

SenseCAP A1101

Chavez Ana

22.01.24

Índice

Introducción.....	1
Características.....	1
LED de estado funcionamiento del sensor.....	2
Modelos de IA integrados para A1101.....	3
Desarrollo.....	4
SENSECAP. Conexión del sensor con SENSECAP.....	4
Conexión con plataforma TTN.....	9
Conexión con la plataforma Datacake.....	11
Entrena tu propio modelo de IA con un conjunto de datos público.....	14
Modelo de reconocimiento de dígitos del medidor de agua con SenseCAP A1101.....	17
1. Recopilación de datos.....	18
2. Generar el conjunto de datos con RoboFlow.....	24
3. Implementar el modelo entrenado y realizar la inferencia.....	38
4. Realizar inferencia con SenseCAP A1101 en SenseCAP Mate.....	40
Conectar Gateway.....	42
Conclusión.....	45
Bibliografía.....	46

Introducción

El sensor de IA de visión SenseCAP A1101 LoRaWAN combina la tecnología de IA TinyML y la transmisión de largo alcance LoRaWAN para permitir una solución de dispositivo de IA de bajo consumo y alto rendimiento. La tecnología de IA TinyML significa que los modelos de aprendizaje automático están optimizados para ejecutarse en dispositivos de muy bajo consumo y tamaño reducido.

Este sensor admite diversos modelos de IA, como reconocimiento de imágenes, conteo de personas, detección de objetos, etc.

Su sencilla configuración permite que los usuarios se enfoquen directamente en el desarrollo de aplicaciones, sin complicaciones. Solo se necesitan unos pocos pasos: instalar el dispositivo, vincularlo mediante el código QR y configurar la red. A partir de ahí, los datos recopilados pueden visualizarse fácilmente desde el portal SenseCAP, que admite protocolos IoT populares como HTTP y MQTT.

Características

- Cámara Himax ultrapotente y potente: DSP de 400Mhz, frecuencia de imagen máxima de la cámara 640*480*VGA 60 FPS, inferencias locales
- Transmisión de Bajo Consumo y Largo Alcance: Hasta 2.3uWh de consumo de energía en modo de reposo, alimentado por módulos Wio-E5 LoRaWAN, transfiere datos de hasta millas.
- Alta seguridad de los datos mediante Edge Computing: Inferencia de imagen local y transfiere los datos del resultado final a la Nube, adecuado para aplicaciones que requieren transmisión de datos limitada y alta privacidad de datos.
- Datos fáciles de visualizar: Pocos clics para visualizar y gestionar los datos a través de SenseCAP Mate App y SenseCAP Dashboard, amplia compatibilidad con otras herramientas de terceros.
- Alto grado de protección industrial: temperatura de funcionamiento de -40 ~ 85 °C y clasificación IP66, apto para despliegue en interiores y exteriores. La abreviatura IP significa protección contra la intrusión, esto indica que un dispositivo con clasificación IP66 tiene un alto grado de protección contra polvo y agua.
- Fácil despliegue escalable: 1 min para añadir y configurar el dispositivo escaneando el código QR del dispositivo, bajo coste de red LoRaWAN y de mantenimiento garantizan la escalabilidad del negocio.

El dispositivo A1101 incorpora el módulo Wio-E5, un dispositivo LoRaWAN de bajo costo, ultra baja potencia, extremadamente compacto y de alto rendimiento diseñado por Seeed Technology Co., Ltd. El Wio-E5 contiene el chip STM32WLE5JC, el primer SoC del mundo que integra un combo de chip LoRa RF y un MCU. Este módulo también incluye un MCU de potencia ultrabaja ARM Cortex M4 y LoRa SX126X, y por lo tanto admite el modo (G) FSK y LoRa.



Imagen. Módulo Wio-E5

LED de estado funcionamiento del sensor



Imagen 1. Botón de dispositivo

- Encender y activar el Bluetooth
Mantén pulsado durante 9s
El LED parpadea con una frecuencia de 1s, esperando la conexión Bluetooth.
Si el Bluetooth no se conecta en 1 minuto, la máquina se apagará de nuevo.
- Reinicie el dispositivo y únase red LoRa
Pulse una vez
 - 1. El LED se encenderá durante 15 segundos para inicialización
 - 2. Esperando para unirse red LoRa: luz intermitente
 - 3. Unirse a la red LoRa éxito: LED parpadea rápido durante 2s
 - 4. Unión a la red LoRa fallo: LED se para de repente.
- Activar Bluetooth de nuevo
Mantén pulsado durante 3s
 - 1. Esperando conexión Bluetooth: El LED parpadea con una frecuencia de 1s
 - 2. Acceder al modo de configuración después de que la conexión Bluetooth se haya realizado correctamente: El LED parpadea con una frecuencia de 2s
 - Si el Bluetooth no se conecta en 1 minuto, el dispositivo se reiniciará y se unirá a la red LoRa.

- Apagado
Mantener pulsado durante 9s
En el 3er segundo empezará a parpadear a una frecuencia de 1s, hasta que la luz se quede fija, suelte el botón, la luz se apagará .
Nota: Despues de apagar, es necesario volver a configurar la banda de frecuencia. Se recomienda apagarlo cuando no se utilice.

Modelos de IA integrados para A1101

Algoritmo	Modelo AI
Detección de Objetos	Detección del Cuerpo Humano; Definido por el Usuario
Conteo de Objetos	Contando Personas; Definido por el usuario
Clasificación de Imagen	Reconocimiento de Persona y Panda; Definido por el usuario

Existen 2 formas de entrenar un modelo de IA

1. One Stop Model Training con Edge Impulse.
Edge Impulse es la plataforma de inteligencia artificial en el borde para equipos empresariales que construyen modelos optimizados en cualquier dispositivo periférico. Entrega valor de manera más rápida y logra innovación de productos con funciones avanzadas de sensores de inteligencia artificial.
2. Modelo de IA para entrenar con Roboflow, YOLOv5, TensorFlow Lite. Este es un método lento de seguir.
Roboflow es una herramienta de anotación basada en línea, facilita la tarea de etiquetar imágenes, añadir procesamiento adicional a estas y exportar conjuntos de datos etiquetados en diversos formatos, como YOLOv5 PyTorch y Pascal VOC. Por otro lado, YOLOv5, qué significa 'You Only Look Once', es un algoritmo que permite la detección y reconocimiento de múltiples objetos en tiempo real. En cuanto a TensorFlow Lite, se trata de un marco de aprendizaje profundo multiplataforma y de código abierto que convierte modelos preentrenados en TensorFlow a un formato especial optimizable para velocidad o almacenamiento.

Desarrollo

SENSECAP. Conexión del sensor con SENSECAP

Además de poder conectarse directamente a una computadora para visualizar los datos de detección en tiempo real, también se tiene la opción de transmitir estos datos a través de LoRaWAN y cargarlos en la plataforma en la nube de SenseCAP.

1. Descargar SenseCAP Mate App
2. Conectar el sensor a la App por medio de Bluetooth
Pulsar el icono “+” ->escanear código QR del sensor
Seleccionar "Sensor de Visión AI"
Pulsar el botón del dispositivo durante 3 segundos hasta que el indicador parpadee lentamente.

Seleccionar el sensor por número de serie



Imagen 2. Código QR de dispositivo

3. Configurar los parámetros necesarios

Información

- **Modelo del dispositivo:** SenseCAP A1101
- **EUI del dispositivo:** 2CF7F1C0528000C3
Identificación única del dispositivo
- **Tipo de Sensor:** Vision AI Sensor
- **Algoritmo:** Image Classification
- **Modelo de IA:** Person & Panda Recognition
- **Versión de Firmware de Respaldo:** 2.0
- **Versión Software:** 2.3
- **Versión Hardware:** V1.2
- **Versión LoRaWAN:** V1.0.3
- **Tipo de clase:** Class A

Hay 3 tipos de clase:

1. Clase A: Los dispositivos de clase A admiten comunicación bidireccional entre un dispositivo y una puerta de enlace. Los

mensajes de enlace ascendente se pueden enviar en cualquier momento.

2. Clase B: Los dispositivos de Clase B amplían la Clase A agregando ventanas de recepción programadas para mensajes de enlace descendente desde el servidor. Utilizando balizas sincronizadas en el tiempo transmitidas por la puerta de enlace, los dispositivos abren periódicamente ventanas de recepción.
3. Clase C: Los dispositivos de Clase C extienden la Clase A manteniendo las ventanas de recepción abiertas a menos que estén transmitiendo. Esto permite una comunicación de baja latencia pero consume mucha más energía que un dispositivo de Clase A.

- **Batería**

Configuraciones

- **Plataforma:** The Things Network
- **Algoritmo:** Image Classification
- **Modelo de IA:** Person & Panda Recognition
- **Umbral de Puntuación:** 0.5

Las puntuaciones representan la fiabilidad de las predicciones del modelo de IA. Cuanto mayor sea la puntuación, más probable es que el objeto identificado sea el objeto objetivo. El Umbral de Puntuación varía de 1 a 100, y solo se cargarán en la aplicación y la nube los resultados con una puntuación mayor que el Umbral de Puntuación.

- **Plan de Frecuencia:** US915
- **Intervalo de Subida (min):** 10
Es el intervalo de tiempo para cargar datos.
- **Tipo de activación:** OTAA (Activación por Aire)
- **Device EUI:** 2CF7F1C0528000C3
Es la identificación única del dispositivo
- **AppEUI:** 553C0DD8086559C9
Es la identificación única de la aplicación.
- **AppKey:** 007AF6834C293DF6CDC9369B4
Es la clave de aplicación.
- **Política de Paquetes:** 2C+1N

Hay 3 parámetros:

1. 2C+1N (2 paquetes de confirmación y 1 de no confirmación) es la mejor estrategia, el modo puede minimizar la tasa de pérdida de paquetes, sin embargo el dispositivo consumirá la mayor cantidad de paquetes de datos en TTN, o créditos de fecha en la red Helium.
2. 1C (1 confirmación) el dispositivo dormirá después de recibir 1 paquete de confirmación del servidor.

3. 1N (1 sin confirmación) el dispositivo sólo envía el paquete y luego el servidor recibe los datos o no.

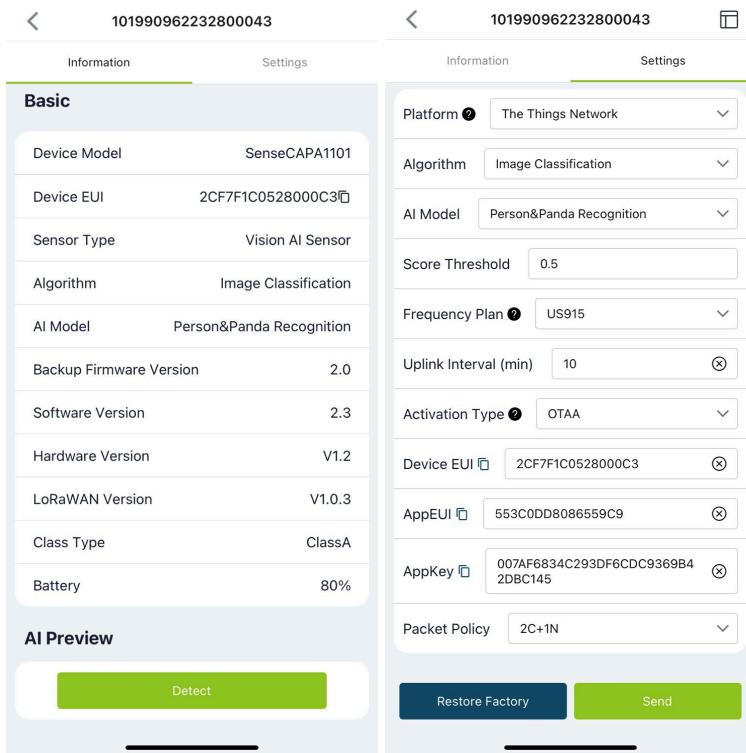


Imagen 3. Información del dispositivo

Dar clic en el botón "General" y, a continuación, en "Detectar". Apuntar la lente hacia el objeto de destino, ajustar el ángulo y la distancia adecuadamente, y el resultado del reconocimiento se mostrará en la parte inferior de la pantalla.

