



XIAO_ESP32S3



XIAO_Image_Capture_JPEG.ino



```
1  /*****
2  * XIAO ESP32-S3 Image Capture for CNN Dataset (96x96x3)
3  * -----
4  * This sketch captures images from the OV2640 camera attached to the Seeed XIAO ESP32-S3,
5  * resizes them to 96x96 RGB, and saves them as JPEG files to the SPI SD card.
6  *
7  * FEATURES:
8  * 1. Prompts the user via Serial Monitor to enter the current date/time at startup.
9  *   - Format: YYYY/MM/DD HH:MM:SS
10 *   - Used to timestamp the filenames.
11 * 2. Prompts the user to enter a label for each batch of images.
12 *   - The label is included in each file name: <label>_YYYYMMDD_HHMMSS.jpg
13 * 3. Captures images in batches (IMAGE_COUNT), applying a 1-second delay between captures.
14 * 4. Resizes each captured frame to 96x96 RGB before saving to SD card.
15 * 5. Uses the working SPI SD card configuration:
16 *   - CS = 21, SCK = 7, MISO = 8, MOSI = 9
17 * 6. Uses the official Seeed XIAO ESP32-S3 OV2640 pinout for camera stability.
18 * 7. JPEG quality adjustable via JPEG_QUALITY.
19 *
20 * REQUIREMENTS:
21 * - SD card formatted as FAT32.
22 * - Libraries:
23 *   - SD.h
```

Output Serial Monitor X

Not connected. Select a board and a port to connect automatically.

```
22:15:53.752 ->
22:15:53.752 -> --- Ready for image capture ---
22:15:53.752 -> Enter the image label for this batch and press Enter:
22:16:03.942 ->
22:16:03.942 -> *** Starting capture batch for label: door ***
22:16:03.942 -> Capturing image 1/10... Saved.
22:16:03.983 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:04.979 -> Capturing image 2/10... Saved.
22:16:05.021 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:05.989 -> Capturing image 3/10... Saved.
22:16:06.024 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:07.024 -> Capturing image 4/10... Saved.
22:16:07.071 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:08.065 -> Capturing image 5/10... Saved.
22:16:08.105 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:09.087 -> Capturing image 6/10... Saved.
22:16:09.162 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:10.135 -> Capturing image 7/10... Saved.
22:16:10.178 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:11.198 -> Capturing image 8/10... Saved.
22:16:11.239 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:12.223 -> Capturing image 9/10... Saved.
22:16:12.266 -> Pausing for 1 second(s). Move camera/change lighting now.
22:16:13.260 -> Capturing image 10/10... Saved.
22:16:13.301 ->
22:16:13.301 -> *** Batch Complete: 10 images saved under label door ***
22:16:13.301 -> Enter the image label for the next batch and press Enter:
```



Visualizing representative images...
Loaded 'door_closed' image: /content/door_closed20251118_094517.jpg
Loaded 'door_open' image: /content/door_open20251118_093312.jpg

Door Closed Example



Door Open Example



Representative images displayed successfully.

CNN Model Summary:
Model: "sequential_4"

