ESP32-S3 STICK with TinyML

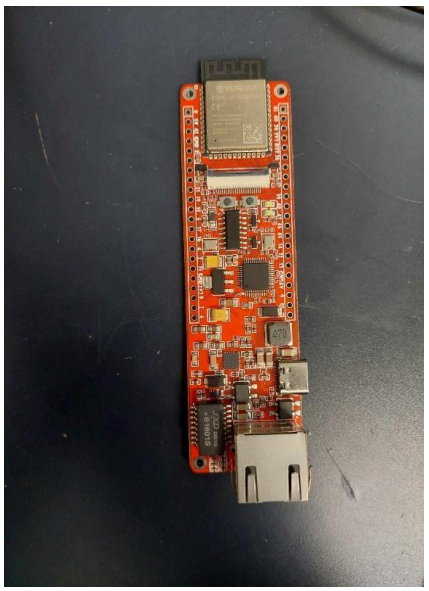
INTRODUCTION:

Hello everyone! In this tutorial, I will demonstrate how to utilize our pre-trained machine learning model with custom data. Specifically, we will be using Edge Impulse to detect the colors. In our previous tutorial, we relied on a PC and a Python script for face detection. However, in this tutorial, we'll shift to using S3 SoC instead of a PC.

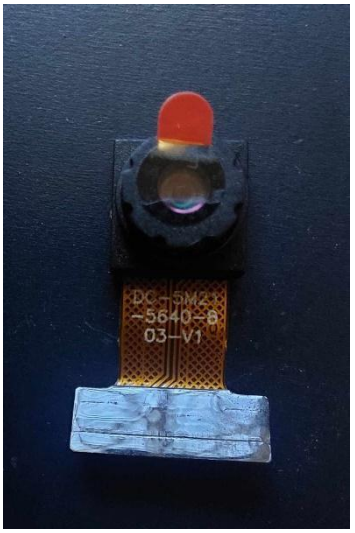
List of hardware you need:

