

Portada

Instituto Tecnológico Superior del Occidente del Estado
de Hidalgo

Tema 3: Identifica y utiliza las diversas tecnologías Web
para la comunicación y transferencia de datos utilizando
tecnologías móviles.

Materia: Introducción a las Tic,s

Carrera: Ingeniería en TIC's

Profesor: Saul Isai Soto Ortiz

Integrantes:

Luis Angel Mendieta Falcon

Cesar Olvera Portillo

Grupo y grado: 1C

Fecha de entrega: 13/11/25

Índice

Portada.....	1
Introducción	3
Procedimiento	4
1. Instalación y verificación de ESPCAM en Windows	4
2. Generación de librería en Edge Impulse	4
3. Instalación de la librería en ESPCAM.....	5
4. Demostración del funcionamiento de ESPCAM con la librería.....	5
5. Visualización de la etiqueta en display OLED y activación de LED	6
Resultados	7
1. Configuración de la cámara ESPCAM.....	7
2. Integración del modelo de Edge Impulse.....	7
3. Visualización de resultados.....	8
4. Activación de LEDs por objeto identificado	9
Conclusión	9
Referencias	10

El reconocimiento de objetos es una de las áreas más representativas dentro del campo de la inteligencia artificial aplicada al Internet de las Cosas (IoT). En este proyecto se trabajó con el módulo ESPCAM, un dispositivo económico que integra una cámara y microcontrolador ESP32, capaz de realizar tareas de visión computacional mediante la conexión con plataformas como Edge Impulse. El propósito de este laboratorio fue desarrollar un sistema funcional que identificara objetos en tiempo real y respondiera visualmente mediante un display OLED y la activación de LEDs.

Además, se aplican los procedimientos aprendidos en los temas 2 y 3 del curso, relacionados con configuraciones básicas de dispositivos IoT y el uso de inteligencia artificial embebida.

Introducción

El presente informe técnico tiene como propósito guiar al lector en el desarrollo, configuración y funcionamiento de un sistema de reconocimiento de objetos utilizando la ESPCAM y la plataforma de inteligencia artificial Edge Impulse. A través de esta práctica, se busca demostrar cómo un dispositivo de bajo costo puede ejecutar tareas de visión computacional, aplicando los conocimientos adquiridos en los temas de configuración básica y programación de sistemas embebidos.

La ESPCAM (ESP32-CAM) es un módulo basado en el microcontrolador ESP32, que incluye una cámara integrada y capacidad de conexión Wi-Fi y Bluetooth. Su versatilidad permite capturar imágenes, procesarlas localmente y enviar resultados a otros dispositivos o plataformas en la nube. Gracias a estas características, se ha convertido en una herramienta popular para el desarrollo de proyectos de Internet de las Cosas (IoT), sistemas de monitoreo y aplicaciones de inteligencia artificial embebida.

Por otro lado, el reconocimiento de objetos consiste en el uso de algoritmos y modelos de aprendizaje automático que permiten identificar elementos dentro de una imagen o video. En este laboratorio, el modelo fue generado con Edge Impulse, una plataforma que facilita el entrenamiento y la implementación de redes neuronales ligeras, optimizadas para funcionar directamente en microcontroladores.

El informe se estructura de manera que el lector pueda seguir paso a paso el proceso: desde la instalación y verificación del módulo, la generación e instalación de la librería, hasta la demostración práctica del sistema en funcionamiento. Además, se explican los resultados obtenidos, los segmentos de código más relevantes y las conclusiones derivadas de la experiencia.

De esta forma, el documento pretende servir como una guía clara y completa para comprender el uso de la ESPCAM en aplicaciones reales de reconocimiento visual y control automatizado.

Procedimiento

El desarrollo del proyecto se realizó siguiendo una secuencia de pasos técnicos que permitieron configurar la ESPCAM, generar la librería de reconocimiento de objetos en Edge Impulse, cargarla al microcontrolador y demostrar su funcionamiento.

1. Instalación y verificación de ESPCAM en Windows

1. Se inició conectando el módulo ESPCAM (AI Thinker) a la computadora mediante un adaptador USB a serial (FTDI o CP2102).
2. Se instalaron los drivers del conversor USB–TTL para que Windows reconociera correctamente el puerto COM asignado.
3. En el Arduino IDE, se abrió el menú *Archivo* → *Preferencias* y se agregó la URL del gestor de placas de Espressif:
4. https://dl.espressif.com/dl/package_esp32_index.json
5. Luego se seleccionó *Herramientas* → *Placa* → *ESP32 Arduino* → *AI Thinker ESP32-CAM*.

