
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
Disk 69.08 GB available
Download the data - after extracting features through a processing block - so we can train a machine learning model.
[1]
2s
import numpy as np
import requests
API_KEY = 'ei_9c [redacted] 1c9fb4'
def download_data(url):
response = requests.get(url, headers={'x-api-key': API_KEY})
if response.status_code == 200:
return response.content
else:
print(response.content)
raise ConnectionError('Could not download data file')
X = download_data('https://studio.edgeimpulse.com/v1/api/42476/training/9/x')
Y = download_data('https://studio.edgeimpulse.com/v1/api/42476/training/9/y')
Store the data in a temporary file, and load it back through Numpy.
0s
with open('x_train.npy', 'wb') as file:
file.write(X)
with open('y_train.npy', 'wb') as file:
file.write(Y)
X = np.load('x_train.npy')
Y = np.load('y_train.npy')[:,0]
0s completed at 3:36 PM
ei-eggs-ai-gra....ipynb
ei-eggs-ai-arduin....zip
ei-eggs-ai-arduin....zip
Show All

Meu Drive - Google Drive | ei-eggs-ai-grayscale-transfer- | 2011.14858.pdf | Project versioning - Eggs AI - E | Transfer learning - Eggs AI Gra | Image Classification with Edge

colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=JfFvaQGbWBsc

ei-eggs-ai-grayscale-transfer-learning.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

bin
boot
content
sample_data
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tensorflow-1.15.2
tmp

Disk 69.08 GB available

+ Code + Text

RAM Disk

Editing

```
[3]  1 X.shape, Y.shape
```

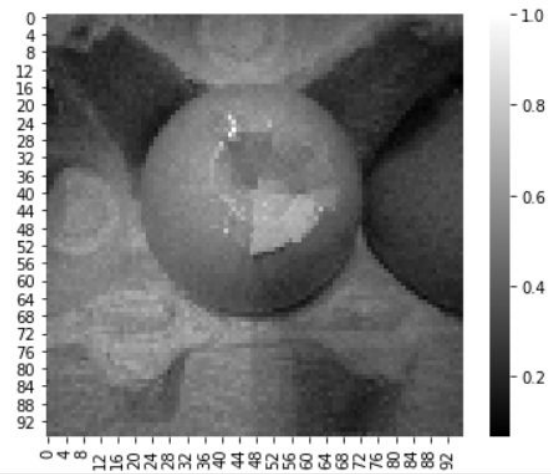
```
((107, 9216), (107,))
```

```
[4]  1 set(Y)
```