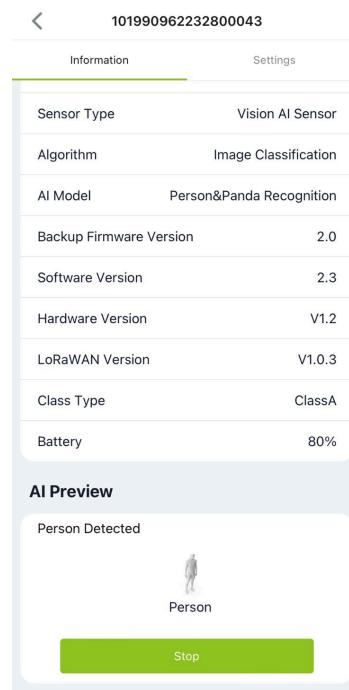


Imagen 4. Sensecap Mate App: Detectar persona

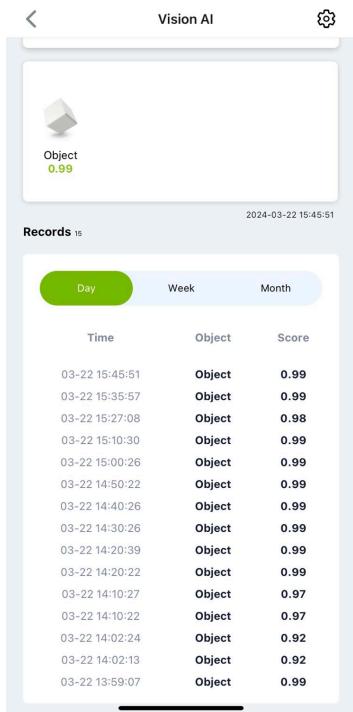


Imagen 5. Datos obtenidos en Sensecap Mate App

4. Acceder a la plataforma de SenseCAP para poder observar más sobre los datos obtenidos. Entrar a la pagina <https://sensecap.seeed.cc/portal/#/login>
En la pestaña del nodo del sensor se pueden observar estos datos.

- General Information (Información general)

Se puede configurar el nombre del dispositivo y verificar el estado del dispositivo, el estado de la batería, el registro en línea reciente, etc.

Battery status: es el estado de la batería. Si se muestra "Batería insuficiente", significa que la batería está por debajo del 10%.

Binding Time: Se refiere al período durante el cual dos dispositivos están conectados entre sí.

Last Message Time : Indica el momento en el que se recibió o se envió el último mensaje.

Service Countdown : Indica el tiempo restante para la finalización de un servicio.

The screenshot shows the SENSECAP Node Details page. The left sidebar has sections for Dashboard, Devices (selected), Gateway, Node Group, Sensor Node (selected), Development Kit, Data (selected), Custom Type, Measurement, Security, and Access API keys. The main content area shows the following details:

- General Information** tab selected.
- Device Name:** Vision AI
- Node EUI:** 2CF7FC0528000C3
- Network Standard:** LoRaWAN
- Online Status:** Offline
- Frequency(MHz):** EU868
- Battery Status:** Better Battery
- Binding Time:** 2024-02-27 15:34:48
- Last Message Time:** 2023-07-20 22:44:58
- Hardware Version:** 1.2
- Firmware Version:** 2.0
- Service Status:** In Service
- Service Countdown:** 182 days
- Service Purchase:** button

Imagen 6. Información general del dispositivo

- Channel (Canal)

Dispositivo de un solo canal: Nodo sólo puede conectar un sensor

Los diferentes tipos de mediciones son las siguientes:

AI Detection No.10-4184 / AI Detection No.09-4183/AI Detection No.08-4182/ AI Detection No.07-4181 /AI Detection No.06-4180 / AI Detection No.05-4179 / AI Detection No.04-4178 / AI Detection No.03-4177/ AI Detection No.02-4176 /AI Detection No.01-4171

The screenshot shows the SENSECAP Node Details page with the Channel tab selected. The main content area shows the following table:

Channel Number	Channel Name	Channel Status	Model No.	Algorithm	Sensor Name	Measurement Name	Prod
1	✓	Normal			Vision AI	AI Detection No.10-4184 / AI Detection No.09-4183 3 / AI Detection No.08-4182 / AI Detection No.07-4181 81 / AI Detection No.06-4180 / AI Detection No.05-4179 / AI Detection No.04-4178 / AI Detection No.03-4177 / AI Detection No.02-4176 / AI Detection No.01-4171	-4175

Imagen 7. Canal

- Data (Datos)

Puede consultar los datos de medición del sensor.

The screenshot shows the SENSECAP Node Details page with the Data tab selected. The main content area shows the following table:

Time	AI Detection No.01-4175	Total of 20 data, up to 15 per page
2024-02-27 16:50:00	0.98	
2024-02-27 16:45:00	0.98	

Below the table, there are tabs for **Channel** and **Measurement Data**. The **Measurement Data** tab is selected.

Imagen 8. Datos

Conexión con plataforma TTN

Crear una aplicación en la plataforma TTN.

Seleccionar consola -> Aplicación -> + Crear aplicación

Asignar un Id aplicación el cual debe de ser único, ya que este identificador permite distinguirla de otras en la plataforma

Añadir sensor a la consola de la plataforma TTN

En Aplicación -> Dispositivos Finales -> Añadir dispositivo final

Al no identificarse el dispositivo en el repositorio de dispositivos LoRaWAN, se decidió agregarlo manualmente.

- Plan de frecuencia: United States 902-928 MHz, FSB2 (used by TTN)
- LoRaWAN version: LoRaWAN Specification 1.0.3.
Este dato fue obtenido de SenseCAP Mate App
- Versión de los parámetros regionales: RP001 Regional Parameters 1.0.3 revision A

Los siguientes parámetros se obtuvieron de la aplicación SenseCAP Mate App

- JoinEUI: 553C0DD8086559C9
- DevEUI: 2CF7F1C0528000C3
- AppKey: 007AF6834C293DF6CDC9369B4
- Id dispositivo final: swm-vision-01

The screenshot shows the 'Register end device' form on the TTN platform. It is divided into two main sections: 'Provisioning information' on the right and 'End device type' on the left.

Provisioning information:

- JoinEUI:** 55 3C 0D D8 08 65 59 C9 (with a 'Reset' button)
- DevEUI:** 2C F7 F1 C0 52 80 00 C3 (with a 'Generate' button and note '1/50 used')
- AppKey:** 00 7A F6 83 4C 29 3D F6 CD C9 36 9B 42 DB C1 45 (with a 'Generate' button)
- End device ID:** swm-vision-01 (Note: This value is automatically prefilled using the DevEUI)

End device type:

- Input method:** Enter end device specifics manually (selected)
- Frequency plan:** United States 902-928 MHz, FSB 2 (used by TTN)
- LoRaWAN version:** LoRaWAN Specification 1.0.3
- Regional Parameters version:** RP001 Regional Parameters 1.0.3 revision A

After registration:

- View registered end device
- Register another end device of this type

Buttons:

- Scan end device QR code
- Device registration help
- Register end device

Imagen 9 Registrar dispositivo final

Se pueden observar los siguientes datos:

Imagen 10. Información general en plataforma TTN

En la pestaña Payload formatters -> Uplink

Seleccionar el tipo de formateador: Custom Javascript formatter

Pegar el código que se encuentra en el siguiente enlace:

https://github.com/Seeed-Solution/TTN-Payload-Decoder/blob/master/decoder_new-v3.js

Este código es un convertidor que se utiliza para interpretar los datos recibidos del dispositivo, extrayendo información sobre la versión del firmware, el estado de la batería, la identificación del sensor, las mediciones, entre otros datos.

Imagen 11. Live Data

Nota: Se ha observado que no es posible utilizar simultáneamente los servidores SenseCAP y TTN, ya que cuando se está utilizando uno, los datos no se actualizan en el otro.

Conexión con la plataforma Datacake

Para facilitar la visualización de los datos recibidos, se utilizará la plataforma de Datacake.

1. Añadir dispositivo LoRaWAN al Dashboard

- En la plataforma *Datacake*, dar clic en el botón " + Añadir dispositivo"
- Seleccionar *LoRaWAN* y dar clic en "siguiente"
- Paso 1. Producto
Datacake Producto.
Seleccionar "*Nuevo producto de plantilla*"
Plantilla de dispositivo
En la sección "Todos los fabricantes", seleccionar Seeed.
Seleccionar plantilla "*Seeed SenseCAP A1101*"
Dar clic en siguiente
- Paso 2. Servidor de red
Seleccionar el Servidor de Red *LoRaWAN* al que está conectado el dispositivo: The Things Stack V3
Dar clic en siguiente
- Paso 3. Dispositivos
Llenar los datos que se solicitan del dispositivo
DEVEUI → EUI del dispositivo (Puede ser copiado de la plataforma TTN)
Nombre → swm-vision-01
Dar clic en siguiente
- Paso 4. Plan
Seleccionar plan que utilizara para este dispositivo: "*Free*"
Nota: La plataforma de *Datacake* te brinda la posibilidad de tener hasta 5 espacios de trabajo de forma gratuita. Si se desea agregar más están los siguientes precios.

Free	Light	Standard	Plus
0,00 € / month	1,00 € / month	3,00 € / month	5,00 € / month
7 days data retention	1 month data retention	3 months data retention	12 months data retention
500 datapoints / day	1000 datapoints / day	2500 datapoints / day	7500 datapoints / day
Máx 5 por workspace	Cancelar en cualquier momento	Cancelar en cualquier momento	Cancelar en cualquier momento
Cancelar en cualquier momento			

Imagen 12. Planes
Dar clic en "Añadir dispositivo"



Imagen 13. Dispositivo agregado

2. Conexión de Things stack a Datacake

Crear integración TTI

- En la plataforma TTN, seleccione "Integraciones" en la barra lateral.
- Navegue en la sección "Webhooks" de las integraciones
- Agregue un nuevo Webhook haciendo clic en el botón "+ Agregar Webhook".
- Esto mostrará una lista de plantillas de webhook disponibles entre las que puede elegir: Seleccione la plantilla webhook "*Datacake*"

Clave API en *Datacake* para la autenticación de Webhook

- En la plataforma *Datacake*, seleccione "Miembros" en la barra lateral
- En la sección de miembros, seleccione "Usuarios de API".
- Copiar Token de *Datacake*

En la plataforma TTN

- Pegar el Token copiado
- Dar clic en el botón "Crear Datacake webhook"

Downlinks

- En la plataforma *Datacake*, ir a la configuración de LoRaWAN en el dispositivo correspondiente
- En la sección LoRaWAN, en el servidor de red hacer clic en "cambio" para abrir la configuración del *downlink*.
- Servidor de red
Seleccionar "The Things Stack V3"

- Rellenar lo que se solicita con la información que se encuentra en la plataforma TTN.

ID de dispositivo TTS → swm-vision-01

URL del servidor TTI → eu1.cloud.thethings.network

App ID de TTI → sensor-a1101

Clave API TTI

Crear clave API TTI

En la plataforma TTN, seleccione "API keys" en la barra lateral izquierda.

Dar clic en "Add API key"

Darle un nombre a la llave

Habilitar los permisos:

- Write downlink application traffic.
- Read application traffic (uplink and downlink)
- Write uplink application traffic

Dar clic en "Create API key"

Copiar la clave y pegarla en la configuración del dispositivo en Datacake

- Dar clic en el botón "Actualizar"
- Dar clic en el botón "Guardar"

En la siguiente imagen se puede observar el Dashboard.

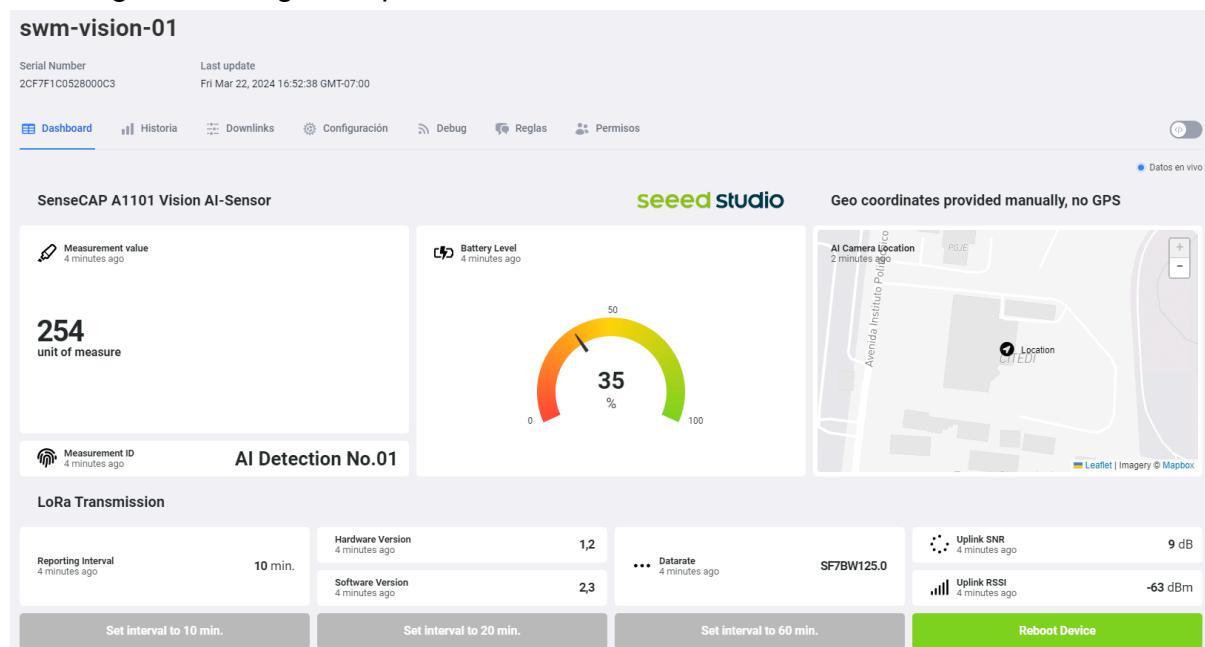


Imagen 14. Dashboard

Entrena tu propio modelo de IA con un conjunto de datos público

Obtener datos para el entrenamiento.

