Construcción de un prototipo de dispensador de alimento para perros y gatos utilizando reconocimiento de imágenes mediante ESP32-CAM. (7 junio 2023)

Jhonattan Alexander Cordero Pareja, *Doctorado en Ingeniería*

El objetivo de este proyecto de clase es aplicar de manera práctica los conocimientos básicos adquiridos en el curso de inteligencia artificial. Para ello, se emplea la plataforma Edge Impulse y se realiza el entrenamiento utilizando Keras en Google Colab. El enfoque principal del proyecto es desarrollar un prototipo de dispensador de alimento para mascotas que incorpore una cámara ESP32. Esta cámara se utilizará para realizar el reconocimiento de la mascota mediante la implementación de un modelo MobileNet V2, el cual permitirá detectar y distinguir a la mascota en cuestión.

El artículo que se va a redactar sobre este proyecto se dividirá en dos partes fundamentales. En la primera parte, se describirá en detalle el proceso de entrenamiento de dos modelos utilizando Google Colab. Para ello, se hará uso de las bibliotecas de TensorFlow y se seguirán las mejores prácticas para la detección de imágenes. Este primer paso es esencial para poder desarrollar un modelo preciso y confiable que sea capaz de identificar correctamente a la mascota.

En la segunda parte del artículo, se abordará el despliegue del modelo entrenado en la plataforma Edge Impulse. Se explicará en profundidad cómo se lleva a cabo este proceso, desde la exportación del modelo entrenado en TensorFlow hasta su implementación en la cámara ESP32 y en el prototipo del dispensador de alimento. Esta etapa implica adaptar el modelo a las características y restricciones de la plataforma Edge Impulse, asegurando su correcto funcionamiento en el dispositivo final.

A lo largo del artículo se incluirán detalles técnicos relevantes, como la configuración específica utilizada, los parámetros de entrenamiento empleados, así como las ventajas y desafíos encontrados durante el desarrollo del proyecto. Además, se proporcionarán ejemplos visuales que ilustren el proceso de reconocimiento de la mascota y se presentarán los resultados obtenidos, evaluando el desempeño y la eficacia del modelo implementado.

Palabras clave—

- **plataforma Edge impulse.**
- **Entrenamiento.**
- **Keras, Google Colab,**
- **Esp32cam.**
- **MobileNet V2**
- **Prototipo.**

I. GENERACIÓN DE UN DATASET PROPIO (UNO BASADO EN CONVOLUCIÓN EN MOBILENET Y INCEPTIONV3.

El primer paso crucial en este proyecto fue la recolección de datos necesarios para el entrenamiento del modelo. Se recopilaron alrededor de 60 imágenes de los cuatro casos que se abordan en el proyecto:

1. Fondo: Imágenes que representan el fondo o entorno en el que se ubicará el dispensador de alimento para mascotas.
2. Terry: Imágenes de la mascota llamada Terry, que se desea detectar y reconocer.
3. Chloe: Imágenes de la mascota llamada Chloe, también objetivo de detección y reconocimiento.
4. Terry y Chloe: Imágenes en las que aparecen ambas mascotas juntas, para entrenar al modelo en el reconocimiento de esta combinación específica.

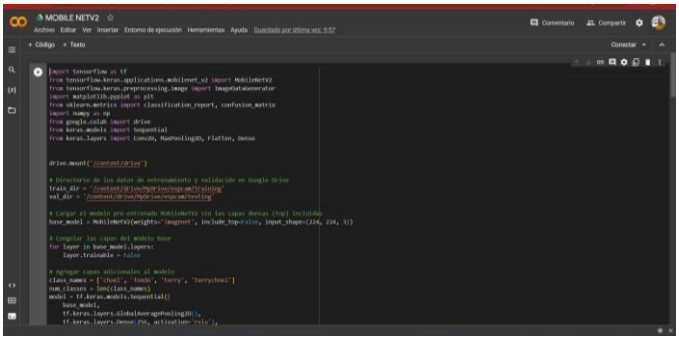
Cada caso hace referencia a una mascota particular que se busca detectar y distinguir. La recopilación de estas imágenes es fundamental para tener un conjunto de datos representativo y realizar un entrenamiento adecuado del modelo.