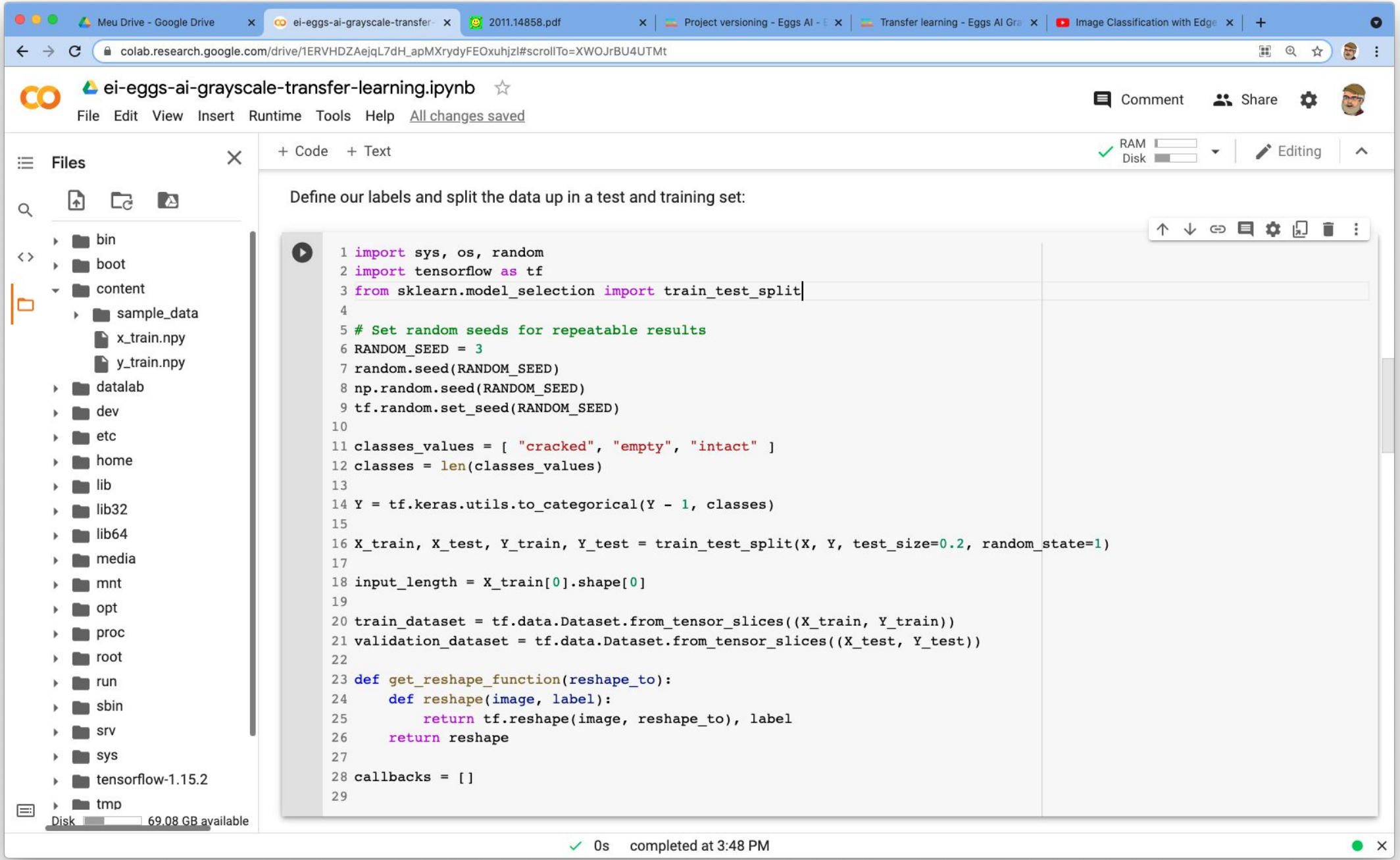
```
{1, 2, 3}
```

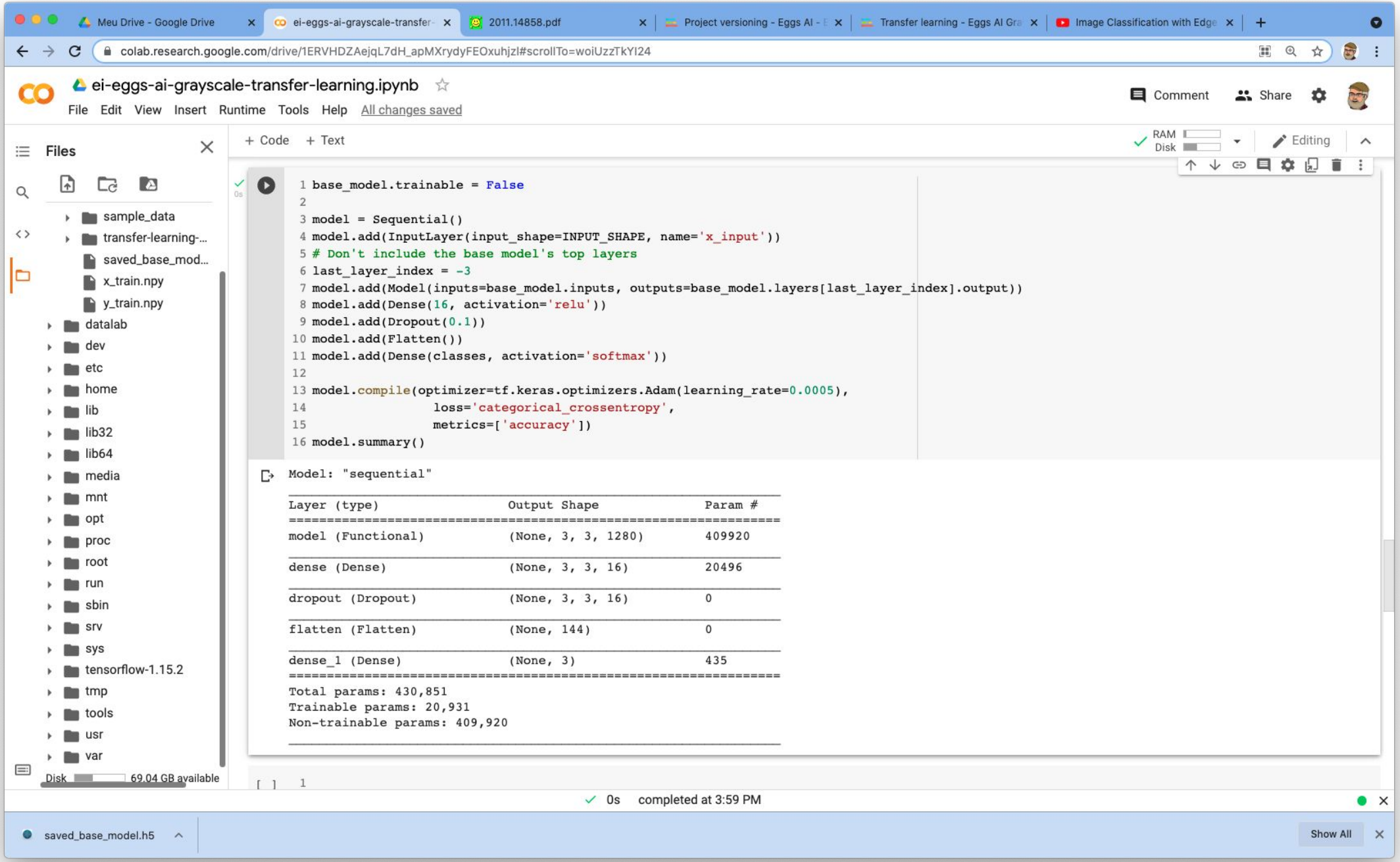
```
[20] 1 import matplotlib.pyplot as plt
     2 import seaborn as sns
```