Utilizar conjuntos públicos ahorra tiempo al evitar la recopilación y anotación manual de datos, ya que estos conjuntos están listos para ser utilizados, permitiendo concentrarse en las aplicaciones de visión de inteligencia artificial.

Se pueden descargar varios conjuntos de datos disponibles públicamente, como COCO y VOC de Pascal, a través de plataformas como Roboflow Universe. También se puede buscar conjuntos de datos de código abierto en Google.

1. Visita esta URL para acceder a un conjunto de datos de Apple Detection disponible públicamente en Roboflow Universe
<https://universe.roboflow.com/lakshantha-dissanayake/apple-detection-5z37o/dataset/1>
2. Crear una cuenta de Roboflow
3. Haga clic Descargar, seleccione Pytorch YOLO v5 como el Formato, haga clic mostrar código de descarga y haga clic Continuar
Esto generará un fragmento de código que se usará más adelante dentro del entrenamiento de Google Colab.

Entrenar usando YOLOv5 en Google Colab

Se utiliza el entorno de Google Colaboratory para realizar entrenamientos en la nube. Además, se utiliza la API de Roboflow dentro de Colab para facilitar la descarga del conjunto de datos.

1. Hacer clic [aqui](#) para abrir un espacio de trabajo de Google Colab ya preparado, siga los pasos mencionados en el espacio de trabajo y ejecute las celdas de código una por una.
Nota: En Google Colab, en la celda de código del Paso 4, debe copiar directamente el fragmento de código de Roboflow que se mencionó anteriormente.

Entrenamiento:

1. Instalar las dependencias de YOLOv5
(Recuerda elegir GPU en el entorno si aún no está seleccionado. Entorno de ejecución --> Cambiar tipo de entorno de ejecución --> Acelerador de hardware --> GPU)
2. Descargar datos personalizados para la detección de objetos YOLOv5
Descargar conjunto de datos desde Roboflow. Utilizar el formato de exportación "YOLOv5 PyTorch". Tener en cuenta que la implementación de Ultralytics requiere un archivo YAML que defina la ubicación de tus datos de entrenamiento y prueba. La exportación de Roboflow también genera este formato.

3. Escribir nuestra configuración de entrenamiento YOLOv5

Escribir un script YAML que defina los parámetros para el modelo, el número de clases, anclajes y cada capa.

Descargar el modelo de preentrenamiento

4. Ejecutar el entrenamiento de YOLOv5

img: define el tamaño de la imagen de entrada

batch: determina el tamaño del lote

epochs: define el número de épocas de entrenamiento.

data: establece la ruta de nuestro archivo YAML

cfg: especifica la configuración de nuestro modelo

weights: especifica una ruta personalizada para los pesos. (Nota: puedes descargar pesos desde la carpeta de Google Drive de Ultralytics)

name: nombres de los resultados

nosave: guardar solo el punto de control final

cache: almacenar en caché imágenes para un entrenamiento más rápido

5. Evaluar el rendimiento de YOLOv5

Las pérdidas de entrenamiento y las métricas de rendimiento se guardan en Tensorboard y también en un archivo de registro definido anteriormente con la bandera --name cuando entrenamos. En nuestro caso, nombramos esto como yolov5s_results. (Si no se proporciona un nombre, el valor predeterminado es results.txt). El archivo de resultados se representa como un archivo PNG después de que se complete el entrenamiento.

Nota de Glenn: Los archivos results.txt parcialmente completados se pueden representar con from utils.utils import plot_results; plot_results().

6. Visualizar los datos de entrenamiento de YOLOv5

Después de que comience el entrenamiento, visualiza las imágenes train*.jpg para ver las imágenes de entrenamiento, las etiquetas y los efectos de aumentación.

7. Ejecutar inferencias de YOLOv5 en imágenes de prueba

Realiza la inferencia con un punto de control preentrenado en el contenido de la carpeta test/images descargada de Roboflow.

8. Descarga del modelo entrenado para futuras inferencias

Ahora que has entrenado tu detector personalizado, puedes exportar el modelo entrenado que has hecho aquí para inferirlo en tu dispositivo en otro lugar

```
print("Download your custom model")
```

Descargar los archivos de pesos entrenados en el siguiente orden:

1. Busque la opción de archivo en el menú a la derecha de la colab
 2. Seleccione el archivo .uf2 generado anteriormente
 3. Pulsar el botón derecho y seleccione Descargar
9. Despliegue su modelo en los dispositivos Grove AI
- Instale la última versión del navegador
 - Conecte su dispositivo Grove AI a su PC.
 - Descarga el modelo en tu dispositivo Grove AI
 - Haga doble clic en el botón de arranque
Puede ver una etiqueta GROVEBOOT en su pc que puede mover la unidad.
 - Arrastre y suelte el archivo .uf2 que ha descargado en su disco extraíble GROVEBOOT
 - Haga clic en el botón Connect y seleccione Grove AI Camera.
 - Vea los resultados especulativos en tiempo real a través de la ventana de vista previa

Modelo de reconocimiento de dígitos del medidor de agua con SenseCAP A1101

Preparación de software

- Roboflow - para anotar. Roboflow es una plataforma que brinda todas las herramientas necesarias para convertir imágenes en información mediante la visión por computadora. Esta herramienta ayuda a organizar y nombrar a las imágenes para que la computadora pueda entenderlas mejor.
- Asistente de modelo SenseCraft - para entrenar. Esta herramienta ayuda a enseñarle a la computadora a hacer cosas específicas, como reconocer patrones en datos.
- TensorFlow Lite - para inferencias. TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Esta es una herramienta que ayuda a la computadora a tomar decisiones basadas en lo que ha aprendido, además de realizar predicciones.



*Imagen 15. Software.
Obtenida de: <https://goo.su/i8ssh3i>*

Configuracion Software

1. Instalar la última versión de Python en la computadora

2. Instalar dependencia

Antes de instalar la dependencia, es importante asegurarse de marcar la casilla correspondiente en windows para agregar python al PATH del sistema.

Para hacerlo, debes editar las variables del entorno del sistema, ir a "Variables de entorno", seleccionar las variables del sistema y buscar "Path". Luego, haz clic en "Editar", añade una nueva entrada, y pega la ruta `C:\Users\Titular\AppData\Local\Programs\Python\Python312\Scripts`.

Guarda los cambios y cierra las ventanas de configuración.

Para instalar un paquete Python usando pip3, debes ejecutar el comando en el shell de tu sistema (Símbolo del sistema)

Windows + R → cmd → enter → copiar y pegar lo siguiente

`pip3 install libusb1`

Pip3 es el sistema de gestión de paquetes predeterminado para Python 3. Es una herramienta que permite a los usuarios instalar, actualizar, eliminar y buscar paquetes de Python desde el Python Package Index (PyPI).

```
Microsoft Windows [Versión 10.0.19045.3930]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Ana>pip3 install libusb1
Collecting libusb1
  Downloading libusb1-3.1.0-py3-none-win_amd64.whl (140 kB)
    140.4/140.4 kB 4.1 MB/s eta 0:00:00
Installing collected packages: libusb1
Successfully installed libusb1-3.1.0

[notice] A new release of pip is available: 23.1.2 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\Ana>
```

Imagen 16. Instalar dependencia

Pasos:

1. Recopilación de datos

- Conecte SenseCAP A1101 a la PC mediante un cable USB tipo C



Imagen 17. Conectar dispositivo

Obtenida de: <https://goo.su/i8ssh3i>

- Hacer doble clic en el botón de inicio para ingresar al modo de arranque



Imagen 18. Modo arranque.

Obtenida de: <https://goo.su/i8ssh3i>

Después de esto, verá una nueva unidad de almacenamiento que se muestra en su explorador de archivos como SENSECAP.

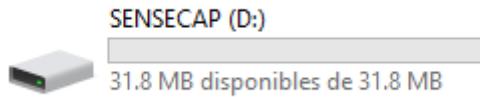


Imagen 19. Unidad de almacenamiento

- c. Arrastre y suelte el siguiente [archivo .uf2](#) en la unidad SENSECAP
En cuanto el uf2 termine de copiarse en la unidad, ésta desaparecerá.
Esto significa que el uf2 se ha cargado correctamente en el módulo.

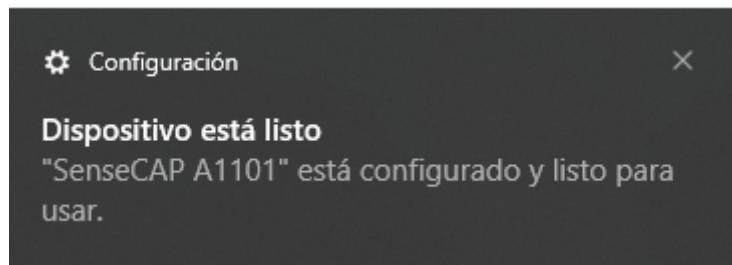


Imagen 20. Confirmación de configuración

- d. Copie y pegue este [script Python](#) dentro de un archivo recién creado llamado **capture_images_script.py** en su PC

Código script Python:

```

import os
import usb1
from PIL import Image
from io import BytesIO
import argparse
import time
import cv2
import numpy as np
from threading import Thread

WEBUSB_JPEG_MAGIC = 0x2B2D2B2D
WEBUSB_TEXT_MAGIC = 0x0F100E12

VendorId = 0x2886 # seeed studio
ProductId = [0x8060, 0x8061]

# La clase Receive_Mess maneja la recepción de datos desde el dispositivo USB
class Receive_Mess():
    def __init__(self, arg, device_id):
        self.showimg = not arg.unshow # Visualización de imágenes
        self.saveimg = not arg.unsave # Almacenamiento de imágenes
        self.interval = arg.interval
        self.img_number = 0
        self.ProductId = []
        os.makedirs("./save_img", exist_ok=True) // Se crea un directorio "save_img" para guardar las imágenes

    # Se inicializan las variables para el manejo de datos
    self.expect_size = 0
    self.buff = bytearray()

    self.device_id = device_id
    self.context = usb1.USBContext() # Se establece contexto USB

    self.get_release_device(device_id, False)

    self.disconnect() # Se desconecta dispositivo

```

```

        self.pre_time = time.time() * 1000
        time.time_ns()

# Método principal que inicia la lectura de datos
def start(self):
    while True:
        if not self.connect():
            continue
        self.read_data()
        del self.handle
        self.disconnect()

# Método para leer datos del dispositivo y procesarlos
def read_data(self):
    # Dispositivo no presente, o usuario no tiene permisos para acceder al dispositivo.
    with self.handle.claimInterface(2):
        # Realizar acciones con los endpoints en la interfaz reclamada.
        self.handle.setInterfaceAltSetting(2, 0)
        self.handle.controlRead(0x01 << 5, request=0x22, value=0x01, index=2, length=2048, timeout=1000)
        # Construir una lista de objetos de transferencia y envíalos para preparar la bomba.
        transfer_list = []
        for _ in range(1):
            transfer = self.handle.getTransfer()
            transfer.setBulk(usb1.ENDPOINT_IN | 2, 2048, callback=self.processReceivedData, timeout=1000)
            transfer.submit()
            transfer_list.append(transfer)
        # Bucle mientras haya al menos una transferencia enviada.
        while any(x.isSubmitted() for x in transfer_list):
            # reading data
            self.context.handleEvents()

# Método para interpretar los datos recibidos del dispositivo
def parse_data(self, data: bytearray):
    if len(data) == 8 and int.from_bytes(bytes(data[4:]), 'big') == WEBUSB_JPEG_MAGIC:
        self.expect_size = int.from_bytes(bytes(data[4:]), 'big')
        self.buff = bytearray()
    elif len(data) == 8 and int.from_bytes(bytes(data[4:]), 'big') == WEBUSB_TEXT_MAGIC:
        self.expect_size = int.from_bytes(bytes(data[4:]), 'big')
        self.buff = bytearray()
    else:
        self.buff = self.buff + data

    if self.expect_size == len(self.buff):
        try:
            Image.open(BytesIO(self.buff))
        except:
            self.buff = bytearray()
            return
    if self.saveimg and ((time.time() * 1000 - self.pre_time) > self.interval):
        with open(f'./save_img/{time.time()}.jpg', 'wb') as f:
            f.write(bytes(self.buff))
        self.img_number += 1
        print(f'\rNumber of saved pictures on device {self.device_id}:{self.img_number}', end="")
        self.pre_time = time.time() * 1000

    if self.showimg:
        self.show_byte()
        self.buff = bytearray()

# Método para mostrar la imagen en una ventana de visualización utilizando OpenCV
def show_byte(self):
    try:
        img = Image.open(BytesIO(self.buff))
        img = np.array(img)
        cv2.imshow('img', cv2.cvtColor(img, cv2.COLOR_RGB2BGR))
        cv2.waitKey(1)
    except:

```

```

    return

# Método para procesar los datos recibidos en una transferencia USB
def processReceivedData(self, transfer):
    if transfer.getStatus() != usb1.TELEGRAM_COMPLETED:
        # transfer.close()
        return

    data = transfer.getBuffer()[:transfer.getActualLength()]
    # Process data...
    self.parse_data(data)
    # Resubmit transfer once data is processed.
    transfer.submit()

# Método para establecer la conexión con el dispositivo
def connect(self):
    """Get open devices"""
    self.handle = self.get_release_device(self.device_id, get=True)
    if self.handle is None:
        print("\rPlease plug in the device!")
        return False
    with self.handle.claimInterface(2):
        self.handle.setInterfaceAltSetting(2, 0)
        self.handle.controlRead(0x01 << 5, request=0x22, value=0x01, index=2, length=2048, timeout=1000)
        print('device is connected')
    return True

# Método para desconectar el dispositivo
def disconnect(self):
    try:
        print('Resetting device...')
        with usb1.USBContext() as context:
            handle = context.getByVendorIDAndProductID(VendorId, self.ProductId[self.device_id],
                                                        skip_on_error=False).open()
            handle.controlRead(0x01 << 5, request=0x22, value=0x00, index=2, length=2048, timeout=1000)
            handle.close()
            print('Device has been reset!')
    except:
        return False

# Método para obtener y liberar el dispositivo USB
def get_release_device(self, did, get=True):
    """Turn the device on or off"""
    tmp = 0
    print('*' * 50)
    print('looking for device!')
    for device in self.context.getDeviceIterator(skip_on_error=True):
        product_id = device.getProductID()
        vendor_id = device.getVendorID()
        device_addr = device.getDeviceAddress()
        bus = '->'.join(str(x) for x in [f'Bus {033[0m}', device.getBusNumber(), device.getPortNumberList()])
        if vendor_id == VendorId and product_id in ProductId and tmp == did:
            self.ProductId.append(product_id)
            print(f'\r{033[4;31mID {vendor_id:04x}:{product_id:04x} {bus} Device {device_addr}{033[0m}', end='')
            if get:
                return device.open()
            else:
                device.close()
                print(
                    f'\r{033[4;31mID {vendor_id:04x}:{product_id:04x} {bus} Device {device_addr}{033[0m}'
                    CLOSED{033[0m',
                    flush=True)
        elif vendor_id == VendorId and product_id in ProductId:
            self.ProductId.append(product_id)
            print(f'{033[0;31mID {vendor_id:04x}:{product_id:04x} {bus} Device {device_addr}{033[0m}')

```

```

        tmp = tmp + 1
    else:
        print(
            f'ID {vendor_id:04x}:{product_id:04x} {bus} Device {device_addr}')
    if __name__ == '__main__':
        opt = argparse.ArgumentParser()
        opt.add_argument('--unsave', action='store_true', help='whether save pictures')
        opt.add_argument('--unshow', action='store_true', help='whether show pictures')
        opt.add_argument('--device-num', type=int, default=1, help='Number of devices that need to be connected')
        opt.add_argument('--interval', type=int, default=300, help='ms,Minimum time interval for saving pictures')
        arg = opt.parse_args()
        if arg.device_num == 1:
            implement(arg, 0)
        elif arg.device_num <= 0:
            raise 'The number of devices must be at least one!'
        else:
            pro_ls = []
            for i in range(arg.device_num):
                pro_ls.append(Thread(target=implement, args=(arg, i)))
            for i in pro_ls:
                i.start()

```

- e. Ejecute el script Python para comenzar a capturar imágenes
[capture_images_script.py](#)

En el “símbolo del sistema”, al pegar el texto anterior se obtuvo el siguiente mensaje: “No se encontró Python; ejecuta sin argumentos para instalar desde Microsoft Store o deshabilita este acceso directo en Configuración > Administrar alias de ejecución de la aplicación”.

Este error era porque para ejecutar un programa en una ventana de terminal, es necesario estar situado en el directorio en el que se encuentra el programa. Por lo que se deberá de abrir en el Explorador de archivos de Windows la carpeta que contiene el programa, hacer clic a la derecha de la ruta para que se muestre y seleccionarla, borrar el texto y escriba en su lugar cmd y dar enter.

Después al volver a enviar el texto se obtuvo el siguiente mensaje

```
C:\Users\Ana\Documents\im>capture_images_script.py
Traceback (most recent call last):
  File "C:\Users\Ana\Documents\im\capture_images_script.py", line 7, in <module>
    import cv2
ModuleNotFoundError: No module named 'cv2'
```

Imagen 21. ModuleNotFoundError: No module named 'cv2'

El mensaje de error *ModuleNotFoundError: No module named 'cv2'* indica que el módulo cv2, que forma parte de la librería OpenCV, no está instalado en tu entorno Python. Por lo que se debe de instalar la

librería OpenCV abriendo el símbolo del sistema e introduciendo el siguiente comando