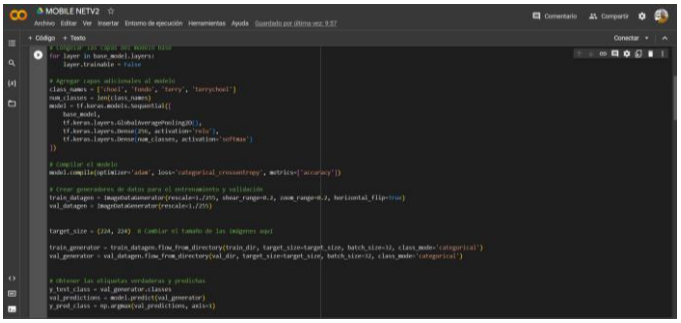
Una vez se dispuso de toda la información necesaria, se procedió a utilizar el entorno de programación Google Colab. En este entorno, se importaron las imágenes recolectadas y se utilizó un modelo preentrenado de MobileNet. El modelo preentrenado se empleó para entrenar el modelo en cuestión.

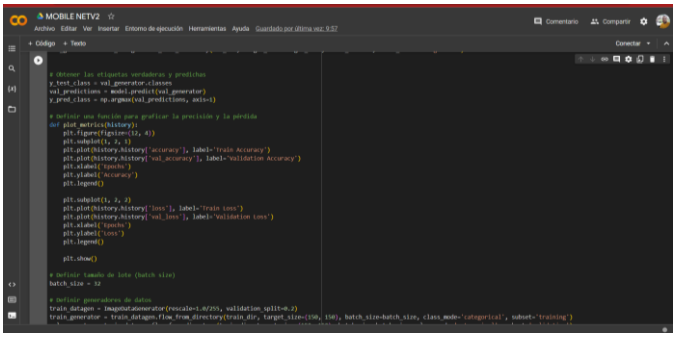
El proceso de entrenamiento consistió en alimentar el modelo con las imágenes de entrenamiento previamente recolectadas, permitiéndole aprender y ajustar sus parámetros para

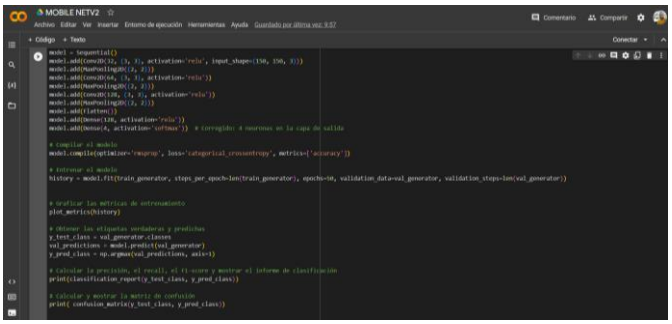
reconocer correctamente a las mascotas. Esta etapa de entrenamiento es crucial para obtener un modelo preciso y confiable que pueda identificar con éxito a las mascotas en las imágenes futuras.

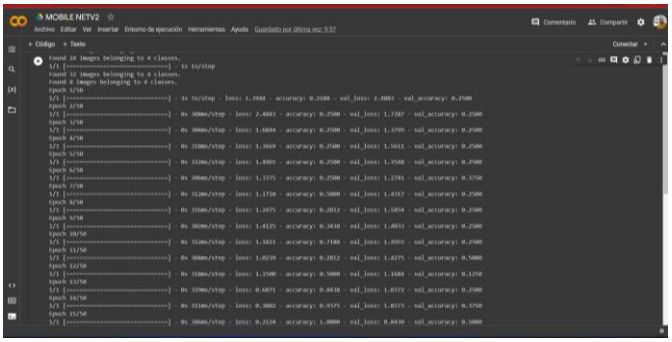
El uso del modelo preentrenado de MobileNet resulta beneficioso, ya que este modelo ha sido entrenado en un conjunto de datos masivo y diverso, lo cual le proporciona una capacidad de reconocimiento amplia y generalizada. Aprovechar este modelo preentrenado ahorra tiempo y esfuerzo en el proceso de entrenamiento, permitiendo enfocarse en los detalles específicos de las mascotas en este caso.