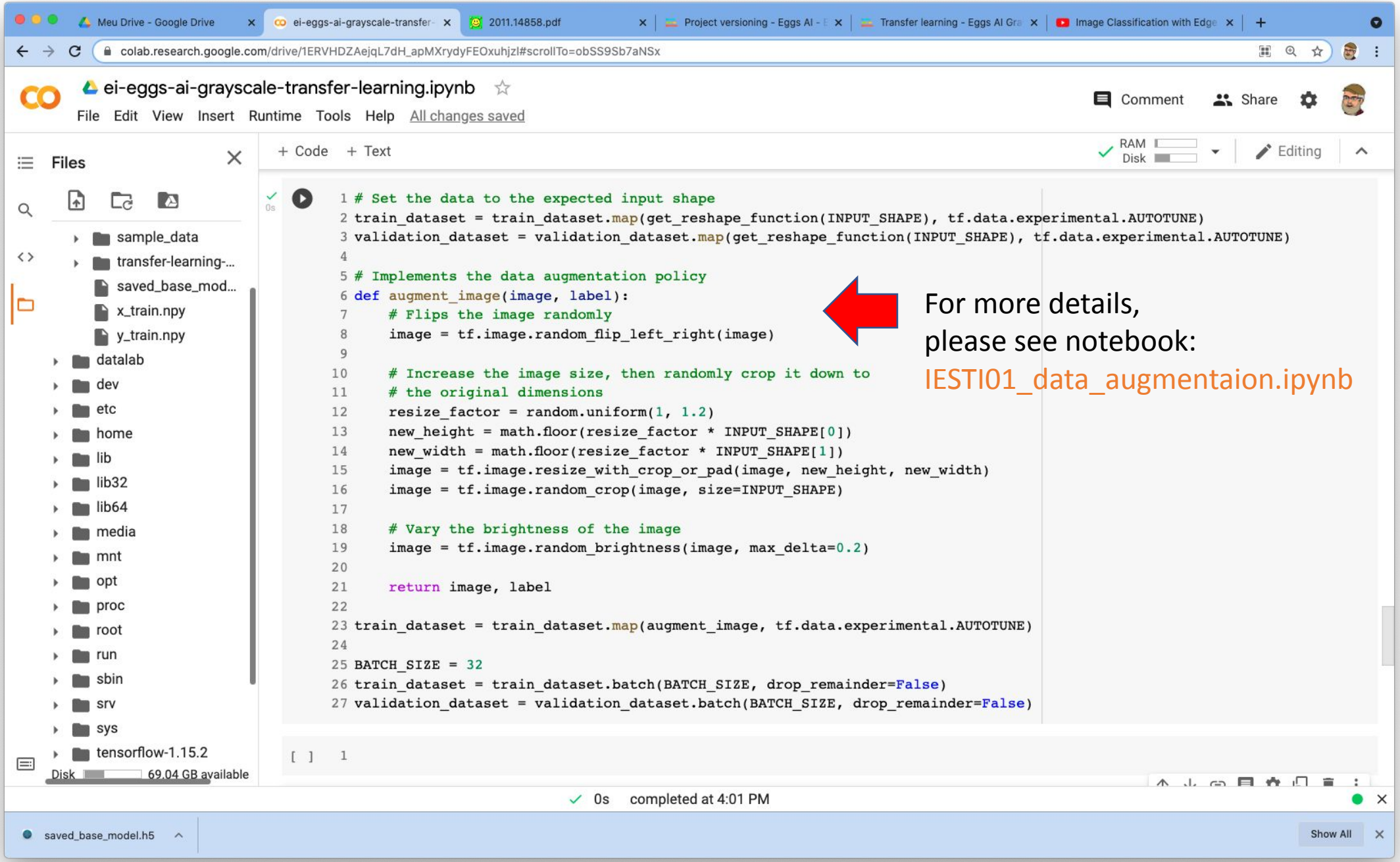
```
1 image = X[90]
2 image = np.reshape(image, (96, 96))
3 plt.figure(figsize=(6,5));
4 sns.heatmap(image, cmap='gray');
```