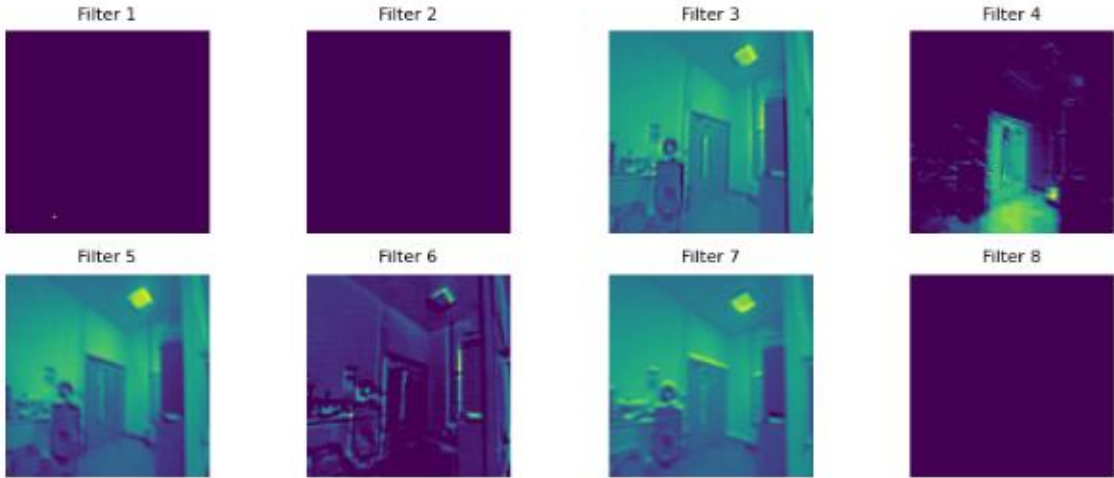
Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 94, 94, 32)	896
max_pooling2d_8 (MaxPooling2D)	(None, 47, 47, 32)	0
conv2d_9 (Conv2D)	(None, 45, 45, 64)	18,496
max_pooling2d_9 (MaxPooling2D)	(None, 22, 22, 64)	0
flatten_4 (Flatten)	(None, 30976)	0
dense_8 (Dense)	(None, 128)	3,965,056
dense_9 (Dense)	(None, 1)	129

Total params: 3,984,577 (15.20 MB)
Trainable params: 3,984,577 (15.20 MB)
Non-trainable params: 0 (0.00 B)
CNN model defined and summary printed successfully.

Visualizing convolutions for a sample 'door_closed' image...
1/1 _____ 0s 144ms/step

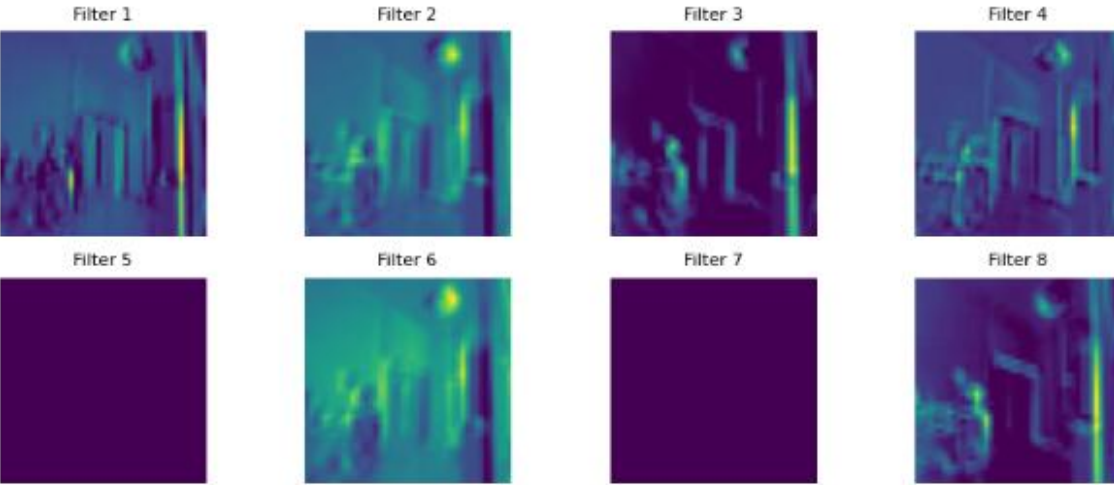
visualizing feature maps for layer: conv2d_8