[pip3 install opencv-python](#)

```
C:\Users\Ana>pip3 install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl (38.6 MB)
    ----- 38.6/38.6 MB 5.1 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.21.2 in c:\python311\lib\site-packages (from opencv-python) (1.24.3)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.9.0.80

[notice] A new release of pip is available: 23.1.2 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Imagen 22. Instalar libreria python

Además ejecutar el siguiente comando:

[pip3 install Pillow](#)

De forma predeterminada, capturará una imagen cada 300 ms. Si desea cambiar esto, puede ejecutar el script en este formato.

[capture_images_script.py --interval <time_in_ms>](#)

Por ejemplo, para capturar una imagen cada segundo

`capture_images_script.py --interval 1000`

Después de ejecutar el script anterior, SenseCAP A1101 comenzará a capturar imágenes de las cámaras integradas continuamente y las guardará todas dentro de una carpeta llamada **save_img** .

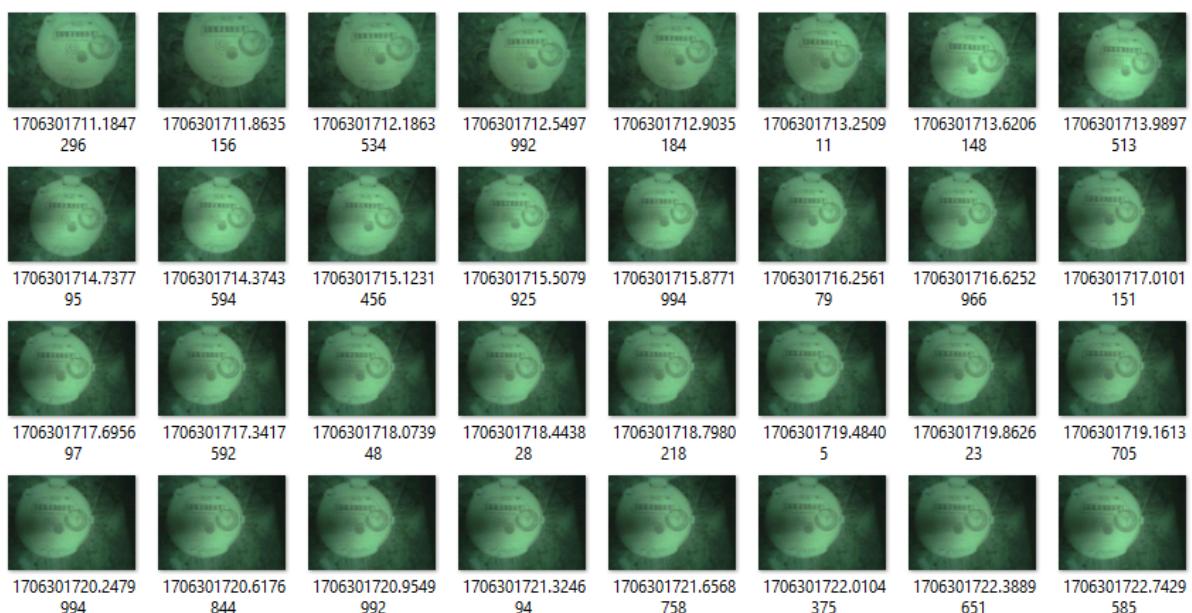


Imagen 23. Fotos tomadas con el dispositivo

Además, abrirá una ventana de previsualización mientras está grabando.

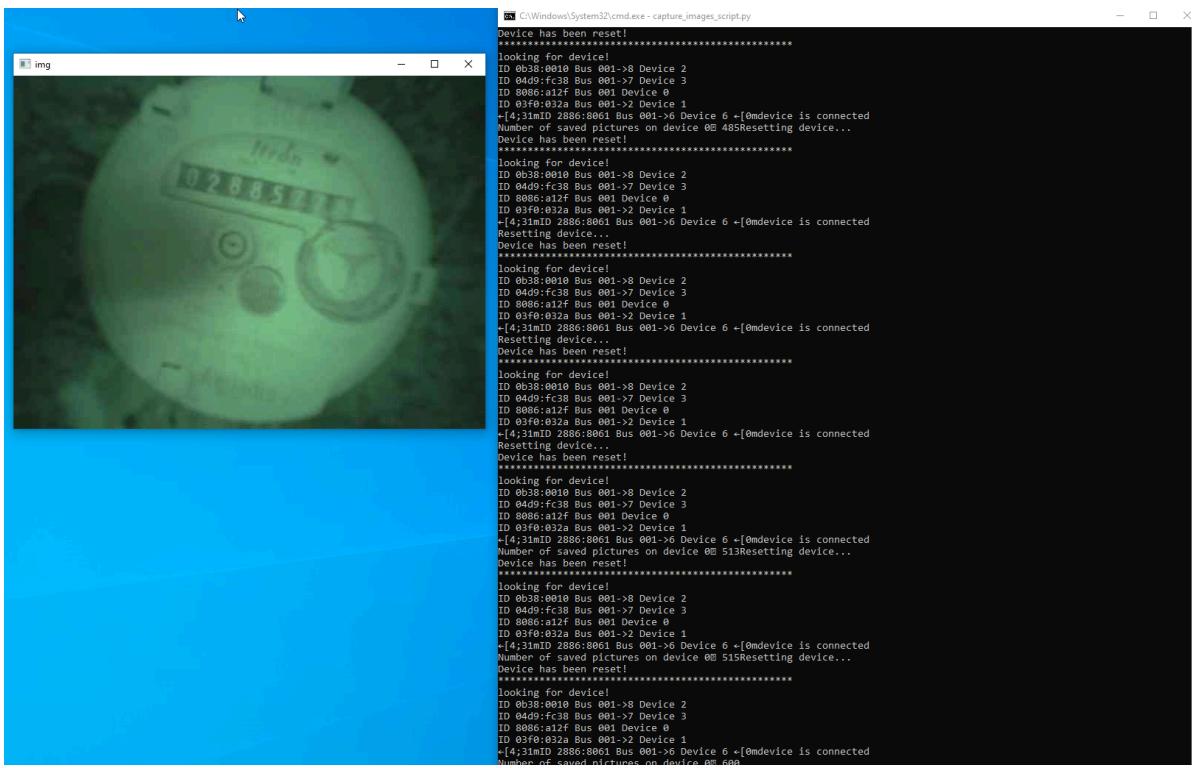


Imagen 24. Ventana de previsualización

Cuando haya capturado suficientes imágenes, haga clic en la ventana de la terminal y presiona las siguientes combinaciones de teclas para detener el proceso de captura.

Ctrl + Break

Cambiar el firmware del dispositivo después de la recopilación

Después de grabar las imágenes para el conjunto de datos, se debe de cambiar el firmware dentro del SenseCAP A1101 de nuevo al original, para que pueda cargar nuevamente modelos de detección de objetos para la detección.

- Acceder al modo de arranque en el SenseCAP A1101
 - Arrastrar y soltar el [archivo .uf2](#) a la unidad SENSECAP.
- En cuanto el uf2 termine de copiarse en la unidad, ésta desaparecerá. Esto significa que el uf2 se ha cargado correctamente en el módulo.

2. Generar el conjunto de datos con RoboFlow

- Crear una cuenta en la plataforma de Roboflow
- Asignar nombre del espacio de trabajo y elegir un plan. En el plan gratuito los conjuntos de datos y modelos son públicos, además se debe de tener en cuenta las funciones limitadas como que solo se puede tener un máximo de 3 proyectos y 3 créditos de entrenamiento.

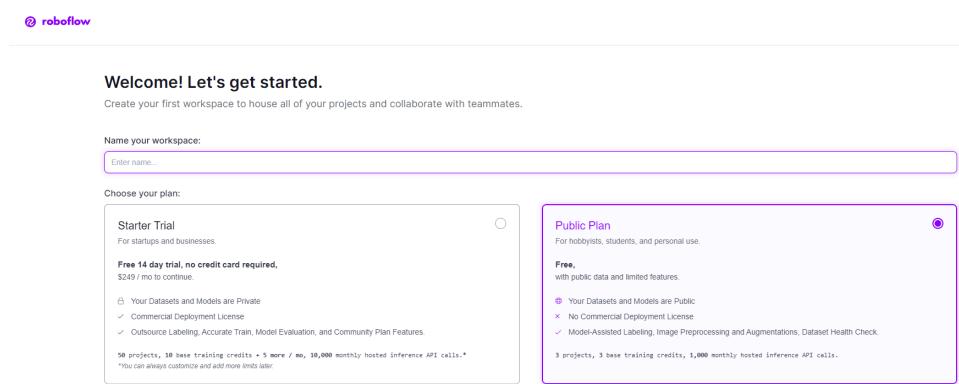


Imagen 25. Plan

c. Dar clic en “Crear nuevo proyecto”

Public
Prueba 0 member

Create Workspace

+ Create New Project

This workspace has no projects.
Create new projects and they'll appear here.

+ Create New Project

Tasks

There are no tasks here yet!
Tasks will appear here once you have images to annotate, submit, or approve.

Imagen 26. Espacio de trabajo

d. Asignar nombre del proyecto, mantener la licencia “CC by 4.0” y el tipo de proyecto “Object Detection (Bounding Box)” como predeterminados.

Let's create your project.

New Public Project

Project Name

Medidor de agua

License

CC BY 4.0

Annotation Group

dgitos

Project Type

Object Detection

Identify objects and their positions with

Classification

Assign labels to the entire image

Instance Segmentation

Detect multiple objects and their actual shape

Keypoint Detection

Identify keypoints ("skeletons") to subjects

Imagen 27. Crear nuevo proyecto

- e. Asignar clases 0,1,2,3,4,5,6,7,8,9

COLOR	CLASS NAME
●	0
●	1
●	2
●	3
●	4
●	5
●	6
●	7
●	8
●	9

Imagen 28. Asignar clases

- f. Arrastra y suelta las imágenes que has capturado usando SenseCAP A1101. En este caso se capturaron 1029 imágenes.

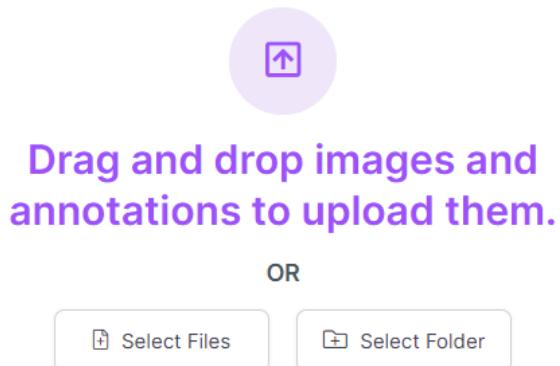


Imagen 29. Arrastrar imágenes

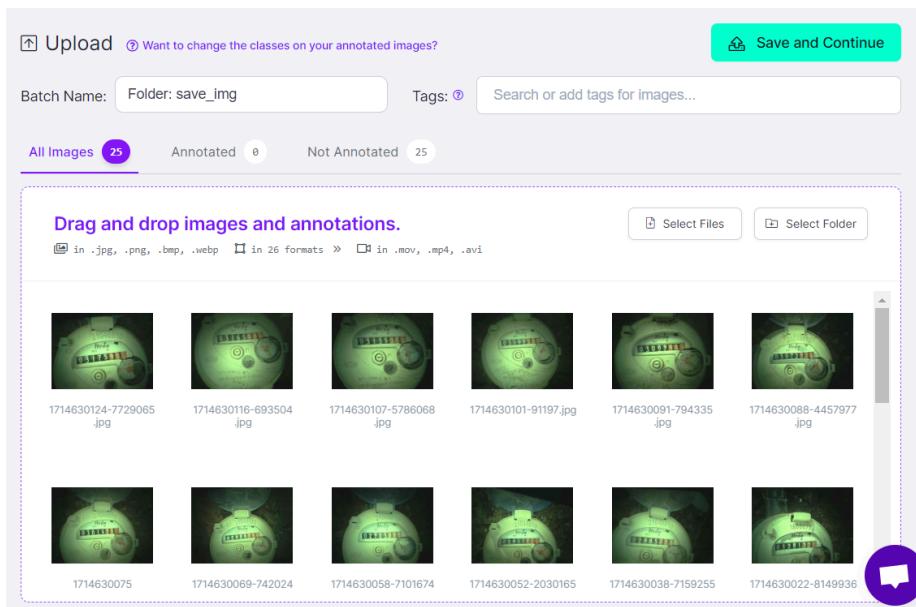


Imagen 30. Guardar imágenes

- g. Una vez procesadas las imágenes, haga clic en “Guardar y continuar”.
- h. Después de cargar las imágenes, hacer clic en “Asignar imágenes”
- i. Seleccione una imagen, dibuje una caja rectangular alrededor de cada dígito y elija la etiqueta que corresponde al número.



Imagen 31. Etiquetar dígitos.

- j. Repite lo mismo para el resto de imágenes
- k. Continuar anotando todas las imágenes del conjunto de datos.
- l. Una vez etiquetadas, haz clic en “Añadir imágenes al conjunto de datos”
- m. Después, dividiremos las imágenes entre "Train, Valid y Test". Si hay más conjuntos de datos, puede ser 80/20. Si los conjuntos de datos son menos, puede ser 85/15. Tener en cuenta que "Train" no debe ser inferior a 80.

Train (entrenamiento): La gran mayoría de tus imágenes irán aquí para entrenar tu modelo.

Valid (válido): Las imágenes de este conjunto se utilizarán durante el entrenamiento para comprobar el rendimiento del modelo y obtener información para mejorarlo.

Test (prueba): Una vez finalizado el entrenamiento, el conjunto de prueba se utiliza para comprobar el rendimiento del modelo en imágenes que no se han utilizado para el entrenamiento. Esto te da una mejor idea de cómo funcionará tu modelo en producción.

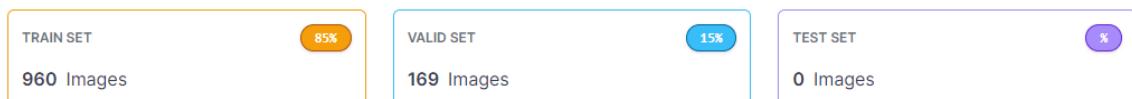


Imagen 32. Agregar imágenes al conjunto de datos

- n. Dar clic en “+ Nueva versión”
- o. Ahora puede añadir Preprocesamiento y Aumento si lo prefiere.
El preprocesamiento permite recortar, redimensionar, rotar y mucho más antes del entrenamiento.
El aumento realiza transformaciones en las imágenes existentes para crear nuevas variaciones y aumentar el número de imágenes de nuestro conjunto de datos. En última instancia, esto hace que los modelos sean más precisos en una gama más amplia de casos de uso.
Cambiar el tamaño de la imagen a 192x192 porque utilizaremos ese tamaño para el entrenamiento y éste será más rápido. De lo contrario, tendrá que convertir todas las imágenes a 192x192 durante el proceso de entrenamiento lo que consume más recursos de la CPU y hace que el proceso de entrenamiento sea más lento.
- p. A continuación, continuar con los valores predeterminados restantes y hacer clic en “Generar”
- q. Haga clic en Exportar , seleccione Formatear como YOLO V5 , seleccione Mostrar código de descarga y haga clic en Continuar.
Esto generará un fragmento de código que usaremos más adelante dentro de Google Colab.

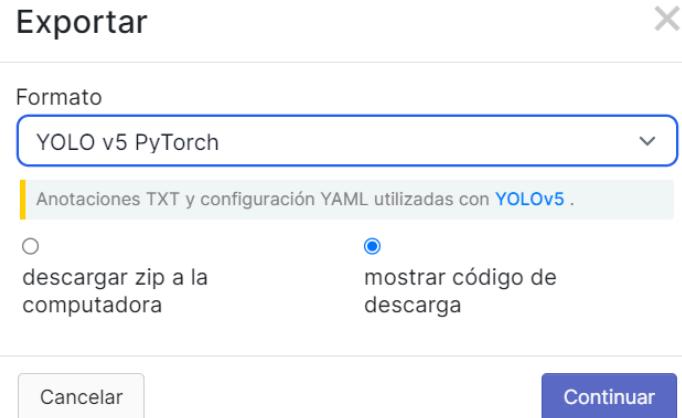


Imagen 33. Exportar código

Entrena usando EdgeLab en Google Colab

Dar clic en el enlace [YOLOV5-training-for-Vision-AI.ipynb](#) para abrir un espacio de trabajo de Google Colab preparado, seguir los pasos mencionados en el espacio de trabajo y ejecutar las celdas de código una a una.

Paso 1. Seleccionar GPU en tiempo de ejecución.
El GPU es la unidad de procesamiento gráfico

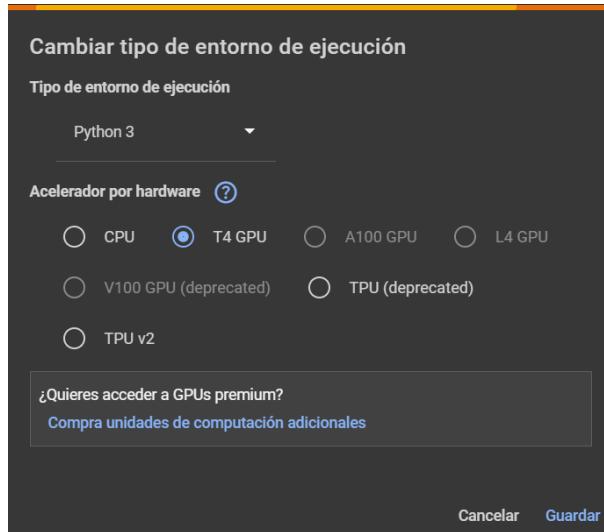


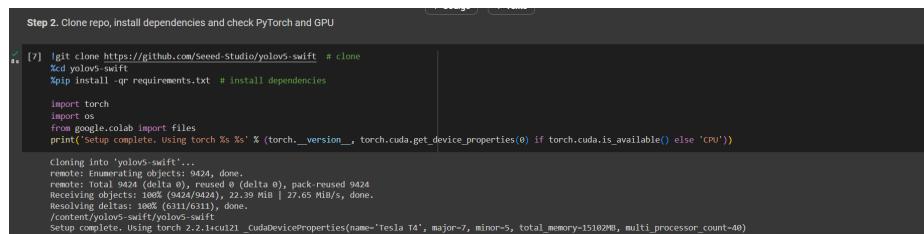
Imagen 34. Cambiar tipo de entorno de ejecución

Paso 2: Clonar el repositorio, instalar dependencias y verificar PyTorch y GPU:

Clonar el repositorio: Este paso clona el repositorio de GitHub yolov5-swift en tu entorno de Google Colab. El signo de exclamación ! indica que se trata de un comando de terminal.

Instalar dependencias: Utiliza pip para instalar las dependencias necesarias del proyecto. Estas dependencias se especifican en el archivo requirements.txt.

Verificar PyTorch y GPU: Importa las bibliotecas necesarias, como torch, os, y google.colab.files. Luego, imprime un mensaje que indica si la configuración está completa y si PyTorch está utilizando la GPU disponible. Esto es útil para asegurarse de que la instalación y configuración se hayan realizado correctamente.