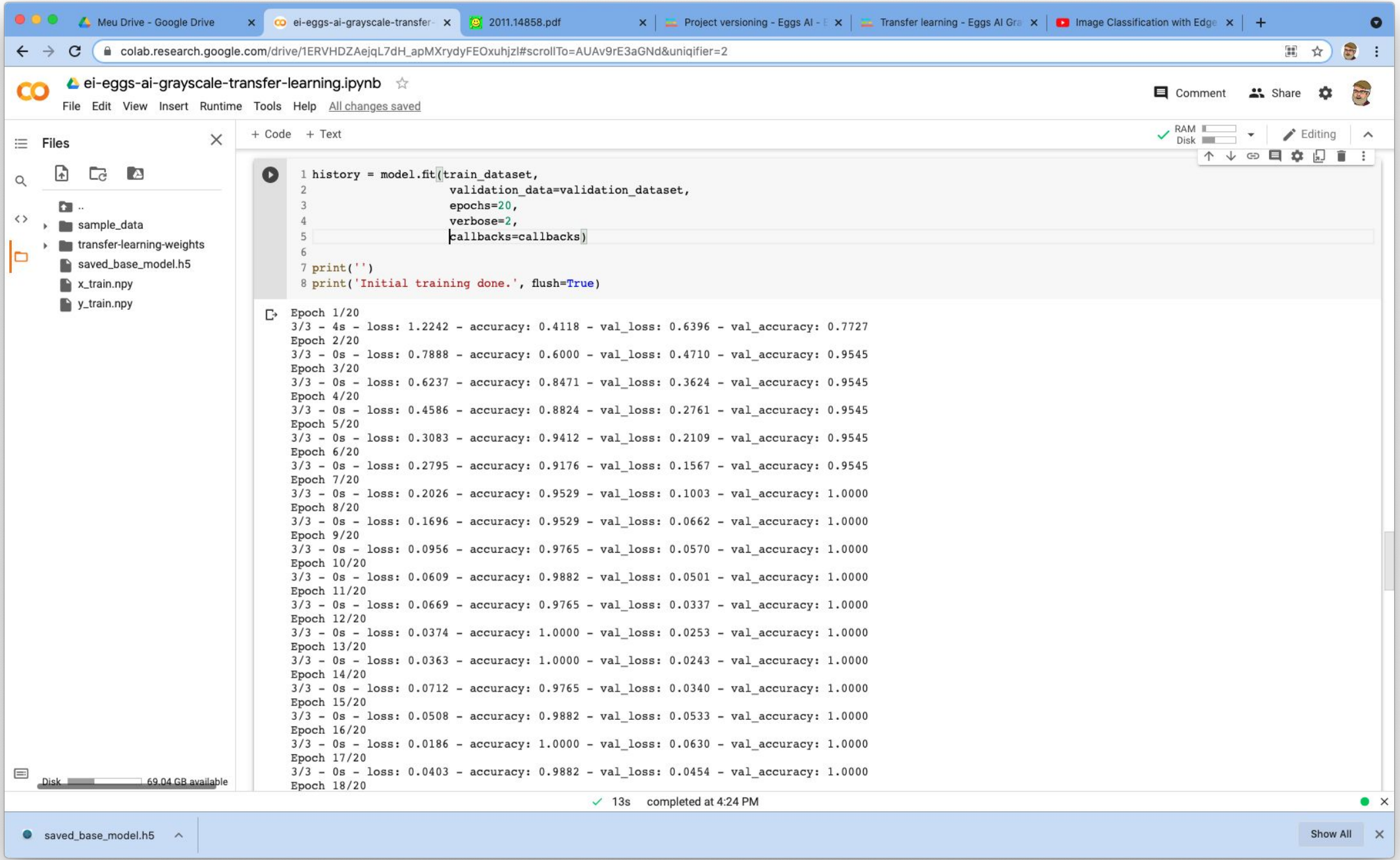


0s completed at 3:48 PM


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=XWOJrBU4UTMt
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
bin
boot
content
sample_data
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tensorflow-1.15.2
tmp
Disk 69.08 GB available
Define our labels and split the data up in a test and training set:
1 import sys, os, random
2 import tensorflow as tf
3 from sklearn.model_selection import train_test_split
4
5 # Set random seeds for repeatable results
6 RANDOM_SEED = 3
7 random.seed(RANDOM_SEED)
8 np.random.seed(RANDOM_SEED)
9 tf.random.set_seed(RANDOM_SEED)
10
11 classes_values = ["cracked", "empty", "intact"]
12 classes = len(classes_values)
13
14 Y = tf.keras.utils.to_categorical(Y - 1, classes)
15
16 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=1)
17
18 input_length = X_train[0].shape[0]
19
20 train_dataset = tf.data.Dataset.from_tensor_slices((X_train, Y_train))
21 validation_dataset = tf.data.Dataset.from_tensor_slices((X_test, Y_test))
22
23 def get_reshape_function(reshape_to):
24   def reshape(image, label):
25   return tf.reshape(image, reshape_to), label
26   return reshape
27
28 callbacks = []
29
0s completed at 3:48 PM