- 1x USB-C cable
- ESP32-S3 stick
- Camera Ov2640 or Ov5640

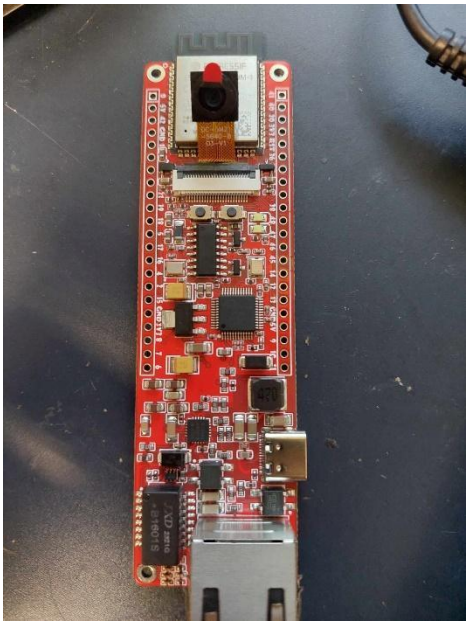
ESP32 STICK



Ov5640 camera



Connected camera to stick

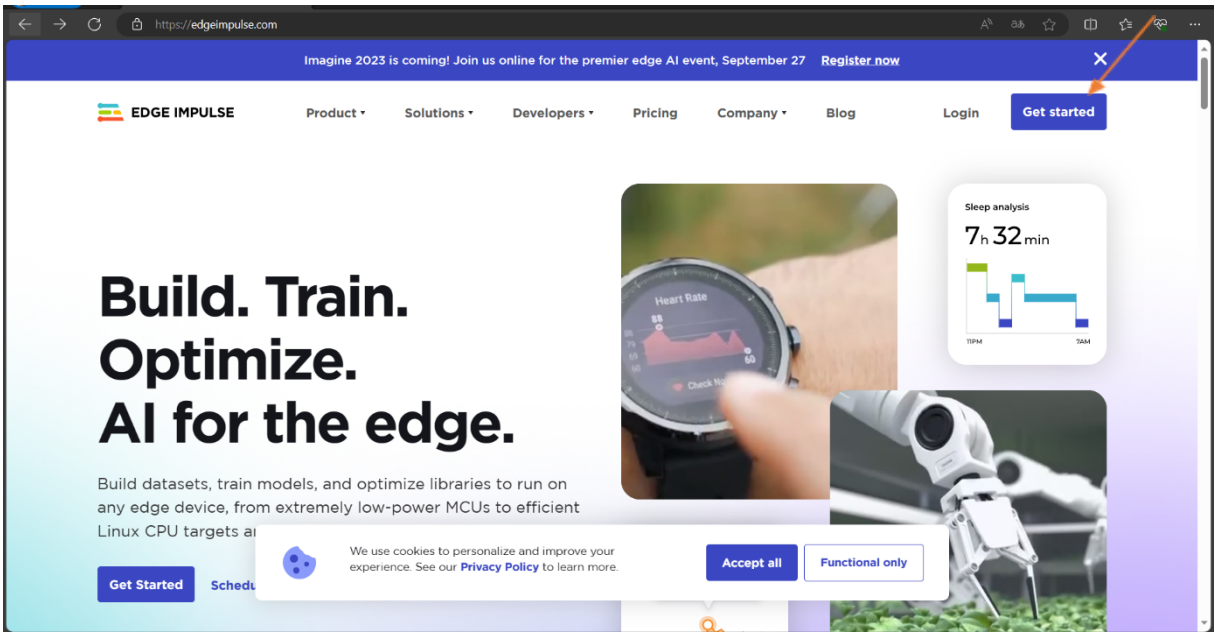


1. Software

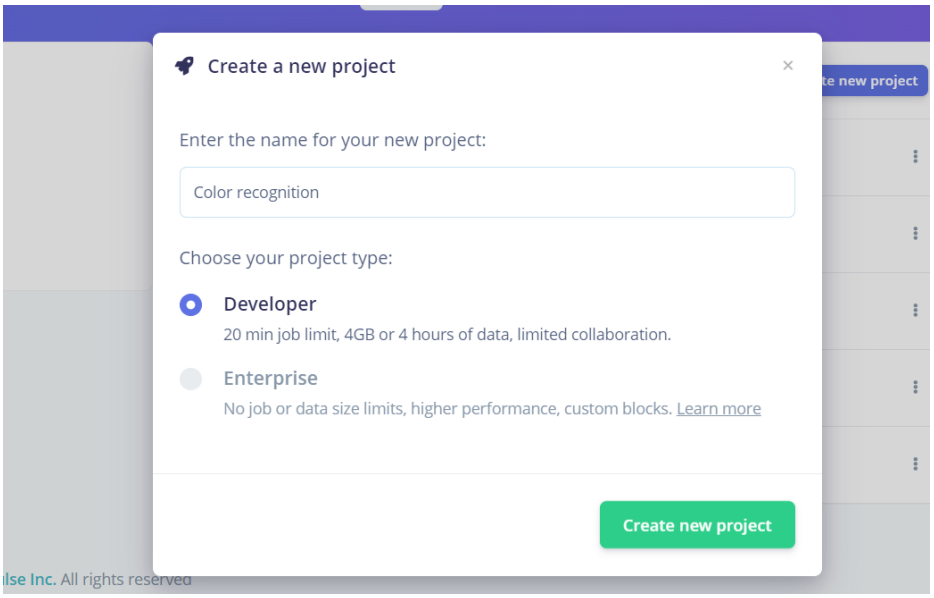
To set up TinyML, we'll be using the Arduino IDE. While we could use VScode as we did in the previous tutorial, Arduino IDE offers a simpler approach because it allows us to include libraries in the form of ZIP files. Additionally, please note that for our pre-trained model, you'll need to register on the Edge Impulse website.

2. Registration

On Edge Impulse website click on button ‘Get started’ and register

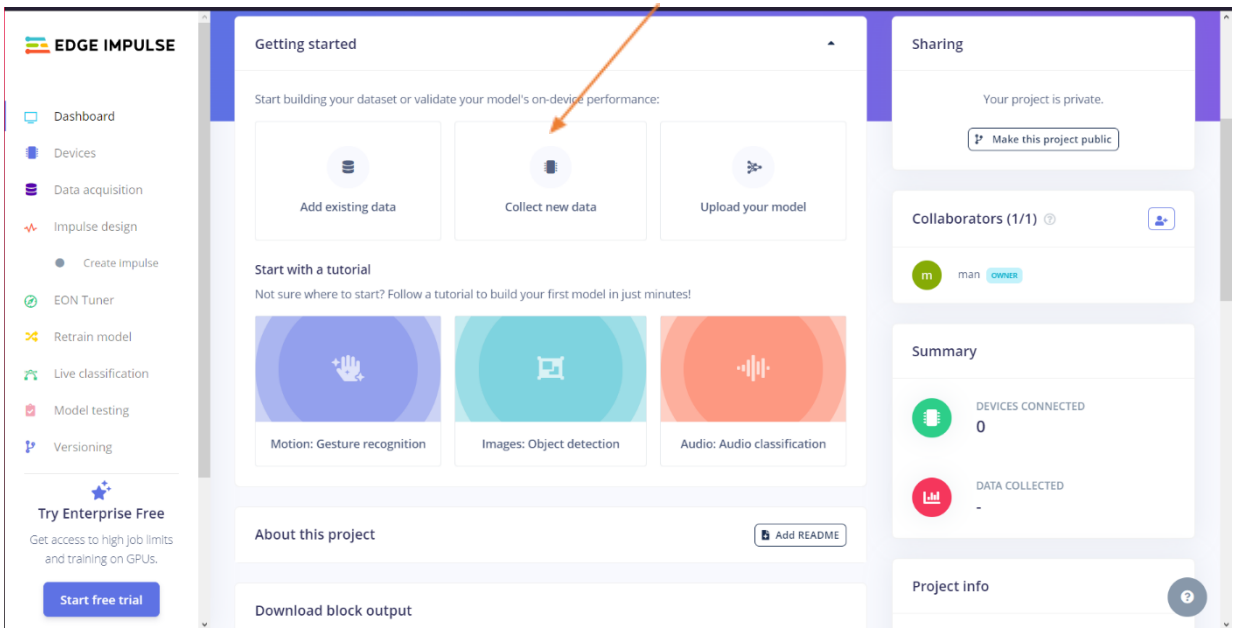


Create a project, enter the name of a project for me it’s ‘color recognition’