```
Step 2. Clone repo, install dependencies and check PyTorch and GPU
[7] !git clone https://github.com/Seed-Studio/yolov5-swift # clone
  %cd yolov5-swift
  %pip install -qr requirements.txt # install dependencies

import torch
import os
from google.colab import files
print("Setup complete. Using torch %s" % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_available() else 'CPU'))

Cloning into 'yolov5-swift'...
remote: Enumerating objects: 9424, done.
remote: Total 9424 (delta 0), reused 0 (delta 0), pack-reused 9424
Receiving objects: 100% (9424/9424), 22.39 MiB | 27.65 MiB/s, done.
Resolving deltas: 100% (6311/6311), done.
/content/yolov5-swift/yolov5-swift
Setup complete. Using torch 2.2.1+cu11_CudaDeviceProperties(name='Tesla T4', major=7, minor=5, total_memory=15102M, multi_processor_count=40)
```

Imagen 35. Paso 2

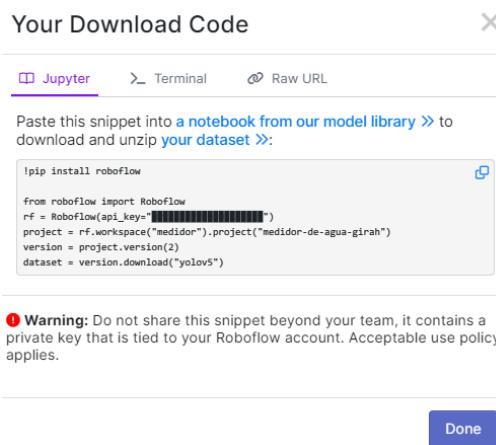
Paso 3. Configurar el entorno. Establece una variable de entorno llamada DATASET_DIRECTORY

Step 3. Set up environment

```
[8] os.environ["DATASET_DIRECTORY"] = "/content/datasets"
```

Imagen 36. Paso 3

Paso 4. Copiar y pegar el fragmento de código mostrado de Roboflow



Your Download Code

Jupyter Terminal Raw URL

Paste this snippet into [a notebook from our model library >>](#) to download and unzip [your dataset >>](#):

```
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="REDACTED")
project = rf.workspace("medidor").project("medidor-de-agua-girah")
version = project.version(2)
dataset = version.download("yolov5")
```

Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Done

Imagen 37. Paso 4

```

Collecting roboflow
  Downloading roboflow-1.1.36-py3-none-any.whl (76 kB)
    ━━━━━━━━━━━━━━━━ 76.7/76.7 kB 3.1 MB/s eta 0:00:00
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from roboflow) (2024.7.4)
Collecting chardet==4.0.0 (from roboflow)
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
    ━━━━━━━━━━━━━━━━ 178.7/178.7 kB 8.0 MB/s eta 0:00:00
Requirement already satisfied: idna==3.7 in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7)
Requirement already satisfied: cyclere in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.25.2)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.10.0.84)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (9.4.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Requirement already satisfied: urllib3>1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.0.7)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.66.4)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0.1)
Collecting requests-toolbelt (from roboflow)
  Downloading requests_toolbelt-1.0.0-py2.py3-none-any.whl (54 kB)
    ━━━━━━━━━━━━━━━━ 54.5/54.5 kB 7.9 MB/s eta 0:00:00
Collecting filetype (from roboflow)
  Downloading filetype-1.2.0-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: contourpy>1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.2.1)
Requirement already satisfied: fonttools>4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.53.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.3.2)
Installing collected packages: filetype, python-dotenv, chardet, requests-toolbelt, roboflow
  Attempting uninstall: chardet
    Found existing installation: chardet 5.2.0
    Uninstalling chardet-5.2.0:
      Successfully uninstalled chardet-5.2.0
Successfully installed chardet-4.0.0 filetype-1.2.0 python-dotenv-1.0.1 requests-toolbelt-1.0.0 roboflow-1.1.36
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in /content/datasets/Medidor-de-agua-2 to yolov5pytorch:: 100%|██████████| 5659/5659 [00:00<00:00, 17593.88it/s]
Extracting Dataset Version Zip to /content/datasets/Medidor-de-agua-2 in yolov5pytorch:: 100%|██████████| 2267/2267 [00:00<00:00, 9679.17it/s]

# this is the YAML file Roboflow wrote for us that we're loading into this notebook with our data
*cat {dataset.location}/data.yaml

```

Imagen 38. Resultados de paso 4

Al ejecutar la celda anterior da el siguiente resultado

```

names:
- '0'
- '1'
- '2'
- '3'
- '4'
- '5'
- '6'
- '7'
- '8'
- '9'
nc: 10
roboflow:
  license: CC BY 4.0
  project: medidor-de-agua-girah
  url: https://universe.roboflow.com/medidor/medidor-de-agua-girah/dataset/2
  version: 2
  workspace: medidor
test: ../test/images
train: /content/datasets/Medidor-de-agua-2/train/images
val: /content/datasets/Medidor-de-agua-2/valid/images

```

Imagen 39. Resultados de paso 4

Paso 5. Descargar un modelo preentrenado adecuado para nuestro entrenamiento.

!wget

<https://github.com/Seeed-Studio/yolov5-swift/releases/download/v0.1.0-alpha/yolov5n6-xiao.pt>

Paso 6. Iniciar el entrenamiento

img: define el tamaño de la imagen de entrada

batch: determina el tamaño del lote

epochs: define el número de epochs de entrenamiento

data: define la ruta a nuestro archivo yaml

cfg: especifica la configuración de nuestro modelo

weights: especifica una ruta personalizada para los pesos

name: nombres de los resultados

nosave: sólo guarda el punto de control final

cache: caché de imágenes para un entrenamiento más rápido

```
!python3 train.py --img 192 --batch 64 --epochs 350 --data
{dataset.location}/data.yaml --cfg yolov5n6-xiao.yaml --weights
yolov5n6-xiao.pt --name yolov5n6_results --cache
```

En este paso al ejecutar el código aparece un error por lo que se deberán de abrir los archivos “datasets.py” y “general.py”. En estos archivos se deberá de reemplazar “np.int” con “int” en los renglones 443 y 485 del primer archivo, y en los renglones 577 y 592 del segundo archivo.

Epoch	gpu_mem	box	obj	cls	labels	img_size	
344/349	0.508G	0.0416	0.01692	0.02133	603	192: 100% 15/15 [00:06<00:00,	2.46it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:01<00:00,
all	169	1141		0.933	0.799	0.908	1.22it/s]
Epoch	gpu_mem	box	obj	cls	labels	img_size	
345/349	0.508G	0.042	0.01741	0.0216	718	192: 100% 15/15 [00:04<00:00,	3.39it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:00<00:00,
all	169	1141		0.931	0.799	0.911	2.22it/s]
Epoch	gpu_mem	box	obj	cls	labels	img_size	
346/349	0.508G	0.04204	0.01761	0.0213	586	192: 100% 15/15 [00:03<00:00,	3.91it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:00<00:00,
all	169	1141		0.934	0.797	0.912	2.34it/s]
Epoch	gpu_mem	box	obj	cls	labels	img_size	
347/349	0.508G	0.04151	0.01701	0.02106	621	192: 100% 15/15 [00:05<00:00,	2.87it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:01<00:00,
all	169	1141		0.851	0.838	0.911	1.44it/s]
Epoch	gpu_mem	box	obj	cls	labels	img_size	
348/349	0.508G	0.0416	0.01741	0.02101	643	192: 100% 15/15 [00:05<00:00,	2.79it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:00<00:00,
all	169	1141		0.935	0.795	0.909	2.36it/s]
Epoch	gpu_mem	box	obj	cls	labels	img_size	
349/349	0.508G	0.04161	0.01723	0.02105	618	192: 100% 15/15 [00:03<00:00,	4.12it/s]
Class	Images	Labels		P	R	mAP@.5 mAP@.5:.95:	100% 2/2 [00:00<00:00,
all	169	1141		0.86	0.839	0.908	2.25it/s]

350 epochs completed in 0.613 hours.
Optimizer stripped from runs/train/yolov5n6_results/weights/last.pt, 1.1MB
Optimizer stripped from runs/train/yolov5n6_results/weights/best.pt, 1.1MB
Validating runs/train/yolov5n6_results/weights/best.pt...
Fusing layers...
Model Summary: 281 layers, 447660 parameters, 0 gradients, 0.8 GFLOPs
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 2/2 [00:03<00:00, 1.72s/it]
all 169 1141 0.91 0.764 0.901 0.638
0 169 280 0.926 1 0.995 0.701
1 169 118 0.985 1 0.995 0.682
2 169 149 0.993 0.988 0.995 0.732
3 169 192 0.96 1 0.995 0.736
4 169 22 1 0 0.772 0.571
5 169 31 0.631 0.386 0.795 0.528
6 169 40 0.848 0.55 0.602 0.389
7 169 68 0.836 0.853 0.904 0.65
8 169 119 0.957 0.992 0.994 0.679
9 169 122 0.964 0.873 0.963 0.712

Results saved to runs/train/yolov5n6_results

Imagen 40. Resultados de paso 6

Paso 7. Exportar archivo TensorFlow Lite

```
!python3 export.py --data {dataset.location}/data.yaml --weights runs/train/yolov5n6_results4/weights/best.pt --imgsz 192 --int8 --include tflite
```

```
Export complete (38.38s)
Results saved to /content/yolov5-swift/runs/train/yolov5n6_results/weights
Detect: python detect.py --weights runs/train/yolov5n6_results/weights/best-int8.tflite
PyTorch Hub: model = torch.hub.load('ultralytics/yolov5', 'custom', 'runs/train/yolov5n6_results/weights/best-int8.tflite')
Validate: python val.py --weights runs/train/yolov5n6_results/weights/best-int8.tflite
Visualize: https://netron.app
```

Imagen 41. Resultados de paso 7

Paso 8. Convertir TensorFlow Lite a archivo UF2

UF2 es un formato de archivo, desarrollado por Microsoft. Seeed utiliza este formato para convertir .tflite a .uf2, permitiendo que los archivos tflite sean almacenados en los dispositivos IoT lanzados por Seeed. Actualmente los dispositivos Seeed soportan hasta 4 modelos, cada modelo (.tflite) es inferior a 1M .

```
[9] # Place the model to index 1
!python3 uf2conv.py -f GROVEAI -t 1 -c runs//train/yolov5n6_results//weights/best-int8.tflite -o model-1.uf2
%cp model-1.uf2 ../

→ Converted to uf2, output size: 1286144, start address: 0x30000000
Wrote 1286144 bytes to model-1.uf2
```

Imagen 42. Resultados de paso 8

Paso 9. Descargar el archivo del modelo entrenado