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=woiUzzTkYI24
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
transfer-learning-...
saved_base_mod...
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tensorflow-1.15.2
tmp
tools
usr
var
Disk 69.04 GB available
+ Code + Text
RAM
Disk
Editing
1 base_model.trainable = False
2
3 model = Sequential()
4 model.add(InputLayer(input_shape=INPUT_SHAPE, name='x_input'))
5 # Don't include the base model's top layers
6 last_layer_index = -3
7 model.add(Model(inputs=base_model.inputs, outputs=base_model.layers[last_layer_index].output))
8 model.add(Dense(16, activation='relu'))
9 model.add(Dropout(0.1))
10 model.add(Flatten())
11 model.add(Dense(classes, activation='softmax'))
12
13 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
14 loss='categorical_crossentropy',
15 metrics=['accuracy'])
16 model.summary()
Model: "sequential"
Layer (type) Output Shape Param #
model (Functional) (None, 3, 3, 1280) 409920
dense (Dense) (None, 3, 3, 16) 20496
dropout (Dropout) (None, 3, 3, 16) 0
flatten (Flatten) (None, 144) 0
dense_1 (Dense) (None, 3) 435
Total params: 430,851
Trainable params: 20,931
Non-trainable params: 409,920
0s completed at 3:59 PM
saved_base_model.h5
Show All


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=obSS9Sb7aNSx
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
Editing
sample_data
transfer-learning-...
saved_base_mod...
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tensorflow-1.15.2
Disk 69.04 GB available
1 # Set the data to the expected input shape
2 train_dataset = train_dataset.map(get_reshape_function(INPUT_SHAPE), tf.data.experimental.AUTOTUNE)
3 validation_dataset = validation_dataset.map(get_reshape_function(INPUT_SHAPE), tf.data.experimental.AUTOTUNE)
4
5 # Implements the data augmentation policy
6 def augment_image(image, label):
7 # Flips the image randomly
8 image = tf.image.random_flip_left_right(image)
9
10 # Increase the image size, then randomly crop it down to
11 # the original dimensions
12 resize_factor = random.uniform(1, 1.2)
13 new_height = math.floor(resize_factor * INPUT_SHAPE[0])
14 new_width = math.floor(resize_factor * INPUT_SHAPE[1])
15 image = tf.image.resize_with_crop_or_pad(image, new_height, new_width)
16 image = tf.image.random_crop(image, size=INPUT_SHAPE)
17
18 # Vary the brightness of the image
19 image = tf.image.random_brightness(image, max_delta=0.2)
20
21 return image, label
22
23 train_dataset = train_dataset.map(augment_image, tf.data.experimental.AUTOTUNE)
24
25 BATCH_SIZE = 32
26 train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
27 validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)
[ ] 1
For more details,
please see notebook:
IESTI01_data_augmentaion.ipynb
0s completed at 4:01 PM
saved_base_model.h5
Show All


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=AUAv9rE3aGNd&uniqifier=2
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
..
sample_data
transfer-learning-weights
saved_base_model.h5
x_train.npy
y_train.npy
1 history = model.fit(train_dataset,
2 validation_data=validation_dataset,
3 epochs=20,
4 verbose=2,
5 callbacks=callbacks)
7 print('')
8 print('Initial training done.', flush=True)
Epoch 1/20
3/3 - 4s - loss: 1.2242 - accuracy: 0.4118 - val_loss: 0.6396 - val_accuracy: 0.7727
Epoch 2/20
3/3 - 0s - loss: 0.7888 - accuracy: 0.6000 - val_loss: 0.4710 - val_accuracy: 0.9545
Epoch 3/20
3/3 - 0s - loss: 0.6237 - accuracy: 0.8471 - val_loss: 0.3624 - val_accuracy: 0.9545
Epoch 4/20
3/3 - 0s - loss: 0.4586 - accuracy: 0.8824 - val_loss: 0.2761 - val_accuracy: 0.9545
Epoch 5/20
3/3 - 0s - loss: 0.3083 - accuracy: 0.9412 - val_loss: 0.2109 - val_accuracy: 0.9545
Epoch 6/20
3/3 - 0s - loss: 0.2795 - accuracy: 0.9176 - val_loss: 0.1567 - val_accuracy: 0.9545
Epoch 7/20
3/3 - 0s - loss: 0.2026 - accuracy: 0.9529 - val_loss: 0.1003 - val_accuracy: 1.0000
Epoch 8/20
3/3 - 0s - loss: 0.1696 - accuracy: 0.9529 - val_loss: 0.0662 - val_accuracy: 1.0000
Epoch 9/20
3/3 - 0s - loss: 0.0956 - accuracy: 0.9765 - val_loss: 0.0570 - val_accuracy: 1.0000
Epoch 10/20
3/3 - 0s - loss: 0.0609 - accuracy: 0.9882 - val_loss: 0.0501 - val_accuracy: 1.0000
Epoch 11/20
3/3 - 0s - loss: 0.0669 - accuracy: 0.9765 - val_loss: 0.0337 - val_accuracy: 1.0000
Epoch 12/20
3/3 - 0s - loss: 0.0374 - accuracy: 1.0000 - val_loss: 0.0253 - val_accuracy: 1.0000
Epoch 13/20
3/3 - 0s - loss: 0.0363 - accuracy: 1.0000 - val_loss: 0.0243 - val_accuracy: 1.0000
Epoch 14/20
3/3 - 0s - loss: 0.0712 - accuracy: 0.9765 - val_loss: 0.0340 - val_accuracy: 1.0000
Epoch 15/20
3/3 - 0s - loss: 0.0508 - accuracy: 0.9882 - val_loss: 0.0533 - val_accuracy: 1.0000
Epoch 16/20
3/3 - 0s - loss: 0.0186 - accuracy: 1.0000 - val_loss: 0.0630 - val_accuracy: 1.0000
Epoch 17/20
3/3 - 0s - loss: 0.0403 - accuracy: 0.9882 - val_loss: 0.0454 - val_accuracy: 1.0000
Epoch 18/20
Disk 69.04 GB available
13s completed at 4:24 PM
saved_base_model.h5
Show All