Feature Maps for conv2d_8

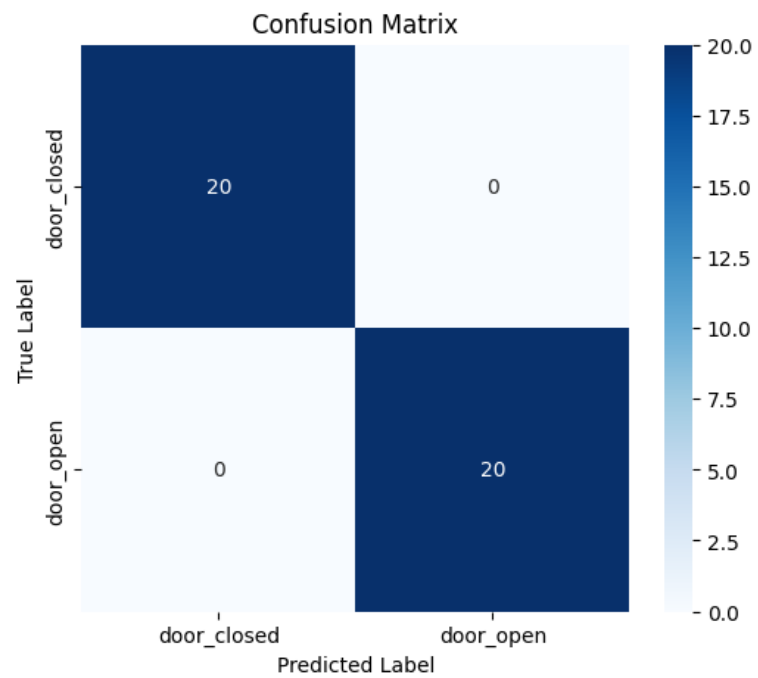
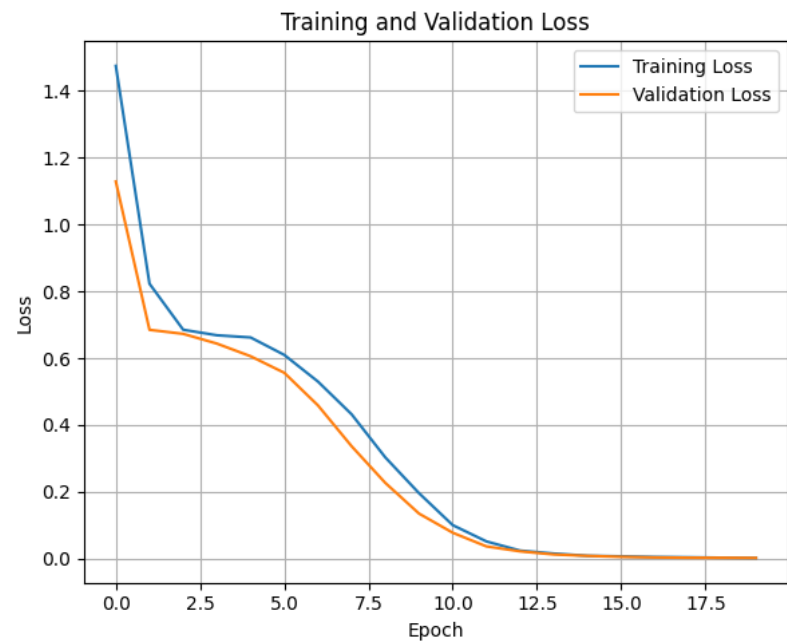
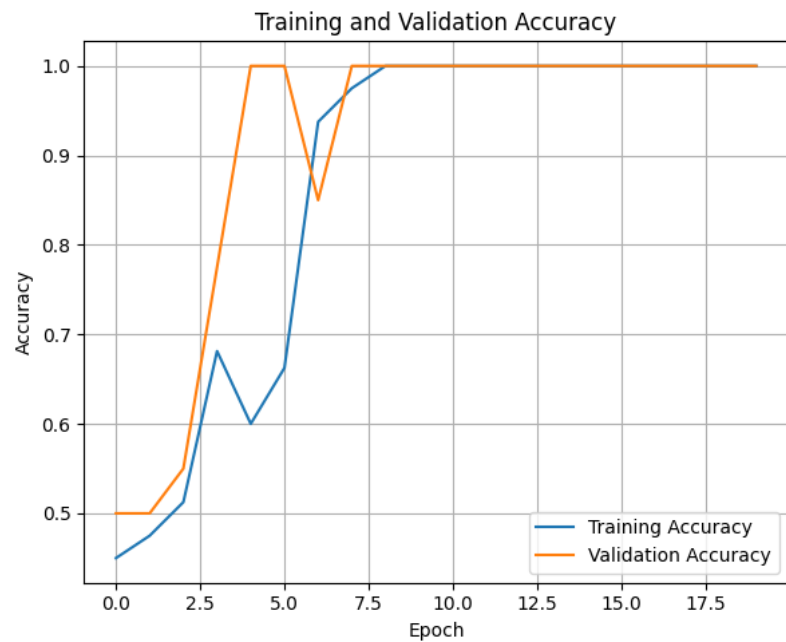


