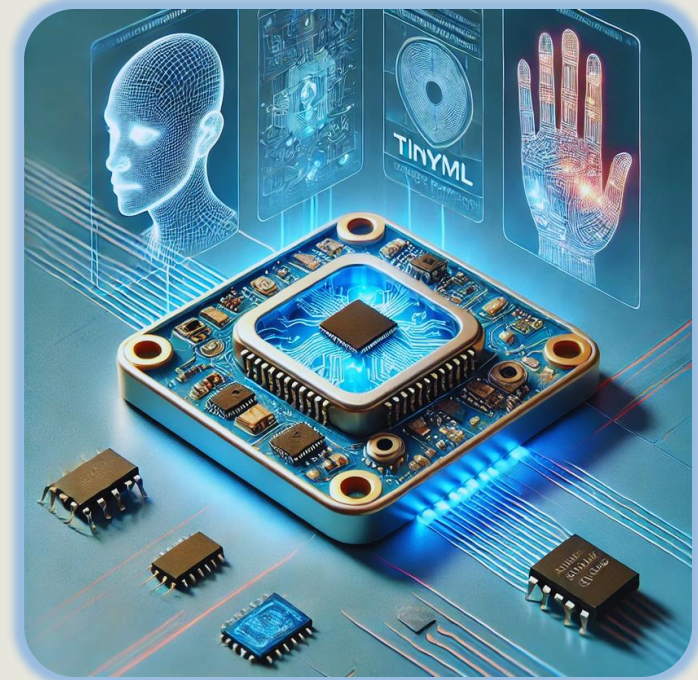


Introducing **FaceSafe** On-Device Authentication & Gesture Recognition

1

Embracing ***TinyML*** for Privacy and Efficiency



Team: Sleeping-Buddha
Member: Afsara Benazir, Shafat Shahnewaz, Samit Hasan

Powered by: **XIAO ESP32S3 Sense**

Embracing TinyML for Privacy and Efficiency

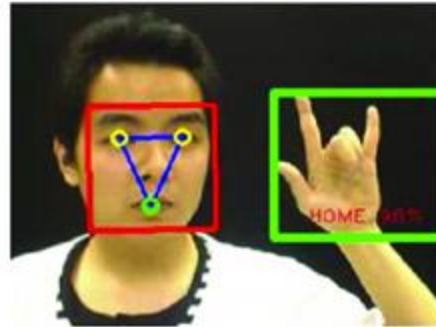
Recognize individual face for authentication and gestures for action identification.

Key Objectives:

- Privacy
- Low Power & High Efficiency
- Real-Time Responsiveness

Usability

- Classroom attendance
- Smart Home features



(a)



(b)

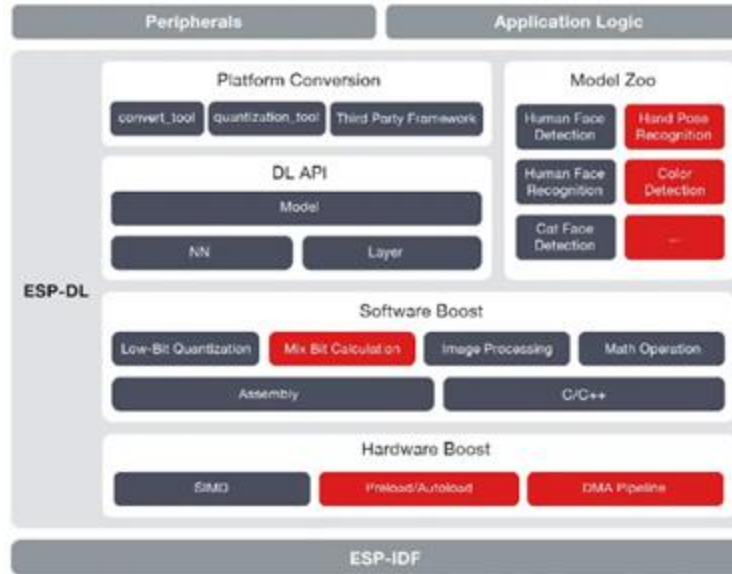
Seeed Studio XIAO ESP32S3: Powering TinyML On-Device Processing

Core Specs of ESP32S3:

- Dual-core Tensilica LX7
- 240 MHz Frequency
- 512KB SRAM & 8MB Flash

Additional Peripherals:

- Integrated Wi-Fi and Bluetooth
- Built-in Camera Support



21 mm

17.5 mm

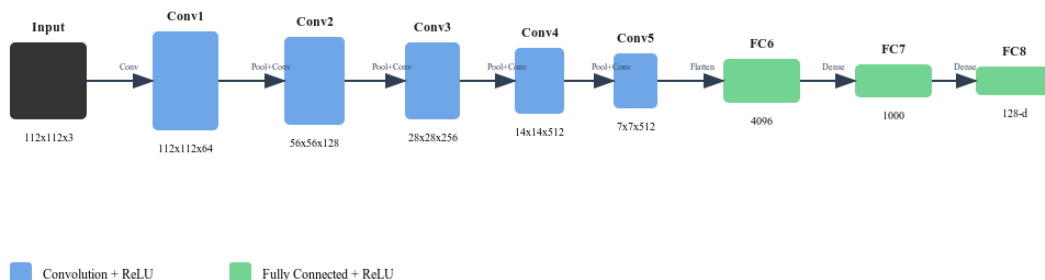
Models Deployed

- Face Detection Model

At first a two -stage face detection model is identified two place a rectangular box around the face so that the input of face recognition is reduced.