Meu Drive - Google Drive | ei-eggs-ai-grayscale-transfer- | 2011.14858.pdf | Project versioning - Eggs AI - E | Transfer learning - Eggs AI Gra | Image Classification with Edge

colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=AUAv9rE3aGNd&uniqifier=2

ei-eggs-ai-grayscale-transfer-learning.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

.. sample_data transfer-learning-weights saved_base_model.h5 x_train.npy y_train.npy

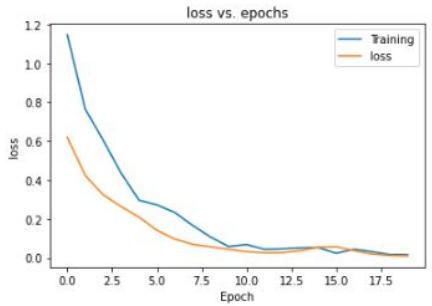
Disk 69.04 GB available

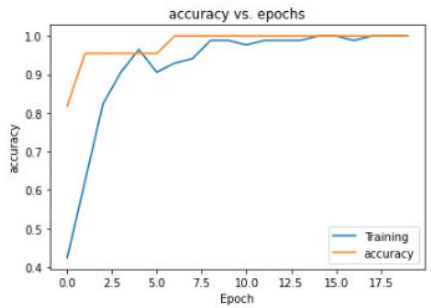
+ Code + Text

RAM Disk Editing

```
[60] 1 def plot_result(history, res= 'loss'):
     2     plt.plot(history.history[res])
     3     plt.plot(history.history['val_'+res])
     4     plt.title(res+' vs. epochs')
     5     plt.ylabel(res)
     6     plt.xlabel('Epoch')
     7     if res == 'loss':
     8         plt.legend(['Training', res], loc='upper right')
     9     else:
    10         plt.legend(['Training', res], loc='lower right')
    11     plt.show()
```

```
1 plot_result(history, res= 'loss')
2 plot_result(history, res= 'accuracy')
```