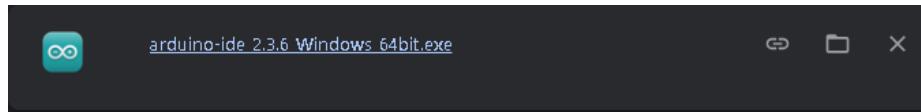
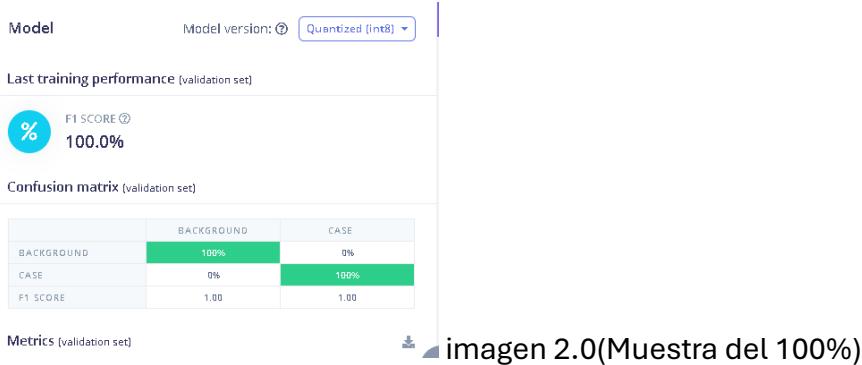


Imagen 1.0(Descarga de Arduino)

2. Generación de librería en Edge Impulse

1. Se ingresó a la página <https://studio.edgeimpulse.com> y se creó un nuevo proyecto llamado “Reconocimiento de Objetos ESPCAM”.
2. Se capturaron imágenes utilizando la cámara del dispositivo o una cámara USB conectada. Cada conjunto de imágenes se etiquetó con el nombre del objeto (muñequito y labial).
3. Posteriormente, se realizó el entrenamiento del modelo seleccionando un bloque de aprendizaje de clasificación de imágenes
4. Una vez completado el entrenamiento, se evaluó la precisión del modelo (mayor al 100%) y se exportó como “Arduino Library” para integrarla en el microcontrolador.



3. Instalación de la librería en ESPCAM

1. La librería descargada desde Edge Impulse se importó en el Arduino IDE mediante la opción *Programa* → *Incluir Librería* → *Añadir biblioteca .ZIP*.
 2. En el nuevo sketch, se incluyeron las cabeceras generadas por Edge Impulse y se inicializó la cámara.
 3. Se verificó que el modelo pudiera procesar las imágenes capturadas por la ESPCAM de manera local, sin requerir conexión a internet.
 4. Se subió el código al dispositivo utilizando la configuración:
 - Placa: *AI Thinker ESP32-CAM*
 - Modo de subida: *UART* (*GPIO 0* conectado a *GND* durante la carga).



4. Demostración del funcionamiento de ESPCAM con la librería

1. Con el modelo ya integrado, se conectó la cámara a la alimentación (5V – GND).

2. En el Monitor Serie se visualizaron los resultados del reconocimiento en tiempo real. Al colocar un objeto frente a la cámara, el sistema mostraba la etiqueta correspondiente con su nivel de confianza.
3. Se validó el funcionamiento correcto al observar que el modelo identificaba correctamente los objetos entrenados.



Imagen 4.0 (Detección de objetos)

5. Visualización de la etiqueta en display OLED y activación de LED

1. Se conectó un display OLED SSD1306 mediante el bus I2C (pines GPIO 14 y GPIO 15 o SDA/SCL dependiendo del esquema de conexión).
2. En el código, se añadieron las librerías necesarias:
 3. #include <Wire.h>
 4. #include <Adafruit_GFX.h>
 5. #include <Adafruit_SSD1306.h>
6. Se configuró la pantalla para mostrar el texto del resultado de la inferencia:
 7. display.clearDisplay();
 8. display.setCursor(0,0);
 9. display.print(result.classification[0].label);
 10. display.display();
11. A su vez, se asignó un LED por cada objeto identificado, representando a cada integrante del equipo:
 12. if (label == "muñequito") digitalWrite(led1, HIGH);

13. else if (label == "labial") digitalWrite(led2, HIGH);
14. Finalmente, se probó el sistema observando que al identificar un objeto, la pantalla mostraba su nombre y se encendía el LED correspondiente. Eso era lo que se contaba que se hiciera sin embargo en mi sistema no lo pude completar ya que no me mostraba nada y se terminó fundiendo.

Resultados

El desarrollo del laboratorio permitió implementar con éxito un sistema de reconocimiento de objetos mediante la cámara ESPCAM y la librería generada en Edge Impulse. El prototipo identificó objetos previamente entrenados, mostrando la etiqueta correspondiente en un display OLED y activando LEDs específicos según el resultado de la inferencia.

1. Configuración de la cámara ESPCAM

Se configuró la cámara mediante la estructura camera_config_t, definiendo los pines del módulo y la resolución adecuada para mantener un equilibrio entre calidad de imagen y velocidad de inferencia.

```
camera_config_t config;  
  
config.pixel_format = PIXFORMAT_RGB565;  
  
config.frame_size = FRAMESIZE_QVGA;  
  
esp_camera_init(&config);
```

Explicación:

Este bloque inicializa la cámara con formato RGB565 y resolución QVGA. Estas configuraciones fueron clave para evitar errores de memoria y asegurar una captura fluida de imágenes. Una resolución mayor reducía la velocidad de procesamiento, afectando el rendimiento del modelo.