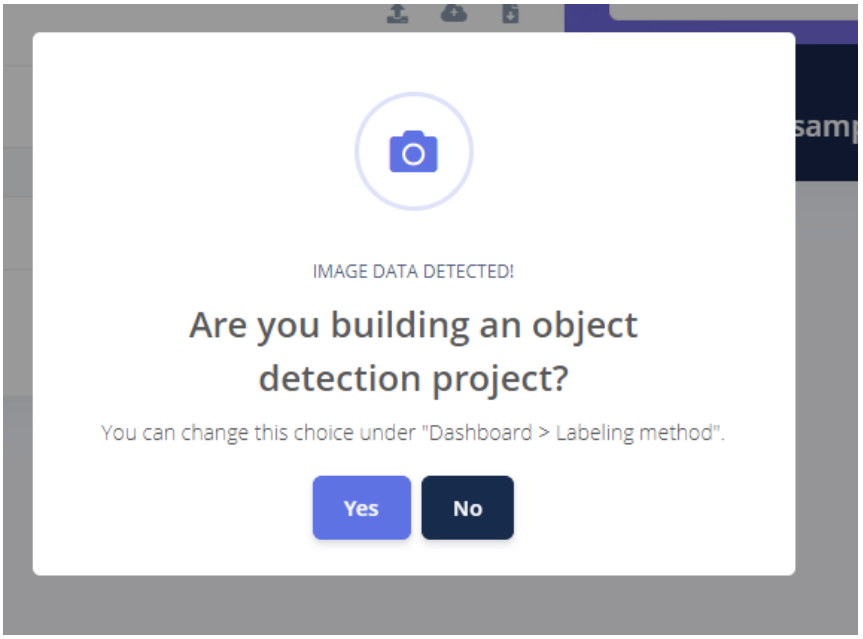


3. Uploading data

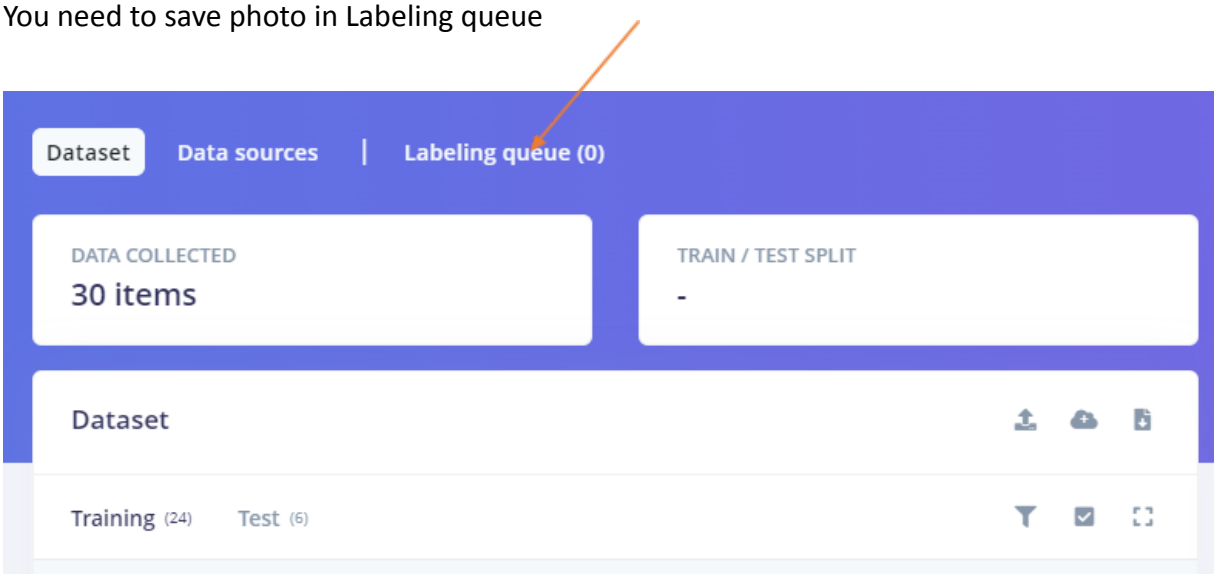
After creating project you should be on dashboard, click on ‘Collect new data’, if you see window with collecting data by computer, phone or board close it, but if you want to you can. Click on ‘add data’ and ‘upload data’.



If you see this while uploading data click ‘No’