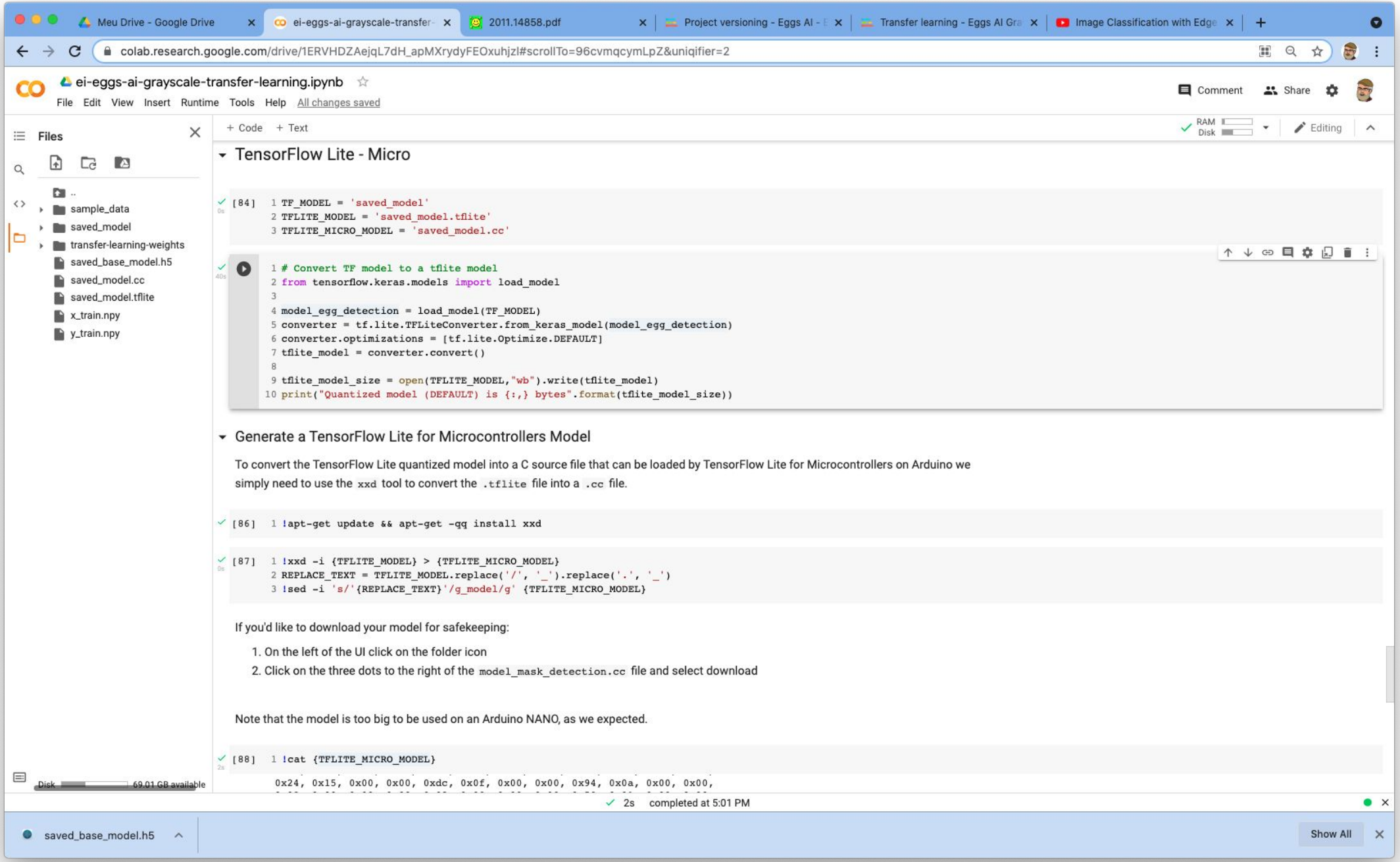


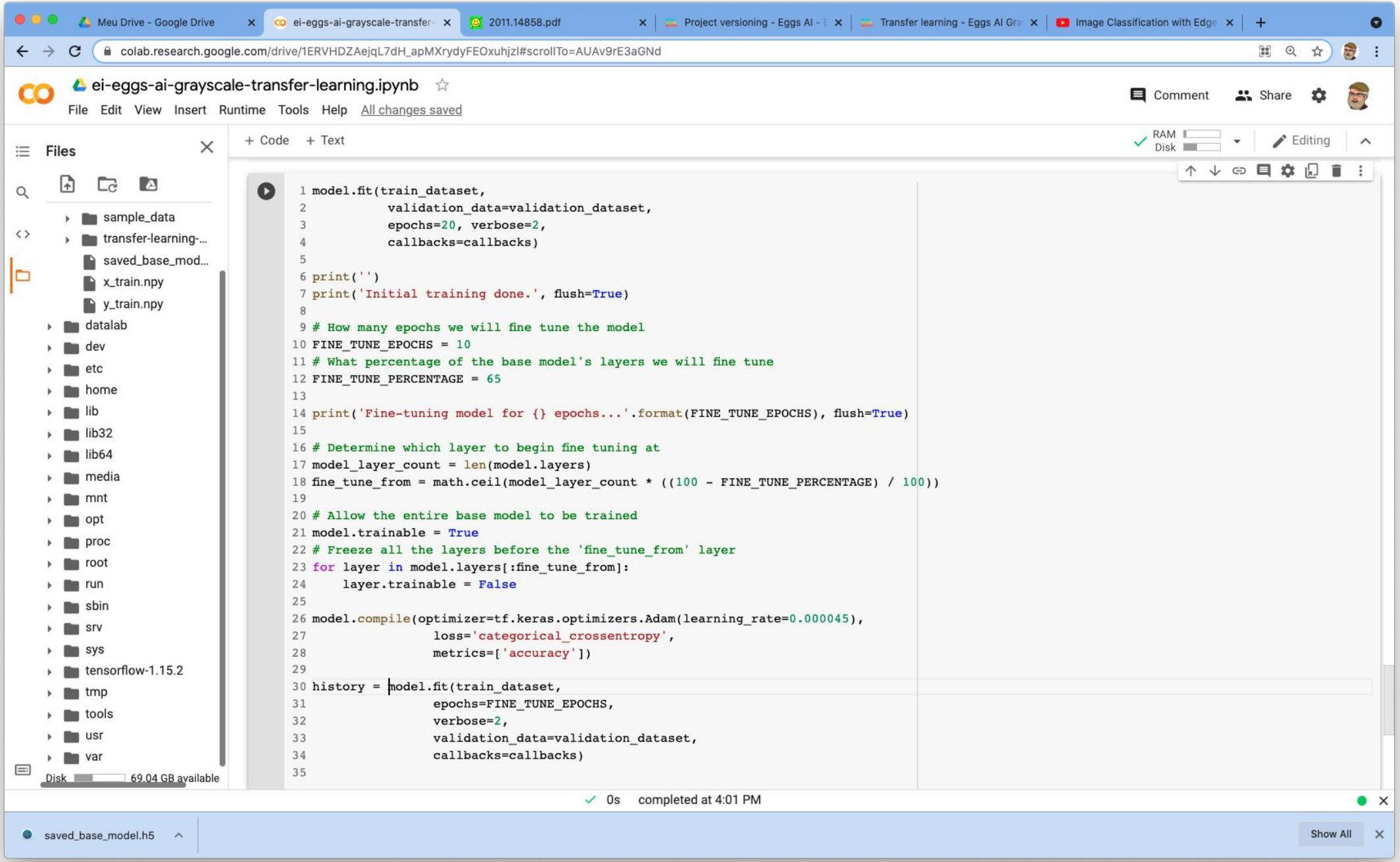


13s completed at 4:24 PM

saved_base_model.h5

Show All


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzl#scrollTo=96cvmqcymLpZ&uniqifier=2
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
saved_model
transfer-learning-weights
saved_base_model.h5
saved_model.cc
saved_model.tflite
x_train.npy
y_train.npy
Disk 69.01 GB available
+ Code + Text
RAM
Disk
Editing
TensorFlow Lite - Micro
[84] 1 TF_MODEL = 'saved_model'
2 TFLITE_MODEL = 'saved_model.tflite'
3 TFLITE_MICRO_MODEL = 'saved_model.cc'
1 # Convert TF model to a tflite model
2 from tensorflow.keras.models import load_model
3
4 model_egg_detection = load_model(TF_MODEL)
5 converter = tf.lite.TFLiteConverter.from_keras_model(model_egg_detection)
6 converter.optimizations = [tf.lite.Optimize.DEFAULT]
7 tflite_model = converter.convert()
8
9 tflite_model_size = open(TFLITE_MODEL,"wb").write(tflite_model)
10 print("Quantized model (DEFAULT) is {:,} bytes".format(tflite_model_size))
Generate a TensorFlow Lite for Microcontrollers Model
To convert the TensorFlow Lite quantized model into a C source file that can be loaded by TensorFlow Lite for Microcontrollers on Arduino we simply need to use the xxd tool to convert the .tflite file into a .cc file.
[86] 1 !apt-get update && apt-get -qq install xxd
[87] 1 !xxd -i {TFLITE_MODEL} > {TFLITE_MICRO_MODEL}
2 REPLACE_TEXT = TFLITE_MODEL.replace('/', '_').replace('.', '_')
3 !sed -i 's/'{REPLACE_TEXT}'/g_model/g' {TFLITE_MICRO_MODEL}
If you'd like to download your model for safekeeping:
1. On the left of the UI click on the folder icon
2. Click on the three dots to the right of the model_mask_detection.cc file and select download
Note that the model is too big to be used on an Arduino NANO, as we expected.
[88] 1 !cat {TFLITE_MICRO_MODEL}
0x24, 0x15, 0x00, 0x00, 0xdc, 0x0f, 0x00, 0x00, 0x94, 0x0a, 0x00, 0x00,
2s completed at 5:01 PM
saved_base_model.h5
Show All


Meu Drive - Google Drive
ei-eggs-ai-grayscale-transfer-
2011.14858.pdf
Project versioning - Eggs AI -