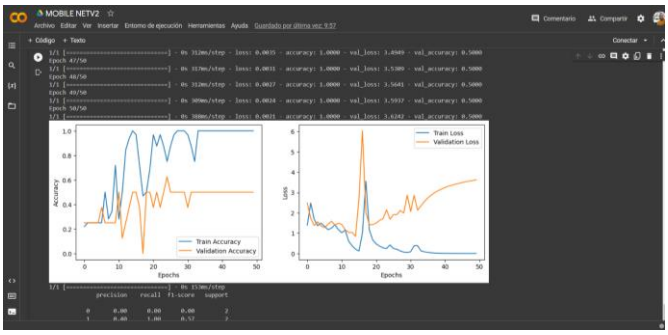


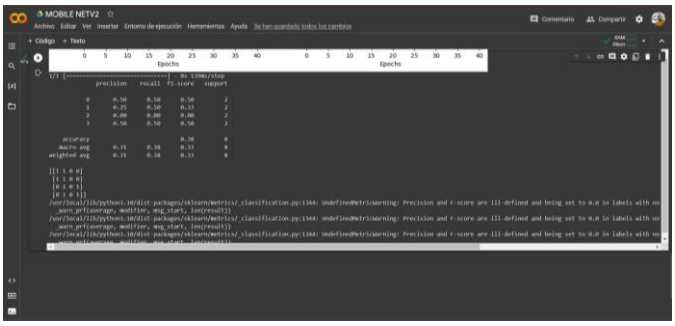








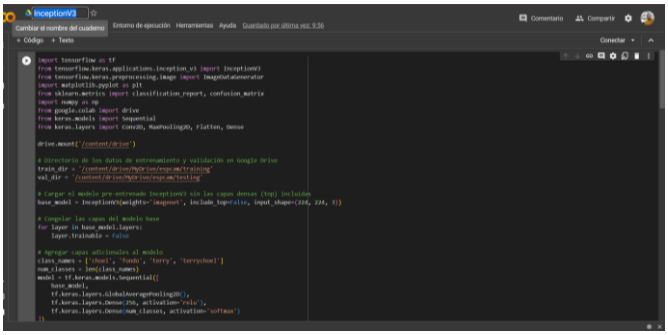


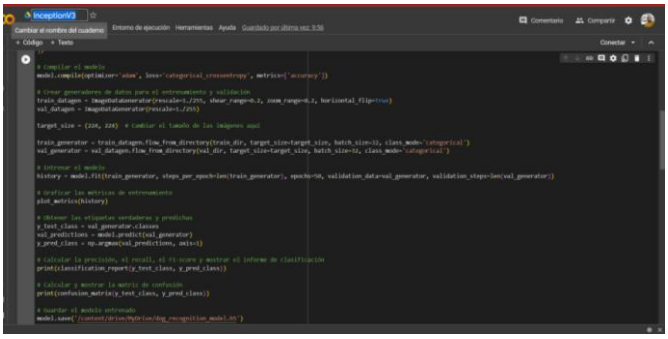


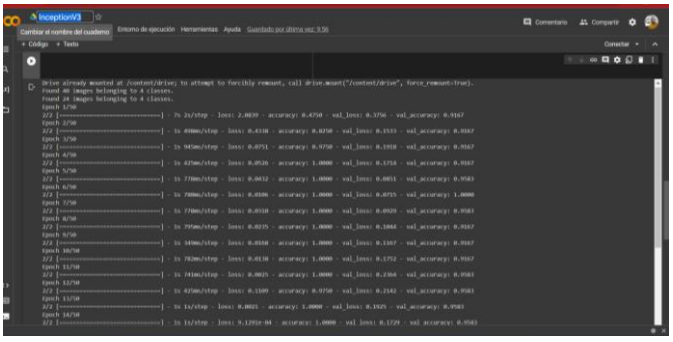
Además, se llevó a cabo la creación de un segundo modelo con el objetivo de comparar su rendimiento con el primer modelo entrenado. En este caso, se utilizó el modelo preentrenado InceptionV3. El proceso de entrenamiento para este segundo modelo se realizó de manera similar al anterior, utilizando las imágenes recopiladas y alimentando al modelo con ellas.