visualizing feature maps for layer: conv2d_9

Feature Maps for conv2d_9



Feature map visualization complete.



Edge_AI_Door_Detect.ipynb - C x +

colab.research.google.com/drive/1YSEEB3ONwUQaTi1QBABDVDli1EwDWfky#scrollTo=095d8b3b

YouTube Maps Gmail Adobe Acrobat

Edge_AI_Door_Detect.ipynb ☆

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- door_open20251118_093713.jpg
- door_open20251118_093714.jpg
- door_open20251118_093716.jpg
- door_open20251118_093717.jpg
- door_open20251118_093718.jpg
- door_open20251118_093806.jpg
- door_open20251118_093807.jpg
- door_open20251118_093809.jpg
- door_open20251118_093810.jpg
- door_open20251118_093811.jpg
- door_open20251118_093812.jpg
- door_open20251118_093813.jpg
- door_open20251118_093814.jpg
- door_open20251118_093816.jpg
- door_open20251118_093817.jpg
- door_open20251118_093907.jpg
- door_open20251118_093908.jpg
- door_open20251118_093909.jpg
- door_open20251118_093910.jpg
- door_open20251118_093911.jpg
- door_open20251118_093912.jpg
- door_open20251118_093914.jpg
- door_open20251118_093915.jpg
- door_open20251118_093916.jpg
- door_open20251118_093917.jpg
- model_quantized.tflite

```
# 5. Save the tflite_model to a file
quantized_model_path = 'model_quantized.tflite'
with tf.io.gfile.GFile(quantized_model_path, 'wb') as f:
    f.write(tflite_model)
print(f"Quantized TFLite model saved to: {quantized_model_path}")

# 6. Calculate the size of the original Keras model
# Save the original model temporarily to get its size
original_model_path = 'original_keras_model.h5'
model.save(original_model_path)
original_model_size = os.path.getsize(original_model_path) / 1024 # in KB
print(f"Original Keras model size: {original_model_size:.2f} KB")

# 7. Calculate the size of the converted TFLite model
quantized_model_size = os.path.getsize(quantized_model_path) / 1024 # in KB
print(f"Quantized TFLite model size: {quantized_model_size:.2f} KB")

# Clean up the temporarily saved original model
os.remove(original_model_path)

print("TensorFlow Lite model conversion and size comparison complete.")
```

Starting TensorFlow Lite model conversion (INT8 quantized)...
Converting model to TFLite INT8...
Saved artifact at '/tmp/tmptsfbu9ox'. The following endpoints are available:

* Endpoint 'serve'
args_0 (POSITIONAL_ONLY): TensorSpec(shape=(None, 96, 96, 3), dtype=tf.float32, name='keras_tensor')
Output Type:
TensorSpec(shape=(None, 1), dtype=tf.float32, name=None)
Captures:
132992285878864: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285880016: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285878672: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285877328: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285879440: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285879056: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285880400: TensorSpec(shape=(), dtype=tf.resource, name=None)
132992285879632: TensorSpec(shape=(), dtype=tf.resource, name=None)
/usr/local/lib/python3.12/dist-packages/tensorflow/lite/python/convert.py:854: UserWarning: Statistics for quantization warnings.warn(
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. Model converted successfully.
Quantized TFLite model saved to: model_quantized.tflite
Original Keras model size: 46732.59 KB
Quantized TFLite model size: 3900.88 KB
TensorFlow Lite model conversion and size comparison complete.

Left-click->download

```
tflite_to_h.py
File Edit View

import argparse

# Define the expected array name in the Arduino sketch
ARRAY_NAME = 'door_status_model'

def generate_header(tflite_path, header_path, array_name):
    with open(tflite_path, 'rb') as f:
        tflite_model = f.read()

    # Format the model data as a C array
    hex_data = ', '.join([f'0x{byte:02x}' for byte in tflite_model])

    # Write the C header file
    with open(header_path, 'w') as f:
        f.write(f'#ifndef MODEL_H\n')
        f.write(f'#define MODEL_H\n')
        f.write(f'// Model size: {len(tflite_model)} bytes\n')
        f.write(f'const unsigned char {array_name}[] = {{\n')
        f.write(f'    {hex_data}\n')
        f.write(f'}};\n')
        f.write(f'const int {array_name}_len = {len(tflite_model)};\n')
        f.write(f'#endif // MODEL_H\n')

    print(f"Successfully generated {header_path} ({len(tflite_model)} bytes) with array name '{array_name}'")

# Define the names of your files
input_file = "model_quantized.tflite"
output_file = "model.h"

# NOTE: The Arduino sketch expects the variable name 'person_model',
# even if your input file is 'door_status_model.tflite'.
generate_header(input_file, output_file, ARRAY_NAME)
```

Make sure input file name matches file that was downloaded in previous step

```
Windows PowerShell
PS C:\Users\Kris\Desktop\Edge_AI> python tflite_to_h.py
Successfully generated model.h (3994504 bytes) with array name 'door_status_model'.
PS C:\Users\Kris\Desktop\Edge_AI>
```



Note: The model is quantised int8 to ensure it is small enough for the Seeed XIAO!

XIAO_Image_Classifier_door				
Start back up > Desktop > Edge_AI > XIAO_Image_Classifier_door				
New Cut Copy Paste Sort View ...				
Home Gallery OneDrive - Personal				
Name	Date modified	Type	Size	
model	18/11/2025 22:17	H File	23,406 KB	
XIAO_Image_Classifier_door	18/11/2025 22:17	INO File	10 KB	

Auto Format Ctrl+T

Archive Sketch

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter

Firmware Updater

Upload SSL Root Certificates

Board: "XIAO_ESP32S3" ▶

Port: "COM11" ▶

Get Board Info ▶

USB CDC On Boot: "Enabled" ▶

CPU Frequency: "160MHz (WiFi)" ▶

Core Debug Level: "None" ▶

USB DFU On Boot: "Enabled (Requires USB-OTG Mode)" ▶

Erase All Flash Before Sketch Upload: "Disabled" ▶

Events Run On: "Core 1" ▶

Flash Mode: "QIO 80MHz" ▶

Flash Size: "8MB (64Mb)" ▶

JTAG Adapter: "Disabled" ▶

Arduino Runs On: "Core 1" ▶

USB Firmware MSC On Boot: "Disabled" ▶

Partition Scheme: "Maximum APP (7.9MB APP No OTA/No FS)" ▶

PSRAM: "OPI PSRAM" ▶

Upload Mode: "UART0 / Hardware CDC" ▶

Upload Speed: "921600" ▶

USB Mode: "Hardware CDC and JTAG" ▶

Programmer ▶

Burn Bootloader

```
point) * output->params.scale;  
params.zero_point) * output->params.scale;
```

```
params.zero_point);
```

```
f)\n",
```

```
.2f)\n",
```