- Face Recognition Model

A deep neural network model employs convolutional layers and depth wise separable convolutions to extract and encode facial features into a compact, 128-dimensional embedding



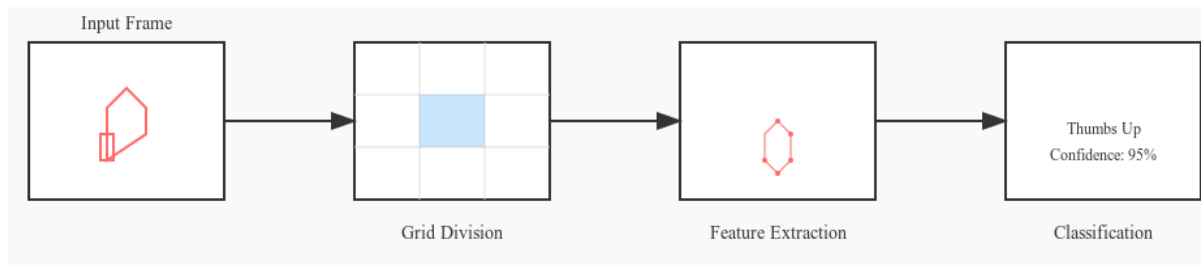
FaceRecognition112V128 Architecture

- Hand Gesture Detection

For hand recognition we used Swift YOLO model which is a variation of the YOLO (You Only Look Once) object detection algorithm tailored for lightweight and efficient real-time inference.

Gesture recognition

- SenseCraft AI framework
 - Rock paper scissors
 - Left/right hand recognition
 - More compatible
- Espressif
 - Less compatible
 - Challenges in deploying, doesn't take streaming data



Path to Success: Key Steps in Project Development

1. Hardware Setup:



- Installation of **camera**, **antenna**, and **microSD card**.
- Configuration of **WiFi antenna**
- Verification of hardware functionality

2. Software Setup:



- Installation **Arduino IDE** for code development.
- Upload initial firmware to test the camera module functionality.
- Mapping of the GPIO pins with the camera slot for seamless communication.
- Enabling **PSRAM** to support larger images.

3. Camera and Model Integration:



- Capturing images using the ESP32-S3 camera.
- Configuring the system to push captured images as inputs to the **image classification model** for face recognition.
- Tuning the system for efficient picture capture and classification.

4. Deployment of User Interface:



- Deploying a **web-browser-based GUI** for a user-friendly experience.
- Live video stream enabling via the camera feed.
- GUI functionality integration over **WiFi**, allowing seamless user access.

5. Gesture Recognition Pipeline:

- Integration of the gesture recognition model after face recognition.
- Configuring the system to transition from face detection to gesture recognition for actionable insights.

Demo video: Face detection & Recognition



- Adds a rectangular box when face is detected.
- If the face is recognized it shows a green box with associated ID and recognition accuracy
- Category: Person ID, Intruder
- Latency of recognition: ~ 0.7 sec
- Current Consumption: ~ 341 mA

Demo video: Rock paper scissors

Process

Output

MQTT

GPIO

Serial Port

Takes in streaming video data

Preview

Stop



IP Address:

Service Status:

MQTT not initialized or not connected

Device Logger

```
" :0}
boxes: [[91.31,193.8,81.99,100.63,56.65,2]]

perf:{"preprocess":70,"inference":660,"postprocess":0}
boxes: [[95.03,199.39,68.95,85.72,53.67,2],[102.49,169.57,102.49,132.3,31.31,2]]

perf:{"preprocess":100,"inference":660,"postprocess":0}
boxes: [[91.31,152.8,74.54,117.4,25.34,2],[100.63,195.66,81.99,100.63,53.67,2]]

perf:{"preprocess":70,"inference":660,"postprocess":0}
boxes: [[106.22,188.21,91.31,115.53,56.65,2]]

perf:{"preprocess":110,"inference":660,"postprocess":0}
boxes: [[108.08,175.16,68.95,63.36,50.69,1],[109.94,195.66,81.99,100.63,74.54,1]]

perf:{"preprocess":90,"inference":660,"postprocess":0}
```


Demo: Left/right hand recognition

The screenshot shows the ClassTrain web interface for training a model for left/right hand recognition. The interface is divided into several sections:

- right 0**: A section for training on right-hand gestures, showing 41 image samples in a row.
- left 0**: A section for training on left-hand gestures, showing 45 image samples in a row.
- Webcam**: A live video feed showing a person pointing their right hand towards the camera. Below the feed is a green button labeled "Hold to Record".
- Step 2: Training**: A section with a dropdown menu set to "Grove (V2) / Watcher" and a green "Start Training" button. Below this is an "Advanced" dropdown.
- Step 3: Deploy**: A section with a dropdown menu set to "Supported Devices" and two green buttons: "Deploy to device" and "Save to SenseCraft".
- ClassTrain**: A section with tabs for "All", "QuickGen", "TextImgGen", and "ClassTrain". The "ClassTrain" tab is active, showing a list of training files. The first file is "ClassTrain_Xiao_24124190022(UTC-5)_vela.tflite (right, left)" with a size of 320.06 KB and a timestamp of 2024/12/4 19:00:22(UTC-5).