```
✓ [17] files.download("/content/model-1.uf2") ?
```

model-1.uf2
1.234 KB • Hecho

Imagen 43. Paso 9

De los archivos creados en la ruta /content/yolov5-swift/runs/train/yolov5n6_results, se pueden observar los resultados del entrenamiento. En una de las pruebas realizadas se obtuvieron los siguientes resultados.

Resultados del entrenamiento

F1 CURVE

La gráfica F1_curve muestra en el eje X la "confianza" (confidence) y en el eje Y el "F1 Score". Confianza se refiere a la probabilidad asignada por el modelo a una predicción, y el F1 Score mide el equilibrio entre la precisión y la exhaustividad. De acuerdo a la gráfica se puede observar que para todas las clases se obtuvo un F1 de 0.90 con un umbral de confianza de 0.336.

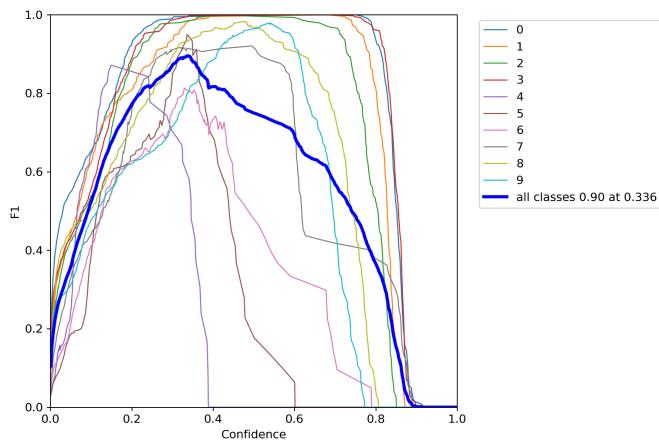


Imagen 44. Gráfica F1_Curve

PR

Se obtiene una gráfica llamada PR (Precision-Recall). En esta gráfica el eje X representa el "Recall" (Recuperación) y el eje Y representa la "Precisión". Para todas las clases se obtuvo un 0.956 mAP@0.5. Esto significa que el promedio de la precisión media (mAP) para todas las clases es de 0.956 cuando se utiliza un umbral de confianza (threshold) de 0.5 para la detección de objetos.

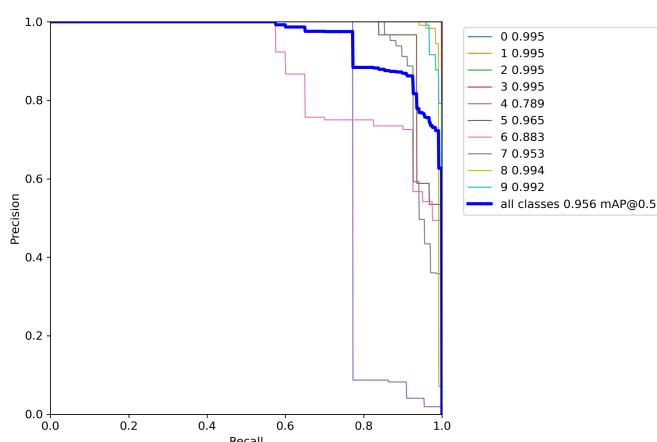


Imagen 45. Gráfica PR

P

En esta gráfica, el eje X representa la "confidence" (confianza) y el eje Y representa la "Precision" (precisión). De acuerdo a la gráfica se obtiene el dato de que todas las clases tienen una precisión de 1.00 cuando la confianza es de 0.752.

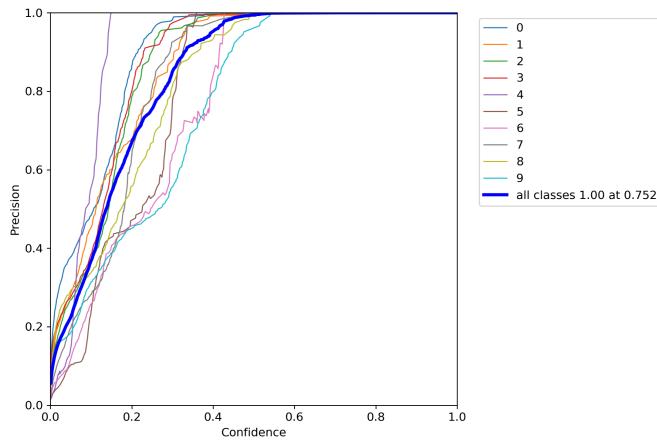


Imagen 46. Gráfica P

R

Esta gráfica representa la relación entre la confianza (confidence) y el recall. En este contexto, el eje X representa la confianza, que indica la certeza o probabilidad de que los dígitos detectados sean los correctos. El eje Y representa el recall, que es la proporción de los verdaderos positivos identificados respecto a todos los objetos de interés presentes en los datos.

Se puede observar que todas las clases tienen un recall de 1.00 a una confianza de 0.000

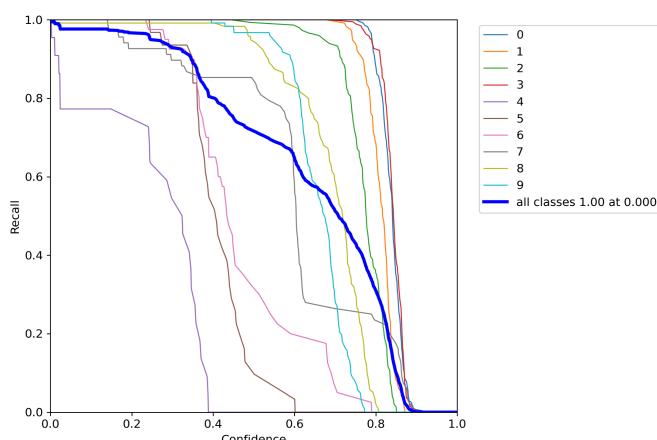


Imagen 47. Gráfica R

CONFUSION MATRIX

La matriz de confusión proporciona una visión detallada de cómo un modelo de clasificación está realizando sus predicciones en términos de verdaderos y falsos positivos y negativos, permitiendo así una evaluación exhaustiva de su rendimiento y posibles áreas de mejora.

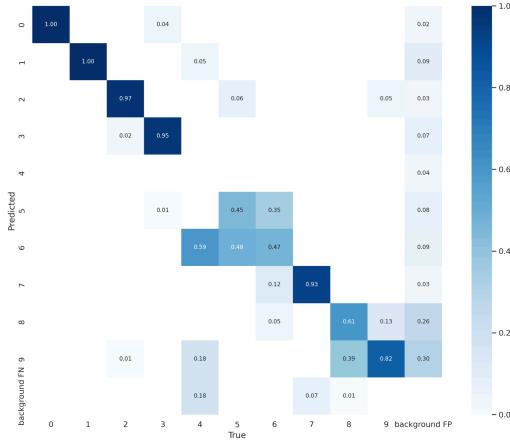


Imagen 48. Confusion Matrix

LABELS

En la siguiente imagen se pueden observar 4 gráficos donde la primera es un gráfico de barras donde en el eje X se muestran las clases, y en el eje Y se representa el número de instancias o apariciones de cada clase.

En el gráfico superior derecho está el gráfico de los cuadros delimitadores.

El gráfico inferior izquierdo es un gráfico de dispersión que representa las coordenadas centrales (x, y) de los cuadros delimitadores detectados.

El gráfico inferior derecho es un gráfico de dispersión que muestra la relación entre el ancho y la altura de los cuadros delimitadores detectados.

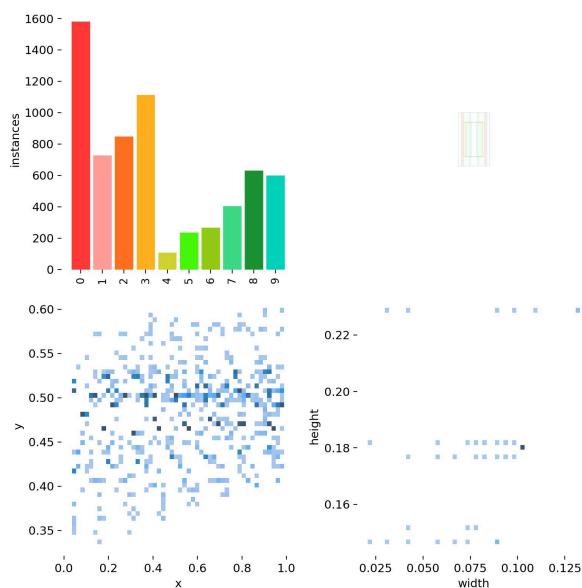


Imagen 49. Labels

LABELS CORRELOGRAM

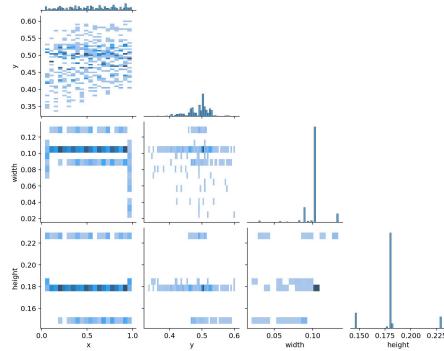


Imagen 50. Labels Correlogram

RESULTS

- train/box_loss: La gráfica muestra la pérdida de precisión en la ubicación de las cajas delimitadoras durante el entrenamiento. Una pérdida menor indica una mejora en la localización de objetos.
- val/box_loss: La gráfica muestra la pérdida de precisión en la ubicación de las cajas delimitadoras durante la validación. Una pérdida menor indica que el modelo mantiene su precisión en datos nuevos.
- train/obj_loss: La gráfica representa la pérdida de confianza en la presencia de objetos durante el entrenamiento. Una pérdida menor sugiere que el modelo se vuelve más seguro al detectar objetos correctamente.
- val/obj_loss: La gráfica representa la pérdida de confianza en la presencia de objetos durante la validación. Una disminución sugiere que el modelo generaliza bien su capacidad de detección en datos nuevos.
- train/cls_loss: La gráfica refleja la pérdida de precisión en la clasificación de los objetos durante el entrenamiento. Una disminución indica una mejora en la clasificación correcta de los objetos.
- val/cls_loss: La gráfica muestra la pérdida de precisión en la clasificación de los objetos durante la validación. Una pérdida menor indica que el modelo clasifica correctamente los objetos en datos nuevos.
- metrics/precision: La gráfica refleja la precisión del modelo en la detección de objetos, mostrando la proporción de detecciones correctas. Una alta precisión indica pocos falsos positivos.
- metrics/mAP_0.5: La gráfica presenta la media de la precisión promediada con un umbral de 0.5. Un valor alto sugiere una alta precisión en la detección con un nivel moderado de solapamiento.
- metrics/recall: La gráfica muestra el recall del modelo, evaluando su capacidad para identificar todos los objetos presentes. Un alto valor de recall indica pocos falsos negativos.

- metrics/mAP_0.5:0.95: La gráfica muestra la media de la precisión promediada con umbrales de 0.5 a 0.95. Un valor alto indica que el modelo mantiene una alta precisión en la detección incluso con mayor solapamiento.

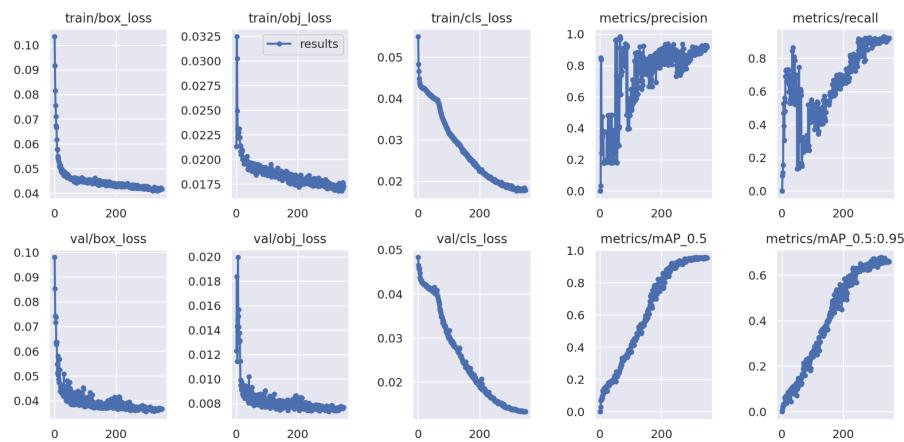


Imagen 51. Results

3. Implementar el modelo entrenado y realizar la inferencia

- Instala la última versión del navegador Google Chrome o Microsoft Edge y ábrelo
- Conecta SenseCAP A1101 a tu PC a través de un cable USB Tipo-C.
- Haga doble clic en el botón de arranque del SenseCAP A1101 para acceder al modo de almacenamiento masivo.
Después de esto, verá una nueva unidad de almacenamiento mostrada en su explorador de archivos como SENSECAP
- Arrastre y suelte el archivo model-1.uf2 a la unidad SENSECAP
En cuanto el uf2 termine de copiarse en la unidad, ésta desaparecerá.
Esto significa que el uf2 se ha cargado correctamente en el módulo.
Nota: Si tiene 4 archivos de modelo listos, puede arrastrar y soltar cada modelo uno por uno. Suelte el primer modelo, espere hasta que termine de copiarse, ingrese nuevamente al modo de inicio, suelte el segundo modelo y así sucesivamente. Si solo ha cargado un modelo (con índice 1) en SenseCAP A1101, cargará ese modelo.
- Dar clic en el siguiente enlace para abrir una ventana de vista previa de la transmisión de la cámara.

<https://vision-ai-demo.seeed.cn/>

seeed studio



Imagen 52. Ventana de vista previa de la transmisión de la cámara .