Default with spiiffs (3MB APP/1.5MB SPIFFS)

✓ Maximum APP (7.9MB APP No OTA/No FS)

TinyUF2 8MB (2MB APP/3.7MB FFAT)

TinyUF2 8MB No OTA (4MB APP/3.7MB FFAT)

```
1  /**  
2   * @file XIAO_Image_Classifier_Door.ino  
3   * @brief FINAL SKETCH: Corrected to use 3-channel (RGB565) input, detection threshold lowered,  
4   * and the classification logic is updated from Person/No-Person to **Door Closed/Door Open**.  
5   * -----  
6   * --- UPDATES APPLIED ---  
7   * 1. Logic Change: Classification updated to **DOOR_CLOSED** and **DOOR_OPEN**.  
8   * 2. Constants Updated: PERSON/NO_PERSON indices changed to DOOR_CLOSED/DOOR_OPEN indices.  
9   * 3. Comments/Prints: Updated throughout the code to reflect the new task.  
10  * -----  
11  */  
12  
13  #include <Arduino.h>  
14  #include <esp_heap_caps.h>  
15  #include <tensorflow/lite/micro/micro_mutable_op_resolver.h>  
16  #include <tensorflow/lite/micro/micro_interpreter.h>  
17  #include <tensorflow/lite/micro/micro_log.h>  
18  #include <tensorflow/lite/micro/system_setup.h>  
19  #include <tensorflow/lite/schema/schema_generated.h>  
20  #include "tensorflow/lite/micro/micro_utils.h"  
21  #include "esp_camera.h"  
22  #include "esp_log.h"  
23  // You must ensure 'model.h' containing 'door_status_model' is in your project directory  
24  #include "model.h"  
25  
26  // --- CONSTANTS AND CONFIGURATION ---  
27  const int kTensorArenaSize = 384 * 1024;  
28  // ASSUMPTION: DOOR_CLOSED is index 0 and DOOR_OPEN is index 1 in the model's output.  
29  #define DOOR_OPEN_INDEX 1  
30  #define DOOR_CLOSED_INDEX 0  
31  #define DETECTION_THRESHOLD 0.70 // Sensitivity for positive detection (DOOR OPEN)  
32  #define MODEL_INPUT_WIDTH 96  
33  #define MODEL_INPUT_HEIGHT 96  
34
```

Output

```
Writing at 0x0040f940 [=====] ] 95.5% 3784704/3961220 bytes...  
  
Writing at 0x00414e17 [=====] ] 96.0% 3801088/3961220 bytes...  
  
Writing at 0x00419e32 [=====] ] 96.4% 3817472/3961220 bytes...  
  
Writing at 0x0041f1e7 [=====] ] 96.8% 3833856/3961220 bytes...  
  
Writing at 0x004256dd [=====] ] 97.2% 3850240/3961220 bytes...  
  
Writing at 0x0042b49d [=====] ] 97.6% 3866624/3961220 bytes...  
  
Writing at 0x00430a77 [=====] ] 98.0% 3883008/3961220 bytes...  
  
Writing at 0x00437921 [=====] ] 98.4% 3899392/3961220 bytes...  
  
Writing at 0x0043f2d3 [=====] ] 98.9% 3915776/3961220 bytes...  
  
Writing at 0x00444ff3 [=====] ] 99.3% 3932160/3961220 bytes...  
  
Writing at 0x0044ad96 [=====] ] 99.7% 3948544/3961220 bytes...  
  
Writing at 0x0044f8d0 [=====] ] 100.0% 3961220/3961220 bytes...  
Wrote 4454608 bytes (3961220 compressed) at 0x00010000 in 38.4 seconds (927.6 kbit/s).  
Hash of data verified.  
  
Hard resetting via RTS pin...
```