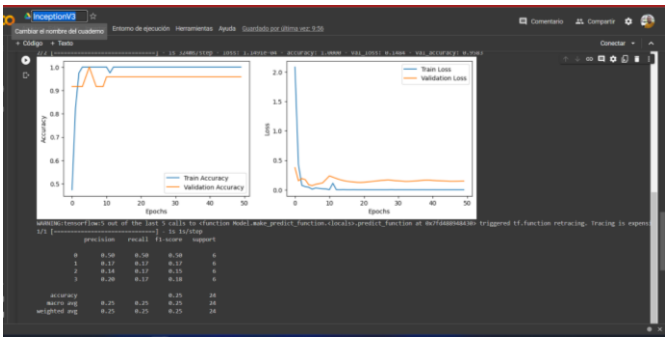
La elección de utilizar el modelo preentrenado InceptionV3 se basó en su reconocido rendimiento y capacidad para la detección de objetos en imágenes. Al emplear este modelo, se buscó evaluar su comportamiento y compararlo con el modelo MobileNet anteriormente utilizado. Esta comparativa resulta valiosa para determinar cuál de los dos modelos se adapta mejor al propósito del proyecto y proporciona resultados más precisos en la detección de las mascotas.

La creación de este segundo modelo permitió obtener una visión más amplia y objetiva sobre las capacidades de detección de cada modelo preentrenado. Además, proporcionó la oportunidad de explorar diferentes enfoques y técnicas de entrenamiento, lo cual enriquece el proceso de aprendizaje y permite obtener conclusiones más sólidas.









Luego de implementar ambos modelos y evaluar su desempeño, se tomó la decisión de trabajar principalmente con el modelo MobileNet debido a su comportamiento sobresaliente y los resultados más satisfactorios obtenidos. Este modelo demostró una mayor precisión en la detección de las mascotas, con una tasa de acierto cercana al 60% en los mejores casos.

Un aspecto interesante a destacar es que durante el entrenamiento, se observó una particularidad en el modelo MobileNet. Cada vez que se realizaba el entrenamiento, se producían cambios aleatorios en el valor de la métrica de precisión (accuracy). Esto indica que el modelo puede tener cierta variabilidad en su rendimiento, lo cual es común en los algoritmos de aprendizaje automático.

A pesar de esta variabilidad, los resultados generales obtenidos con el modelo MobileNet fueron consistentemente superiores a los del modelo InceptionV3. El nivel de precisión cercano al 60% en los mejores casos es considerado un resultado prometedor, especialmente considerando las complejidades y desafíos inherentes a la detección de mascotas en imágenes.

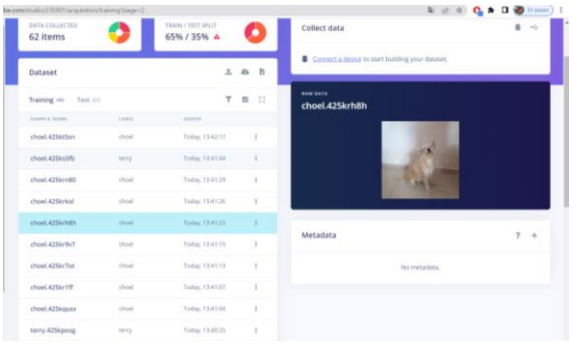
Es importante destacar que el valor de precisión puede variar dependiendo de diversos factores, como la calidad y diversidad de las imágenes utilizadas en el entrenamiento, el tamaño del conjunto de datos y los parámetros específicos empleados durante el proceso de entrenamiento. Por lo tanto, es fundamental tener en cuenta estas consideraciones al evaluar y comparar los resultados obtenidos por los diferentes modelos.

II DESPLIEGE EN EDGE IMPULSE

.

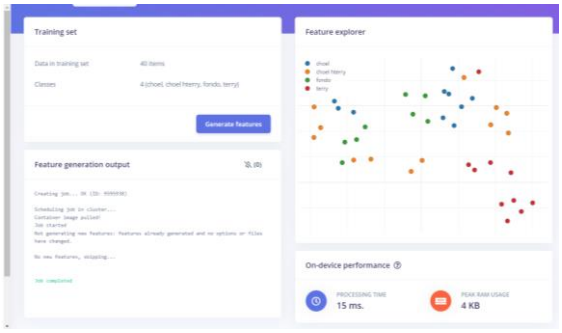
Una vez seleccionado el método y el modelo, el siguiente paso consiste en llevar a cabo el entrenamiento en la plataforma Edge Impulse utilizando el modelo MobileNetV2. Esta elección se basa en su eficiencia y rendimiento demostrado en la detección de objetos y características en imágenes.