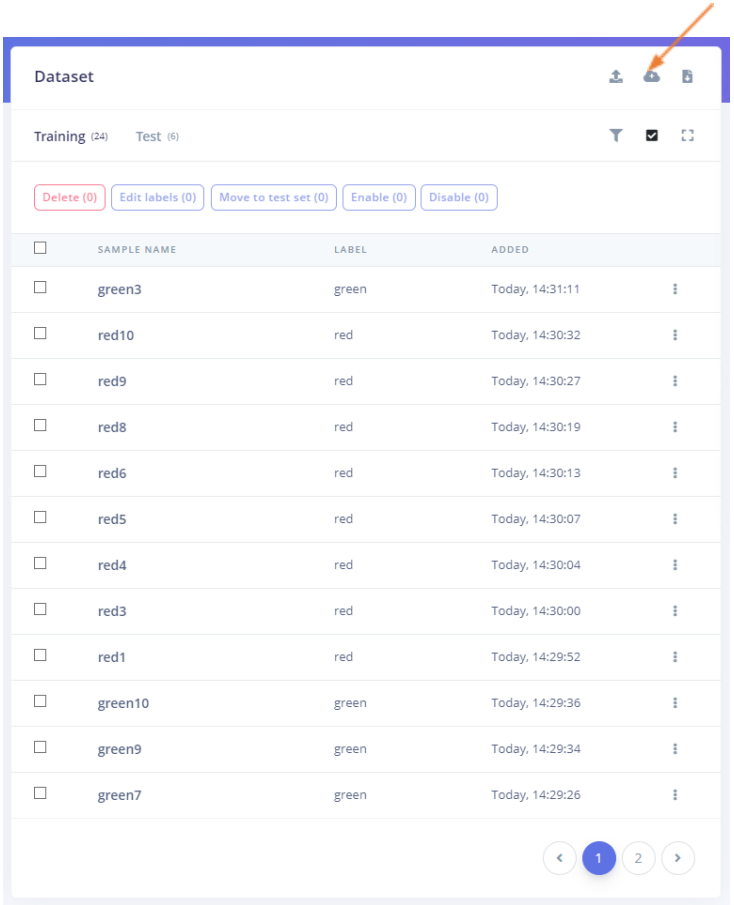


You need to save photo in Labeling queue



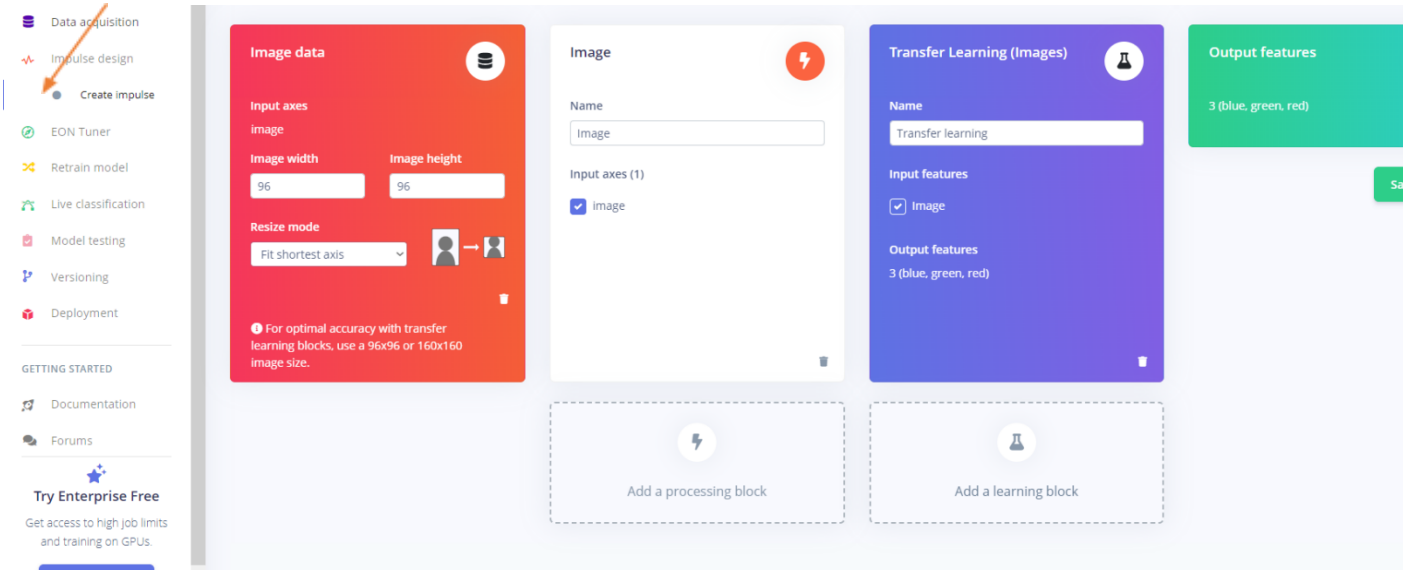
4. Distribution

Once you have uploaded your data, click on the icon that allows you to select multiple items. From there, you can edit the names of the labels, such as "red," "green," and "blue." If you wish to create a testing dataset, you can move two photos of each color into the testing dataset.

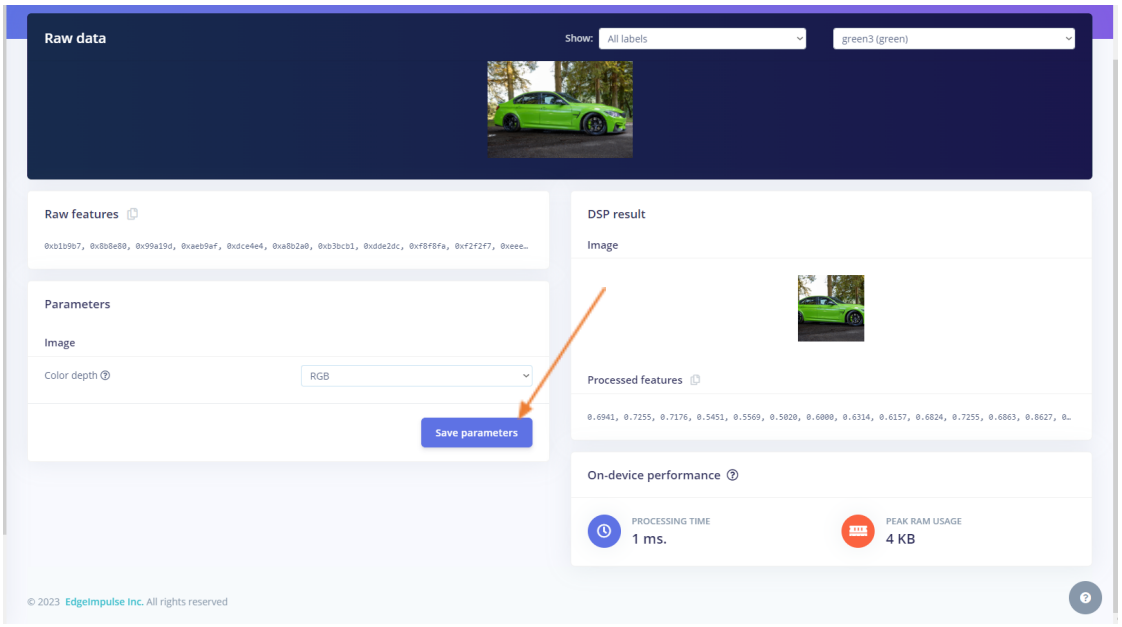


5. Training model

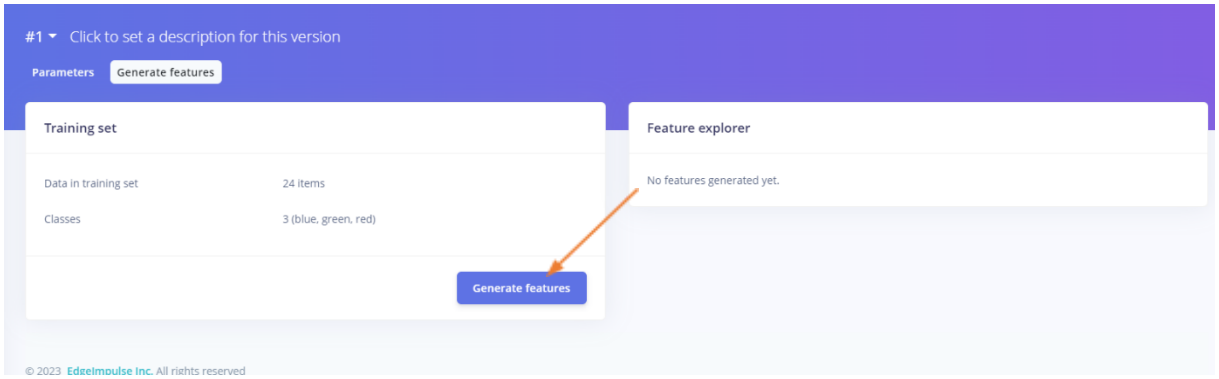
Next on sidebar click on ‘Create impulse’