camera_usb_viewer.ino

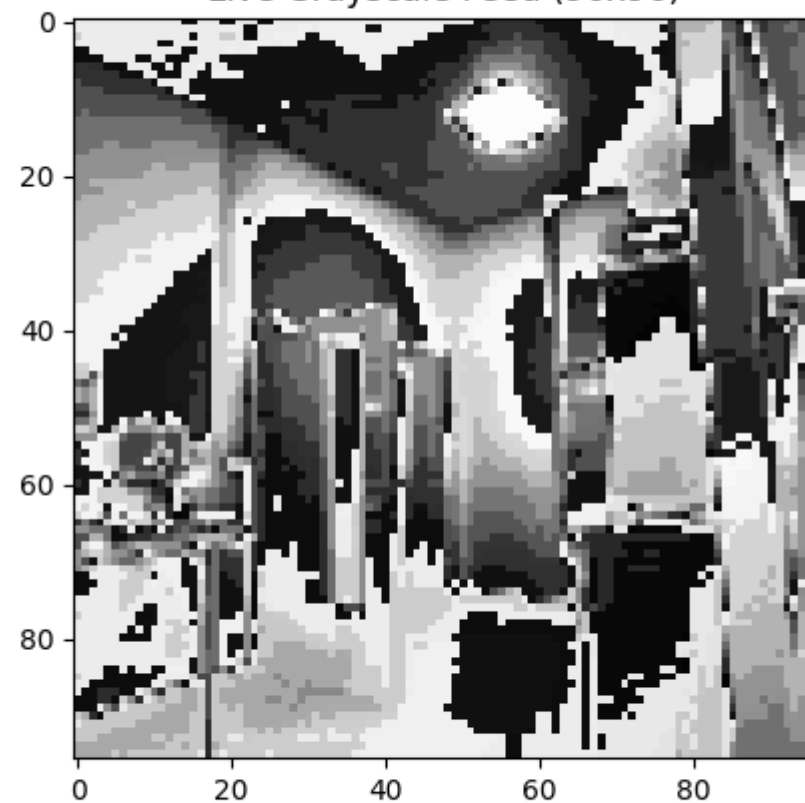
```
1  /**
2   * @file Camera_USB_Viewor.ino
3   * @brief Standalone sketch to stream 96x96 grayscale image data over USB serial.
4   * This is used to diagnose image brightness/quality issues (like the door-open scene being too dark).
5   */
6
7  #include <Arduino.h>
8  #include "esp_camera.h"
9  #include "esp_log.h"
10
11  #define IMAGE_WI
12  #define IMAGE_HE
13  #define FRAME_DE
14
15  // Camera config
16  static camera_co
17  .pin_pwdn =
18  .pin_reset =
19
20  // --- CORRE
21  .pin_xclk =
22  .pin_sccb_sd
23  .pin_sccb_sc
24
25  // Data Pins (D7-D0)
26  .pin_d7 = 48, .pin_d6 = 11, .pin_d5 = 12, .pin_d4 = 14,
27  .pin_d3 = 16, .pin_d2 = 18, .pin_d1 = 17, .pin_d0 = 15,
28
29  // Sync Pins
30  .pin_vsync = 38, .pin_href = 47, .pin_pclk = 13,
31  .xclk_freq_hz = 20000000,
32
33  .ledc_timer = LEDC_TIMER_0,
34  .ledc_channel = LEDC_CHANNEL_0,
35  .pixel_format = PIXFORMAT_RGB565,
36  .frame_size = FRAMESIZE_96X96,
37  .jpeg_quality = 12,
38  .fb_count = 1,
39  .fb_location = CAMERA_FB_IN_PSRAM.
```