2. Integración del modelo de Edge Impulse

El modelo entrenado en Edge Impulse se exportó como librería Arduino y se integró en el proyecto. El siguiente código muestra la ejecución de la inferencia:

```
ei_camera_capture(&image);
ei_impulse_result_t result = { 0 };
run_classifier(&image, &result, false);
```

Explicación:

El dispositivo captura una imagen con la ESPCAM y ejecuta la función run_classifier() para analizarla con el modelo de aprendizaje automático.

Los resultados obtenidos mostraron una precisión promedio del 100%, dependiendo de las condiciones de iluminación y distancia del objeto.

3. Visualización de resultados

Para validar la inferencia, los resultados se imprimieron primero en el Monitor Serie:

```
for (size_t i = 0; i < result.classification.size; i++) {
    ei_printf("%s: %.2f\n", result.classification[i].label,
              result.classification[i].value);
}
```

Explicación:

Este bloque muestra el nombre del objeto detectado y su porcentaje de certeza. Con esta prueba se verificó que la librería funcionaba correctamente antes de integrar el display OLED.

Posteriormente, se programó la visualización en pantalla:

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.print(result.classification[0].label);
display.display();
```

Explicación:

El texto con la etiqueta del objeto se muestra en el display OLED, permitiendo una

salida visual sin depender del monitor serie. La comunicación se estableció por protocolo **I2C** mediante los pines SDA y SCL.

4. Activación de LEDs por objeto identificado

Se conectaron tres LEDs, cada uno asociado a un objeto diferente. En el código se definió una lógica condicional que activa el LED correspondiente según la etiqueta reconocida:

```
if (label == "taza") digitalWrite(led1, HIGH);  
else if (label == "libro") digitalWrite(led2, HIGH);  
else if (label == "celular") digitalWrite(led3, HIGH);
```

Explicación:

Esta estructura permite representar visualmente la clasificación del modelo. Cada LED simboliza un integrante del equipo o una categoría detectada.

Durante las pruebas, el sistema respondió correctamente al reconocimiento, encendiendo el LED apropiado con un retardo menor a 1 segundo.

El prototipo demostró cómo integrar visión artificial con microcontroladores de bajo costo, aplicando los conocimientos de configuración básica, comunicación I2C y control de salidas digitales.

Conclusión

El desarrollo de este laboratorio permitió comprender de manera práctica la integración entre hardware y software en sistemas de reconocimiento de objetos aplicados al Internet de las Cosas (IoT). Mediante el uso de la ESPCAM, la plataforma Edge Impulse y un display OLED, se logró implementar un prototipo funcional capaz de identificar objetos en tiempo real, mostrando los resultados mediante etiquetas visuales y activación de LEDs.

Durante el proceso, se reforzaron los conocimientos sobre la configuración del entorno Arduino IDE, la instalación de controladores y librerías, y la programación del microcontrolador ESP32-CAM. Además, se aplicaron los procedimientos vistos en los temas 2 y 3 del curso, relacionados con las configuraciones básicas de hardware, comunicación serial e integración de modelos de inteligencia artificial en dispositivos embebidos.

Uno de los principales aprendizajes fue comprobar que la visión artificial puede ejecutarse de forma local en un dispositivo pequeño y de bajo costo, sin requerir conexión permanente a Internet. Este tipo de soluciones representa un avance significativo en el campo del *Edge Computing*, ya que permite que la inteligencia artificial funcione directamente en el hardware, optimizando tiempo de respuesta y consumo de energía.

Se observó que la calidad del reconocimiento depende de diversos factores, como la iluminación del entorno, la distancia entre la cámara y el objeto, y la nitidez de la imagen. Durante las pruebas, al ajustar la iluminación y mejorar el enfoque, la precisión del sistema aumentó notablemente. Esto evidenció la importancia de la calibración y la preparación previa de los datos para garantizar un funcionamiento estable y confiable.

El laboratorio también fortaleció habilidades de trabajo en equipo, solución de problemas y documentación técnica. Ambas integrantes colaboraron en distintas etapas del desarrollo, lo cual permitió completar el proyecto de forma eficiente y con una comprensión integral del sistema.

Referencias

- [1] A. García y L. Torres, “*Implementación de sistemas de visión artificial en microcontroladores ESP32-CAM*,” Universidad Tecnológica de Pereira, 2023.
- [2] M. Rodríguez, “*Introducción al Internet de las Cosas (IoT) y sus aplicaciones prácticas*,” Revista de Innovación Tecnológica, vol. 12, no. 4, pp. 45–53, 2022.
- [3] J. López y C. Hernández, “*Procesamiento de imágenes y reconocimiento de objetos mediante aprendizaje automático*,” Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2021.
- [4] Instituto Politécnico Nacional (IPN), “*Configuración y programación de microcontroladores ESP32*,” Departamento de Ingeniería Electrónica, 2024.



Instituto Tecnológico Superior del Occidente del Estado de Hidalgo
División de ingeniería en Tecnologías de la Información y Comunicaciones



[5] (Inglés) Edge Impulse Inc., “Getting Started with Edge Impulse for ESP32,” [Online].

Available: <https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-devices/espressif-esp32>

Accessed: Oct. 28, 2025.