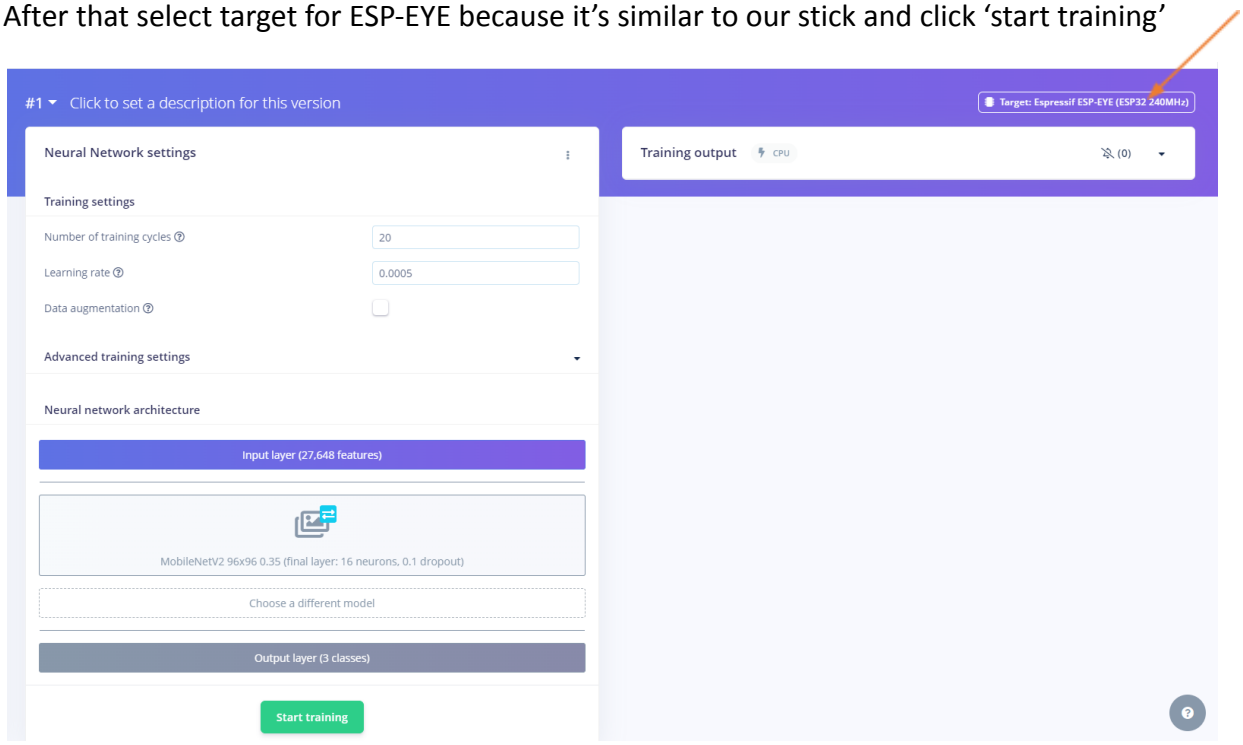
Save parameters, color RGB



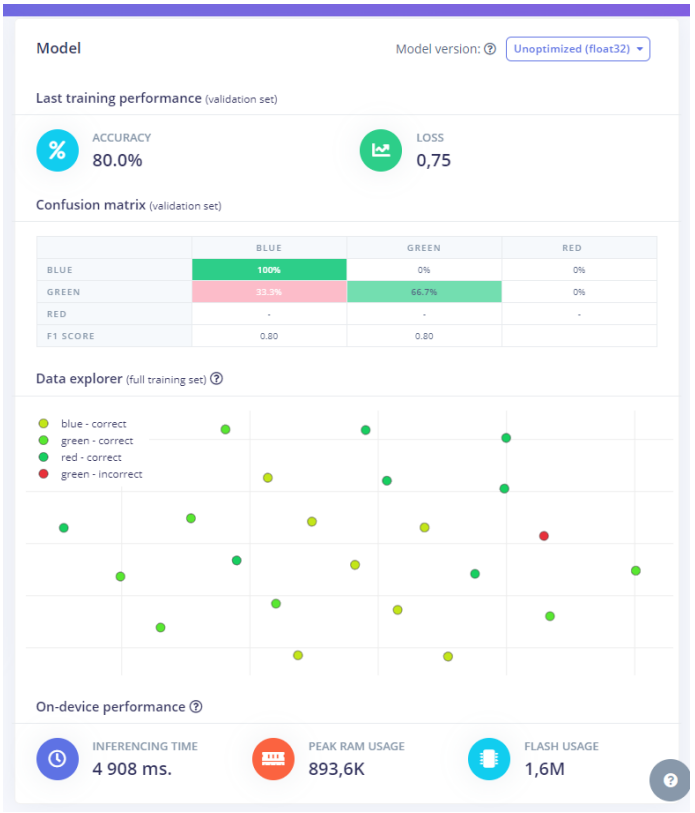
Generate features, and after generating click on sidebar ‘Transfer learning’



After that select target for ESP-EYE because it’s similar to our stick and click ‘start training’