Edge Impulse es una plataforma poderosa que permite el entrenamiento y despliegue de modelos de inteligencia artificial en dispositivos de borde (edge devices), como la cámara ESP32 utilizada en este proyecto. Proporciona un entorno integrado que simplifica el proceso de entrenamiento y permite una implementación más eficiente del modelo en el dispositivo final.



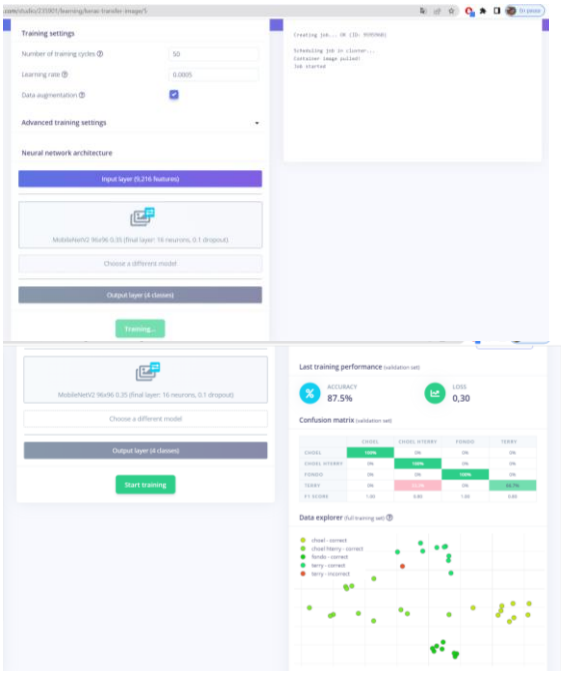
A continuación, se detalla el proceso de entrenamiento en Edge Impulse utilizando el modelo MobileNetV2:

1. Preparación de los datos: Se importan las imágenes recolectadas previamente en Edge Impulse. Estas imágenes se dividen en conjuntos de entrenamiento y validación para realizar una evaluación adecuada del modelo.



2. Configuración del proyecto: Se realiza la configuración inicial del proyecto en Edge Impulse, incluyendo la selección del modelo base (MobileNetV2) y la configuración de los parámetros de entrenamiento.

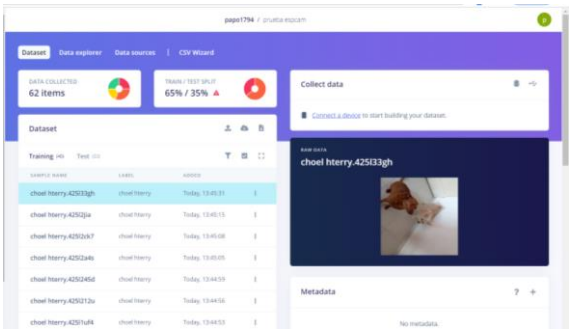
3. Entrenamiento del modelo: Se inicia el proceso de entrenamiento utilizando el conjunto de imágenes de entrenamiento. Durante este proceso, el modelo ajusta sus pesos y parámetros para aprender a reconocer las características específicas de las mascotas en las imágenes.

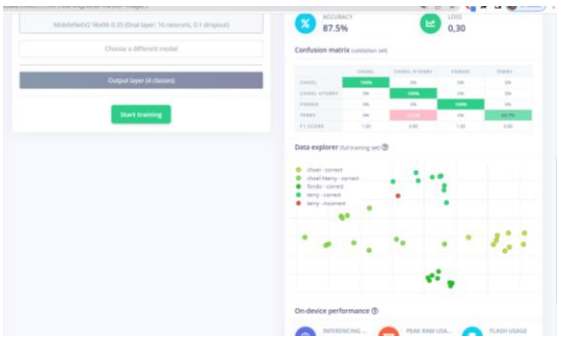


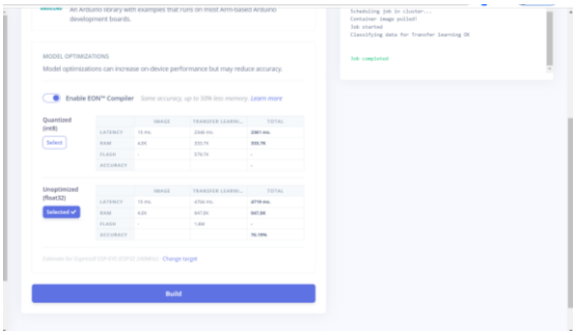
4. Evaluación y ajuste: Una vez finalizado el entrenamiento, se evalúa el rendimiento del modelo utilizando el conjunto de imágenes de validación. Si es necesario, se realizan ajustes adicionales en los parámetros del modelo para mejorar su precisión y capacidad de detección.