Output

```
Writing at 0x0003a8b3 [=====] 41.0% 81920/199599 bytes...
Writing at 0x00040177 [=====] 49.3% 98304/199599 bytes...
Writing at 0x0004662f [=====] 57.5% 114688/199599 bytes...
Writing at 0x0004c9e7 [=====] 65.7% 131072/199599 bytes...
Writing at 0x00053017 [=====] 73.9% 147456/199599 bytes...
Writing at 0x0005b431 [=====] 82.1% 163840/199599 bytes...
Writing at 0x00060ed8 [=====] 90.3% 180224/199599 bytes...
Writing at 0x00066e24 [=====] 98.5% 196608/199599 bytes...
Writing at 0x000683f0 [=====] 100.0% 199599/199599 bytes...
Wrote 361456 bytes (199599 compressed) at 0x00010000 in 2.7 seconds (1075.9 kbit/s).
Hash of data verified.
Hard resetting via RTS pin...
```

Figure 1

Live Grayscale Feed (96x96)




```
1  /**
2   * @file XIAO_Image_Classifier_Door.ino
3   * @brief FINAL DEFINITIVE SKETCH: All previous fixes (TFLite, memory, camera stability) are retained.
4   * Classification logic is FIXED to trigger on the reliable DROP in the "Closed" score, as the "Open"
5   * score is unstable.
6   */
7
8  #include <Arduino.h>
9  #include <esp_heap_caps.h>
10 // TFLite Includes
11 #include <tensorflow/lite/micro/micro_mutable_op_resolver.h>
```

Output Serial Monitor x

Message (Enter to send message to 'XIAO_ESP32S3' on 'COM12')

```
12:45:09.134 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:10.648 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:10.649 -> DEBUG: Raw Scores: Closed=-30, Open=-128
12:45:10.649 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:12.194 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:12.194 -> DEBUG: Raw Scores: Closed=-124, Open=-128
12:45:12.194 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:13.703 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:13.703 -> DEBUG: Raw Scores: Closed=-30, Open=-128
12:45:13.703 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:15.228 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:15.228 -> DEBUG: Raw Scores: Closed=51, Open=-128
12:45:15.228 -> 🚪 Door Closed. Highest 'Closed' Score: 0.70 ('Open' Score: 0.00)
12:45:16.743 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:16.743 -> DEBUG: Raw Scores: Closed=-57, Open=-128
12:45:16.743 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:18.213 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:18.213 -> DEBUG: Raw Scores: Closed=-69, Open=-128
12:45:18.213 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:19.726 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:19.726 -> DEBUG: Raw Scores: Closed=-104, Open=-128
12:45:19.726 -> ■ **DOOR OPENED DETECTED!** Confidence: 0.00 (Closed Score Drop: < 0.55)
12:45:21.267 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:21.267 -> DEBUG: Raw Scores: Closed=44, Open=-128
12:45:21.267 -> 🚪 Door Closed. Highest 'Closed' Score: 0.67 ('Open' Score: 0.00)
12:45:22.780 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:22.780 -> DEBUG: Raw Scores: Closed=15, Open=-128
12:45:22.780 -> 🚪 Door Closed. Highest 'Closed' Score: 0.56 ('Open' Score: 0.00)
12:45:24.289 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:24.289 -> DEBUG: Raw Scores: Closed=104, Open=-128
12:45:24.289 -> 🚪 Door Closed. Highest 'Closed' Score: 0.91 ('Open' Score: 0.00)
12:45:25.801 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:25.801 -> DEBUG: Raw Scores: Closed=95, Open=-128
12:45:25.801 -> 🚪 Door Closed. Highest 'Closed' Score: 0.87 ('Open' Score: 0.00)
12:45:27.302 -> DEBUG: Scale=0.0039, ZeroPoint=-128
12:45:27.302 -> DEBUG: Raw Scores: Closed=98, Open=-128
12:45:27.302 -> 🚪 Door Closed. Highest 'Closed' Score: 0.88 ('Open' Score: 0.00)
```