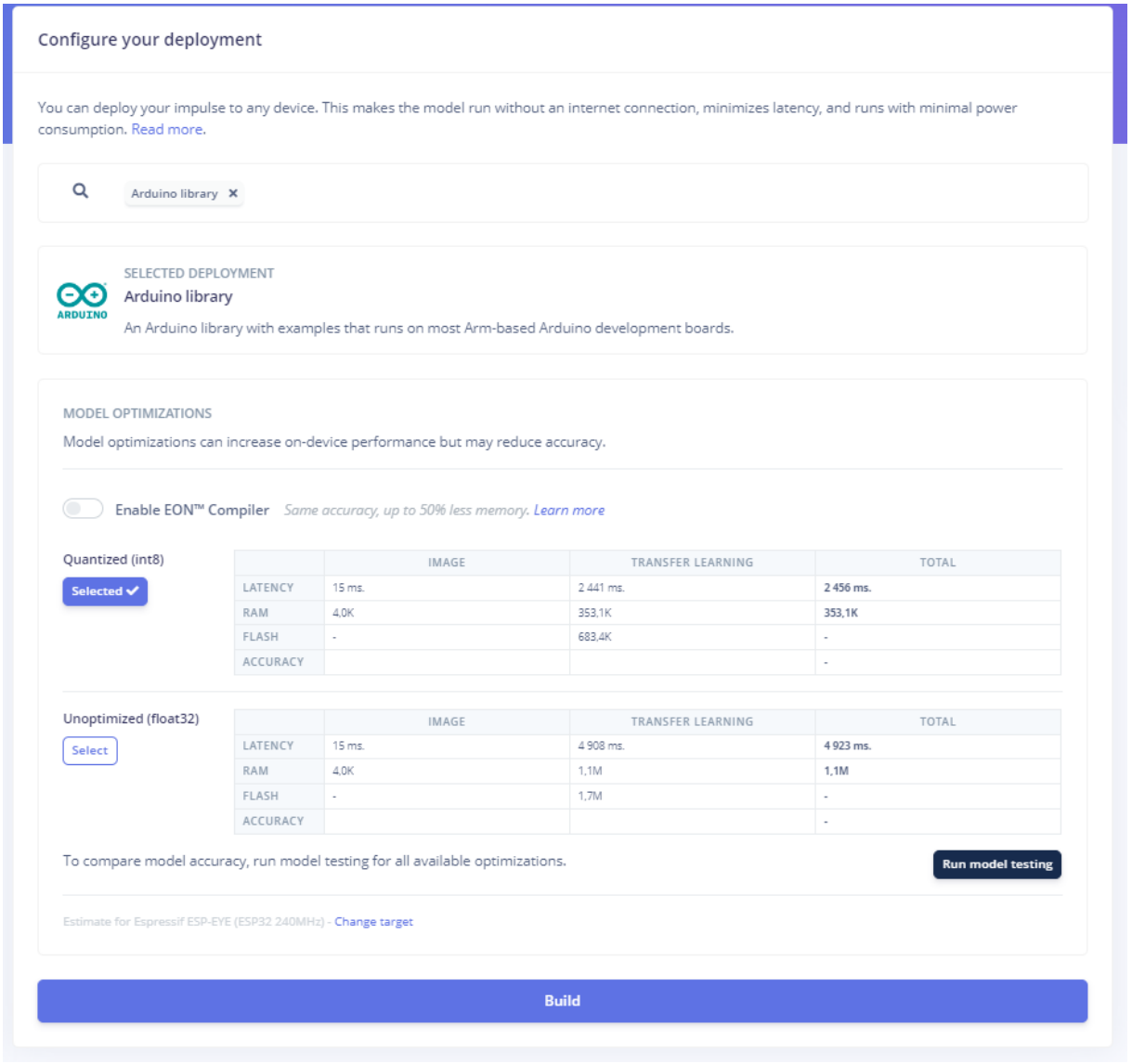


Accuracy is 60% for optimized and 80% for unoptimized but **inferencing time** is twice as much as optimized. Accuracy could be better, but for training we had only 24 pictures and for testing 6 pictures.