Transfer learning - Eggs AI Gra
Image Classification with Edge
colab.research.google.com/drive/1ERVHDZAejqL7dH_apMXrydyFEOxuhjzI#scrollTo=AUAv9rE3aGNd
ei-eggs-ai-grayscale-transfer-learning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
Editing
sample_data
transfer-learning-...
saved_base_mod...
x_train.npy
y_train.npy
datalab
dev
etc
home
lib
lib32
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tensorflow-1.15.2
tmp
tools
usr
var
Disk 69.04 GB available
1 model.fit(train_dataset,
2 validation_data=validation_dataset,
3 epochs=20, verbose=2,
4 callbacks=callbacks)
5
6 print('')
7 print('Initial training done.', flush=True)
8
9 # How many epochs we will fine tune the model
10 FINE_TUNE_EPOCHS = 10
11 # What percentage of the base model's layers we will fine tune
12 FINE_TUNE_PERCENTAGE = 65
13
14 print('Fine-tuning model for {} epochs...'.format(FINE_TUNE_EPOCHS), flush=True)
15
16 # Determine which layer to begin fine tuning at
17 model_layer_count = len(model.layers)
18 fine_tune_from = math.ceil(model_layer_count * ((100 - FINE_TUNE_PERCENTAGE) / 100))
19
20 # Allow the entire base model to be trained
21 model.trainable = True
22 # Freeze all the layers before the 'fine_tune_from' layer
23 for layer in model.layers[:fine_tune_from]:
24 layer.trainable = False
25
26 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.000045),
27 loss='categorical_crossentropy',
28 metrics=['accuracy'])
29
30 history = model.fit(train_dataset,
31 epochs=FINE_TUNE_EPOCHS,
32 verbose=2,
33 validation_data=validation_dataset,
34 callbacks=callbacks)
35
0s completed at 4:01 PM
saved_base_model.h5
Show All

Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning (Coursera)
- Text Book: "TinyML" by Pete Warden, Daniel Situnayake

I want to thank Shawn Hymel and Edge Impulse, Pete Warden and Laurence Moroney from Google, and especially Harvard professor Vijay Janapa Reddi, Ph.D. student Brian Plancher and their staff for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the TinyML4D, an initiative to make TinyML education available to everyone globally.

Thanks

And stay safe!

UNIFEI