- Haga clic en el botón “Conectar” . Luego verá una ventana emergente en el navegador. Seleccione “SenseCAP Vision AI - Paired” y haga clic en “Conectar”



Imagen 53. Conectar .

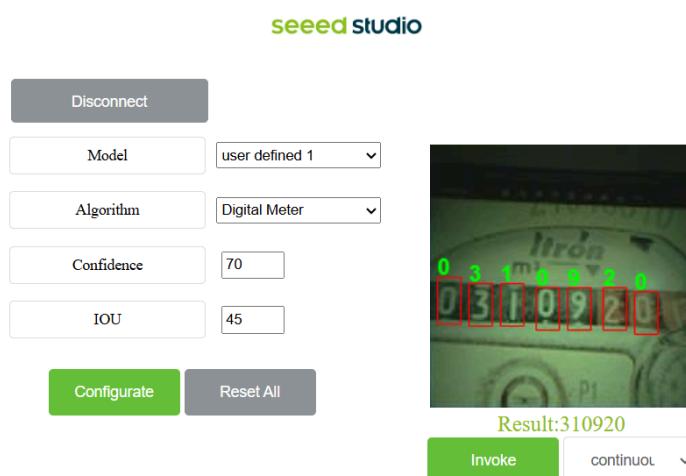
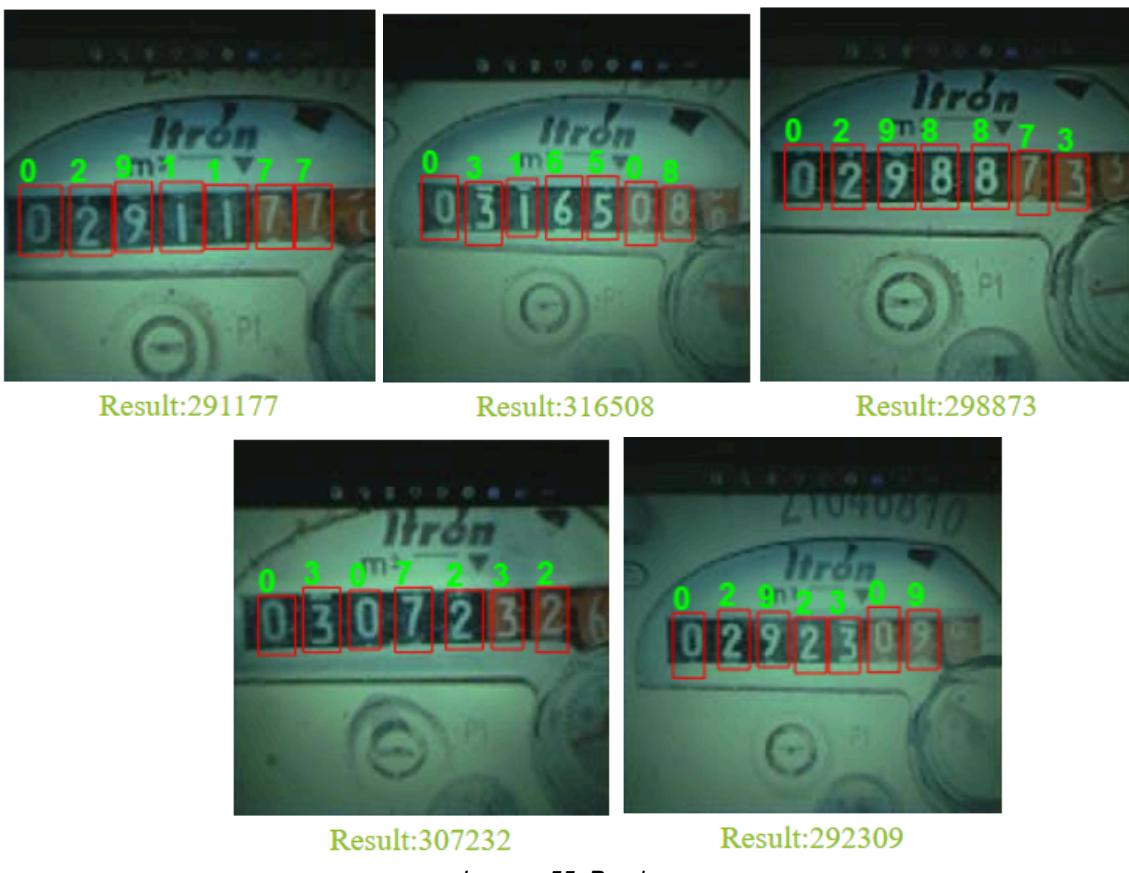


Imagen 54. Transmisión de cámara

- Confidence: La confianza se refiere al nivel de certeza o probabilidad que asigna un modelo a sus predicciones.
- IOU: El IoU se utiliza para evaluar la precisión de los recuadros delimitadores predichos en comparación con los recuadros delimitadores reales.

En las siguientes imágenes se pueden observar varias pruebas que se realizaron con diferentes números.



4. Realizar inferencia con SenseCAP A1101 en SenseCAP Mate

- Borrar el firmware del A1101 haciendo uso del archivo [erase_model.uf2](#). Luego actualice el firmware A1101 a la última versión y coloque el modelo de reconocimiento de dígitos del medidor de agua en A1101. [sensecap_ai_v02-00.uf2](#)
- Abrir ventana de vista previa de la transmisión de la cámara.
- Haga clic en el botón “Conectar” . Luego verá una ventana emergente en el navegador. Seleccione “SenseCAP A1101 - Paired” y haga clic en “Conectar”
- Abra SenseCAP Mate y emparejelo con su propio A1101, seleccione el mismo modelo y algoritmo que el anterior. Luego haga clic en “General” y haga clic en “Detectar“ en la parte inferior.

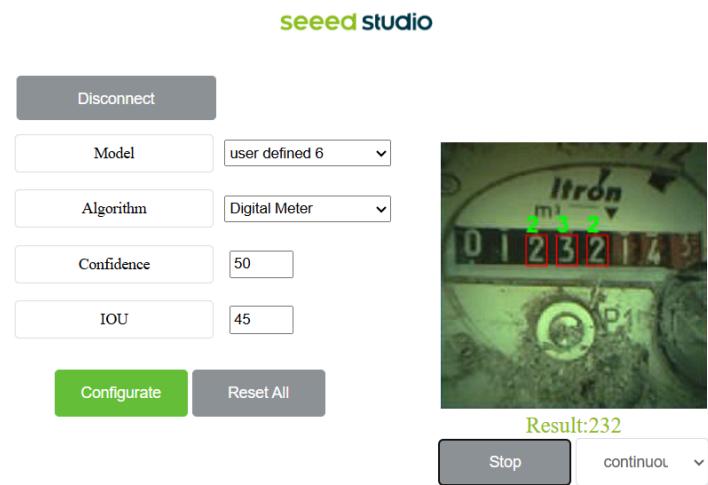


Imagen 56. Transmisión de cámara

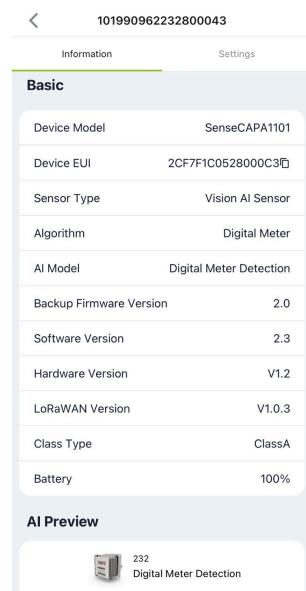


Imagen 57. SenseCAP Mate

Coneectar Gateway

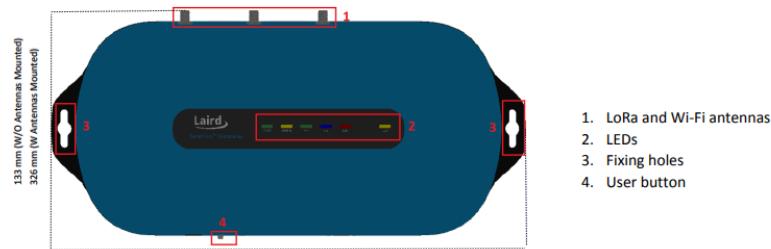


Figure 1: Top of the Sentrius™ RG1xx gateway

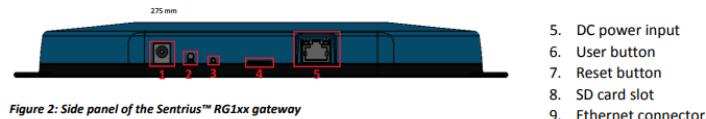


Figure 2: Side panel of the Sentrius™ RG1xx gateway

Imagen 58. Parte superior y lateral de Gateway .

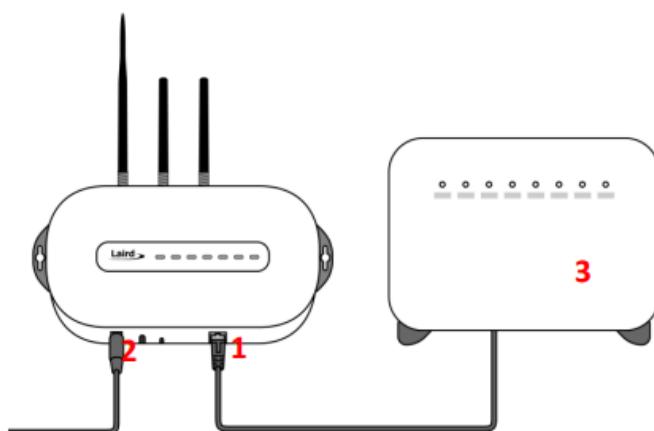


Imagen 59. Conexión de gateway

1. Coneectar las 3 antenas

- Colocar las dos antenas más cortas en los puertos de 2.4/5.5 GHz (Wi-Fi).
- Colocar la tercera antenna, más larga, en el puerto de 868 MHz/900 MHz (LoRa).

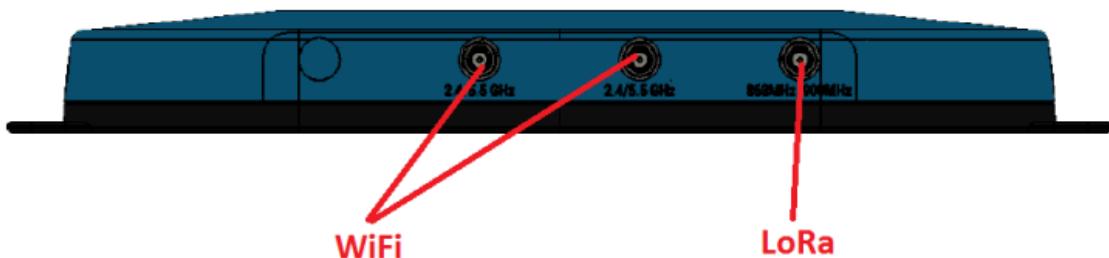


Imagen 60. Colocar antenas

2. Conecte el suministro de energía
3. Conectar Gateway al enrutador usando el cable Ethernet.

Configuración rápida Wi-Fi

El gateway incluye un modo que le permite configurarla sin acceso Ethernet, en caso de que se desee conectarse a una red inalámbrica.

1. Presionar y mantener presionado el botón de usuario durante siete segundos.
2. Desde un dispositivo habilitado para conexión inalámbrica, realice un escaneo de redes.
3. Conectarse al punto de acceso rg1xx2a2e58, donde "2A2E58" son los últimos seis dígitos de la dirección MAC Ethernet que se encuentra en la etiqueta en la parte inferior del gateway.

La red está protegida con WPA2 y la contraseña es la misma que el SSID.

La contraseña se puede cambiar en la página web Wi-Fi > Avanzado.

Al cerrar sesión o desconectar el cliente, la Configuración Rápida Wi-Fi se desactiva y se reanuda el funcionamiento normal.

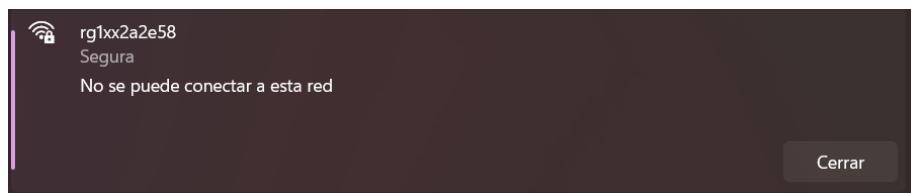


Imagen 61. Wi-Fi

Iniciar sesión en la interfaz web

1. Determine los últimos tres bytes de la dirección MAC Ethernet del gateway. Esto se puede encontrar en la etiqueta en la parte inferior del gateway.



Imagen 62. Etiqueta de gateway

2. Ingrese la URL en el navegador web para acceder a la interfaz web. El URL es <https://rg1xx2a2e58.local>, donde "2a2e58" son los últimos seis dígitos de la dirección MAC Ethernet.
3. Acepte el certificado de seguridad en el navegador.
4. Dar clic en Avanzado.
5. Dar clic en continuar



Imagen 63. Acceder a la interfaz web

6. Iniciar sesión con los siguientes datos:

username: citedi-sentrius

password: Idetic-2020

password wifi: Idetic-2020

Imagen 64. Iniciar sesión

Imagen 65. Dashboard de interfaz

Conclusión

La forma de entrenar el modelo resultó ser sencilla. Sin embargo, al implementar el modelo entrenado y realizar la inferencia, se presentaron dificultades para que el modelo detectara o leyera todos los números correctamente. Una posible causa de este problema podría ser la poca claridad de algunos números, lo que afectó la precisión del modelo en la identificación y lectura de los mismos.

Bibliografía

- Train Water Meter Digits Recognition Model with SenseCAP A1101. (s. f.). seeed studio. Recuperado 22 de enero de 2024, de <https://wiki.seeedstudio.com/train-water-meter-digits-recognition-model-with-sensecap-a1101>
- What is TinyML | Seeed Studio Wiki. (2023, 31 enero). Recuperado 24 de enero de 2024, de <https://wiki.seeedstudio.com/Wio-Terminal-TinyML/>
- Ejecutar programas. Python. Bartolomé Sintes Marco. www.mclibre.org. (s. f.). Recuperado 26 de enero de 2024, de <https://www.mclibre.org/consultar/python/otros/python-uso.html>
- Manual sensor A1101
- Build Vision Models with Roboflow | Roboflow Docs. (s. f.). Recuperado 26 de enero de 2024, de <https://docs.roboflow.com/>
- TensorFlow. (s. f.). TensorFlow. Recuperado 26 de enero de 2024, de <https://www.tensorflow.org/?hl=es-419>
- SenseCraft AI. (s. f.). Recuperado 26 de enero de 2024, de <https://seeed-studio.github.io/SenseCraft-Web-Toolkit/#/setup/process>
- Solawetz, J. (2024, 9 abril). How to Train a YOLOv5 Model On a Custom Dataset. Roboflow Blog. Recuperado 17 de abril de 2024, de <https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/#installing-the-yolov5-environment>
- Dwyer, B. (2024, 15 abril). Getting Started with Roboflow. Roboflow Blog. Recuperado 17 de abril de 2024, de <https://blog.roboflow.com/getting-started-with-roboflow/>
- Glossary. (s. f.). The Things Stack For LoRaWAN. Recuperado 17 de abril de 2024, de <https://www.thethingsindustries.com/docs/reference/glossary/>
- Seeed-Solution. (s. f.). TTN-Payload-Decoder/decoder_new-v3.js at master · Seeed-Solution/TTN-Payload-Decoder. GitHub. Recuperado 17 de abril de 2024, de https://github.com/Seeed-Solution/TTN-Payload-Decoder/blob/master/decoder_new-v3.js
- Ezurio RG191+LTE LoRaWAN-Enabled Gateway V2.0. (s. f.). Mouser Electronics. Recuperado 2 de mayo de 2024, de <https://www.mouser.mx/new/ezurio/laird-rg191-lte-gateway-v2/#Video-6>
- User Guide Sentrius RG1xx Version 2.2
- Wio-E5 Wireless Module (Bulk) - STM32WLE5JC, ARM Cortex-M4 and SX126x embedded, supports LoRaWAN on EU868 & US915. (2024, 4 julio). <https://www.seeedstudio.com/LoRa-E5-Wireless-Module-p-4745.html>
- WIO-E5 STM32WLE5JC Module | Seeed Studio Wiki. (2023, 30 enero). https://wiki.seeedstudio.com/LoRa-E5_STM32WLE5JC_Module/
- YOLOV5-training-for-Vision-AI.ipynb. (s. f.). Gist. <https://gist.github.com/lakshanthad/b47a1d1a9b4fac43449948524de7d374>