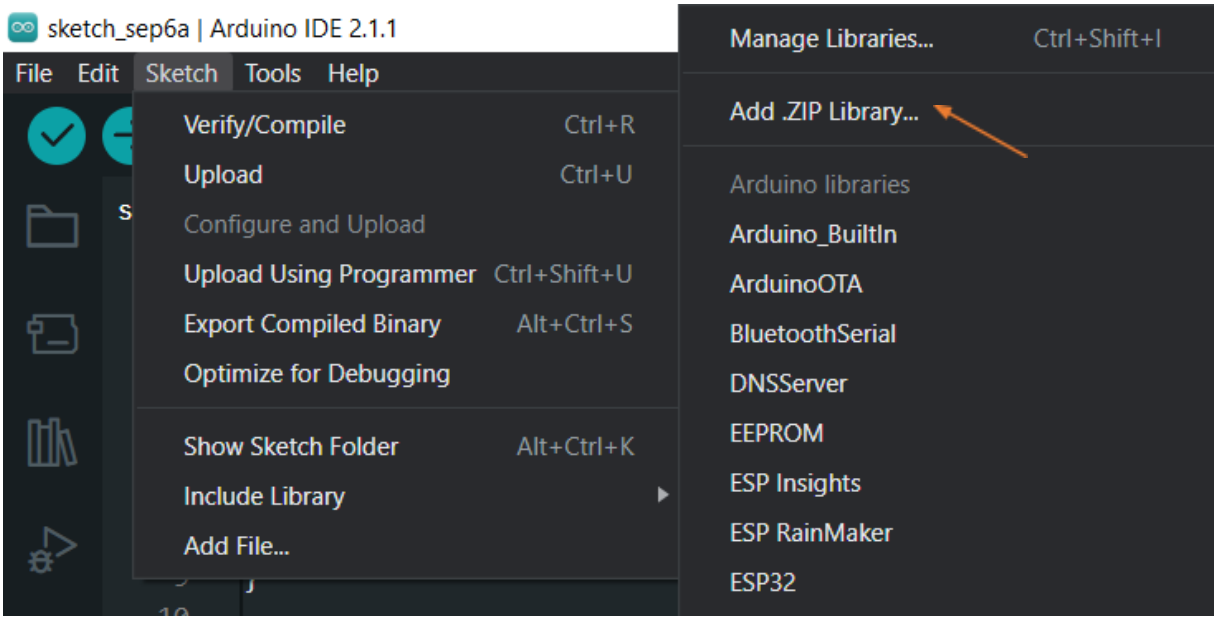
6. Deployment

In sidebar click on ‘Deployment’, choose Arduino library, turn off EON compiler because for ESP32-S3 Edge Impulse hasn’t released SDK yet and click on build

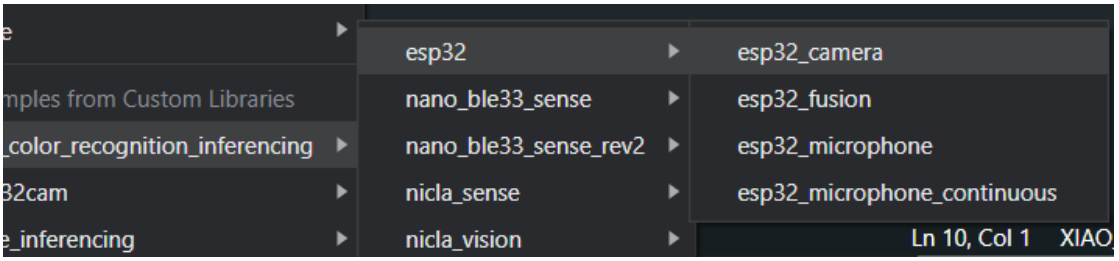


7. Including to the library

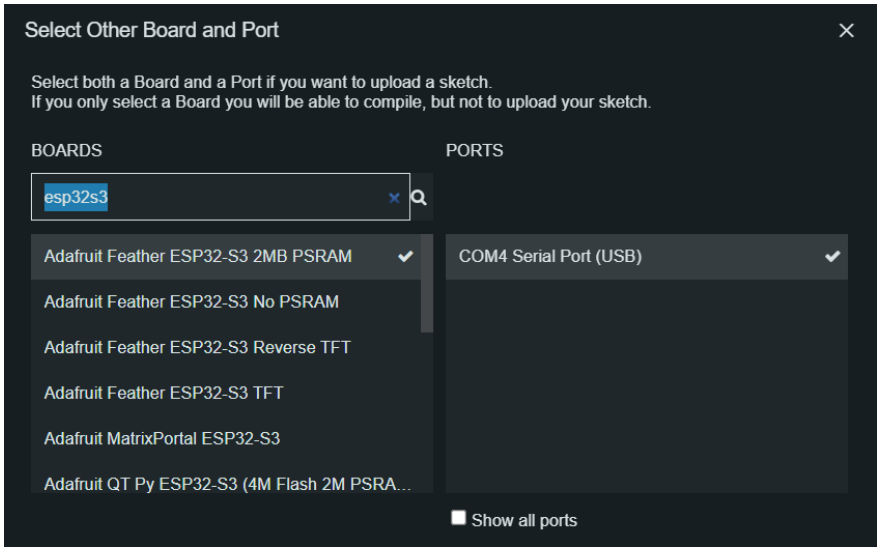
After downloading zip file include file in Arduino IDE



After library is installed go to File>Examples and scroll to the bottom for examples from custom libraries. Library name should be same name as project and choose camera



You can select this board with PSRAM or XIAO_ESP32S3, it's same SoC



8. Configuration

You must configurate pins, in this screenshot you can delete it and set up like this below:

```
#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
#define Y3_GPIO_NUM      13
#define Y2_GPIO_NUM      34
#define VSYNC_GPIO_NUM   5
#define HREF_GPIO_NUM    27
#define PCLK_GPIO_NUM    25
#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
```

```
#include "esp_camera.h"

// Select camera model - find more camera
// https://github.com/espressif/arduino-e

#define CAMERA_MODEL_XIAO_ESP32S3 // Has

#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM     15
#define SIOD_GPIO_NUM     4
#define SIOC_GPIO_NUM     5

#define Y9_GPIO_NUM       16
#define Y8_GPIO_NUM       17
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       12
#define Y5_GPIO_NUM       10
#define Y4_GPIO_NUM       8
#define Y3_GPIO_NUM       9
#define Y2_GPIO_NUM       11
#define VSYNC_GPIO_NUM    6
#define HREF_GPIO_NUM     7
#define PCLK_GPIO_NUM     13
```

Don't forget to disable the ESP NN acceleration. Failure to do so may result in the following error message:

```
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
Arduino\libraries\Car_color_recognition_inferencing\src\edge-impulse-sdk\porting\espressif\ESP-NN\src\convolution/esp_nn_depthwise_conv_s8_esp32s3.c:494: undefined refe
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
Arduino\libraries\Car_color_recognition_inferencing\src\edge-impulse-sdk\tensorflow\lite\micro\kernels/add.cpp:1115: undefined reference to `esp_nn_add_elementwise_s8_e
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
cal/arduino15/packages/esp32/tools/xtensa-esp32s3-elf-gcc/esp-2021r2-patch5-8.4.0/bin/../lib/gcc/xtensa-esp32s3-elf/8.4.0/../../../../xtensa-esp32s3-elf/bin/ld.exe: C:\
Arduino\libraries\Car_color_recognition_inferencing\src\edge-impulse-sdk\tensorflow\lite\micro\kernels/fully_connected.cpp:1539: undefined reference to `esp_nn_fully_co
eturned 1 exit status
```