Below the "ClassTrain" section, there is a "Preview" section showing two progress bars: "right" at 81% and "left" at 18%.

Train data: 86 image samples
Model: Swift-YOLO
Size: < 2MB
Latency: 700-800ms

Espressif: Details and issues

Project directory and inference:

```

> .devcontainer
> .vscode
> build
> components
  > esp-dl
  > esp-who
> main
  C app_main.cpp
  M CMakeLists.txt
> model
  C handrecognition_coefficient.cpp
  C handrecognition_coefficient.hpp
  C model_define.hpp
  .gitmodules
  M CMakeLists.txt
  dependencies.lock
  partitions.csv
  README.md
  sdkconfig
  sdkconfig.defaults
  sdkconfig.defaults.esp32
  sdkconfig.defaults.esp32s2
  sdkconfig.defaults.esp32s3
  sdkconfig.old

```

Message (Enter to send message to 'XIAO_ESP32S3' on '/dev/cu.usbmodem2101')

```

[0;32mI (852) cpu_start: Project name:      deploy [0m
[0;32mI (852) cpu_start: App version:          bb27ee3 [0m
[0;32mI (852) cpu_start: Compile time:         Dec 3 2024 18:58:19 [0m
[0;32mI (852) cpu_start: ELF file SHA256:     28fff01e28434653... [0m
[0;32mI (852) cpu_start: ESP-IDF:              v4.4.5 [0m
[0;32mI (853) cpu_start: Min chip rev:         v0.0 [0m
[0;32mI (853) cpu_start: Max chip rev:         v0.99 [0m
[0;32mI (853) cpu_start: Chip rev:             v0.2 [0m
[0;32mI (853) heap_init: Initializing. RAM available for dynamic allocation: [0m
[0;32mI (853) heap_init: At 3FC9B968 len 0004DDA8 (311 KiB): D/IRAM [0m
[0;32mI (854) heap_init: At 3FCE9710 len 00005724 (21 KiB): STACK/DIRAM [0m
[0;32mI (854) heap_init: At 600FE000 len 00002000 (8 KiB): RTCRAM [0m
[0;32mI (854) spiram: Adding pool of 8192K of external SPI memory to heap allocator [0m
[0;32mI (855) spi_flash: detected chip: gd [0m
[0;32mI (855) spi_flash: flash io: qio [0m
[0;32mI (857) sleep: Configure to isolate all GPIO pins in sleep state [0m
[0;32mI (857) sleep: Enable automatic switching of GPIO sleep configuration [0m
[0;32mI (858) cpu_start: Starting scheduler on PRO CPU. [0m
[0;32mI (0) cpu_start: Starting scheduler on APP CPU. [0m
[0;32mI (858) spiram: Reserving pool of 32K of internal memory for DMA/internal allocations [0m

```

```

SIGN::forward: 710316 us
19.410315, 17.129026, 16.344234, 15.779153, 16.601696, 14.735577,
Palm: 0

```



Challenges with Espressif: Flashing to device

```
flash_mode dio --flash_size detect --flash_freq 80m 0x0 bootloader/
```

⚠ Can't perform JTag flash, because OpenOCD server is not running!

❌ Failed to flash (via JTag), due to some unknown error in tcl, ⚙ ×
please try to relaunch open-ocd

Source: ESP-IDF

❌ Failed to flash the device (JTag), please try again [got response: '-1', expecting: '0']

Source: ESP-IDF

Report

Cancel

ℹ Build Successfully

Info : clock speed 40000 KHz

Info : JTAG tap: esp32s3.cpu0 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)

Info : JTAG tap: esp32s3.cpu1 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

Warn : target esp32s3.cpu1 examination failed

Info : starting gdb server for esp32s3.cpu0 on 3333

Info : Listening on port 3333 for gdb connections

Info : accepting 'tcl' connection on tcp/6666

Info : dropped 'tcl' connection

Info : [esp32s3.cpu0] Target halted, PC=0x40041A79, debug_reason=00000000

Info : [esp32s3.cpu0] Reset cause (21) - (USB UART reset)

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

❌ Error: [esp32s3.cpu1] Unexpected OCD_ID = 00000000

Overall challenges

- Setting up WiFi
- Enabling streaming data – configuring GPIO port
- Model Deployment: memory constraint, serial port connection
 - Made different choice of models
 - Resolved after bootloading multiple times
- Thermal throttling
 - Soln: heat sink
- Issue: always need to be reset
 - Acceptable given device constraint