Go to:

Documents\Arduino\libraries\Car_color_recognition_inferencing\src\edge-impulse-sdk\classifier/ei_classifier_config.h

And change this line from 1 to 0 and save:

```
#define EI_CLASSIFIER_TFLITE_ENABLE_ESP_NN       0
```

You must do some changes in this code because generated code be EI it doesn't work for this stick, I mean code works but in serial monitor it don't have any outputs

Change every: Serial > Serial0

ei_printf > Serial0.printf

it should look like this:

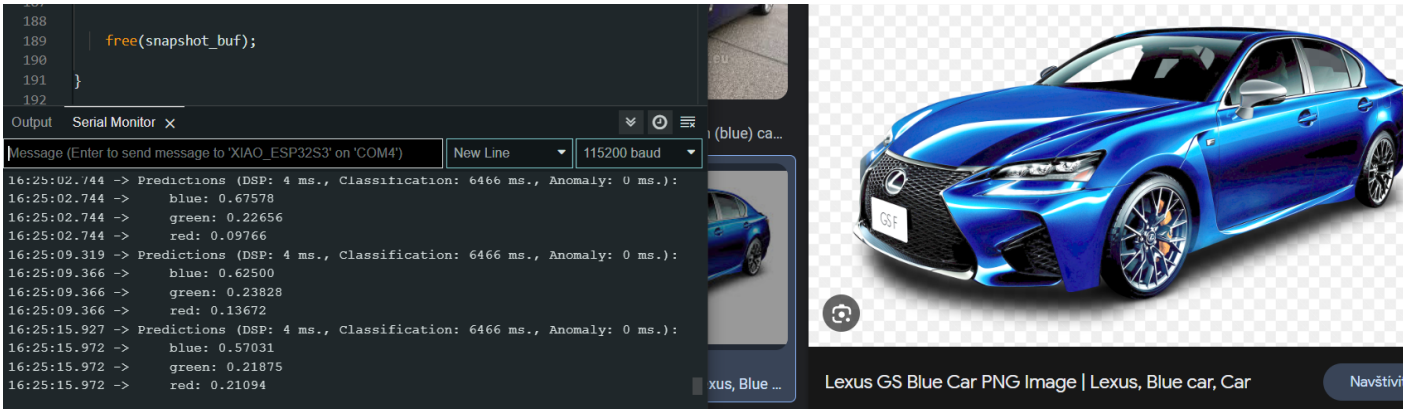
```
void setup()
{
    // put your setup code here, to run once:
    Serial0.begin(115200);
    //comment out the below line to start inference immediately after upload
    while (!Serial0);
    Serial0.println("Edge Impulse Inferencing Demo");
    if (ei_camera_init() == false) {
        Serial0.printf("Failed to initialize Camera!\r\n");
    }
    else {
        Serial0.printf("Camera initialized\r\n");
    }

    Serial0.printf("\nStarting continious inference in 2 seconds...\n");
    ei_sleep(2000);
}
```

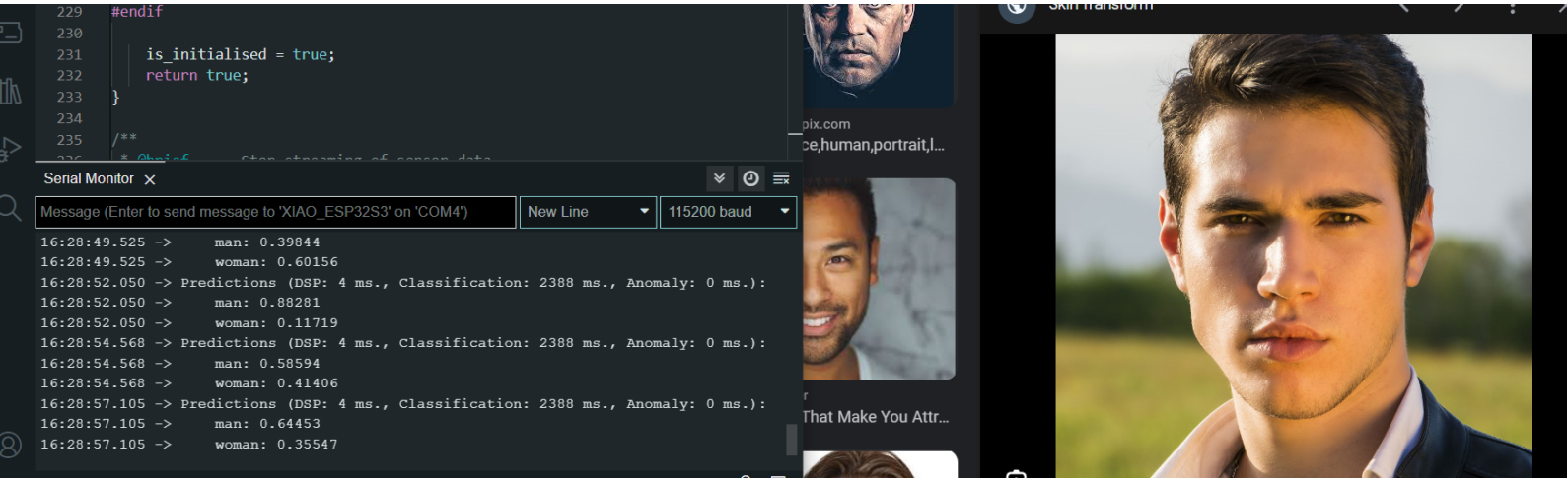
9 . Test

You can see it, I pointed with camera on image of green car, it took 6 seconds which is slow you can lower image size(height and width)

Accurate of around 60% for blue



Here's another model for recognition with an accuracy rate of 75%. This model incorporates a larger dataset with approximately 65 images, evenly distributed between men and women.



It takes some time to recognize if it’s a woman or a man