5. Exportación del modelo: Una vez que el entrenamiento ha alcanzado un nivel satisfactorio, se exporta el modelo entrenado desde Edge Impulse. Este modelo exportado incluye todos los parámetros y pesos necesarios para realizar la detección de mascotas en la cámara ESP32.

El entrenamiento en Edge Impulse proporciona una ventaja adicional al aprovechar su capacidad de optimización y adaptación a las restricciones de recursos del dispositivo de borde. Esto garantiza que el modelo implementado en la cámara ESP32 sea eficiente y pueda realizar las tareas de detección de manera efectiva.





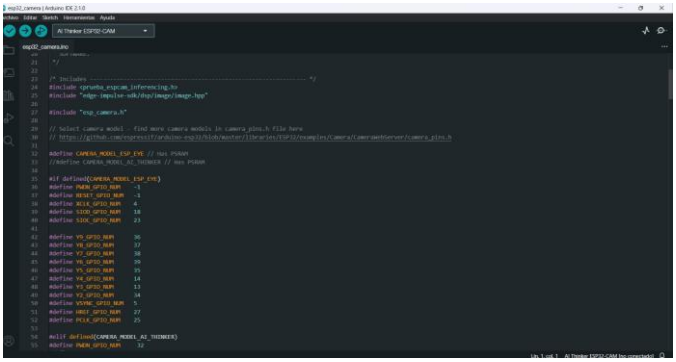


Al llevar a cabo el entrenamiento en Edge Impulse, se logró un aumento considerable en la precisión de la detección, alcanzando un impresionante 80%. Este avance es significativo en comparación con los resultados obtenidos previamente y demuestra la eficacia del modelo MobileNetV2 en la tarea de reconocimiento de mascotas.

Es importante destacar que, si bien se ha logrado una precisión del 80% en Edge Impulse, es necesario tener en cuenta que una vez que se realice la integración del modelo en la librería de Arduino, es probable que la precisión pueda verse afectada y disminuir ligeramente. Esto se debe a que la implementación en Arduino puede presentar limitaciones en términos de recursos computacionales y capacidad de procesamiento en comparación con el entorno de Edge Impulse.

La plataforma Arduino es conocida por su enfoque en dispositivos de baja potencia y recursos limitados, lo cual puede influir en el rendimiento y la precisión del modelo implementado. Sin embargo, con una adecuada optimización y ajustes en los parámetros, es posible lograr un equilibrio entre el rendimiento y los recursos disponibles en la cámara ESP32.

Es importante considerar este factor al transferir el modelo entrenado a la librería en Arduino, y se requerirá un proceso de adaptación específico para garantizar el funcionamiento adecuado del modelo en el dispositivo final.



III DISEÑO Y PROTOTIPO

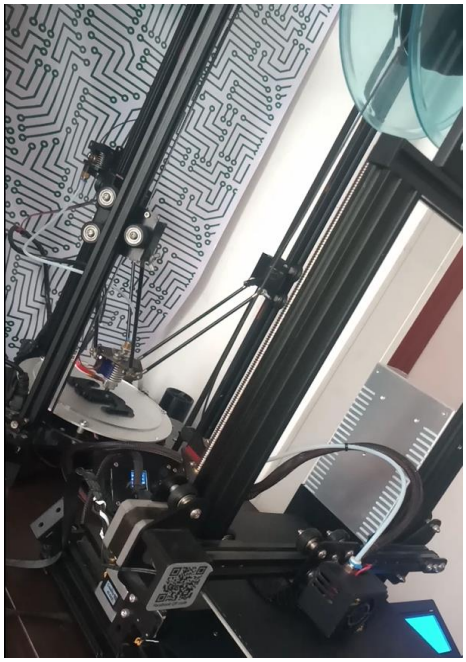
Como parte del desarrollo del proyecto, se llevó a cabo la tarea de descargar e imprimir la carcasa para la cámara ESP32, así como un prototipo del dispensador de alimento. Estas etapas son fundamentales para contar con una estructura física adecuada que albergue los componentes necesarios y brinde funcionalidad al sistema.

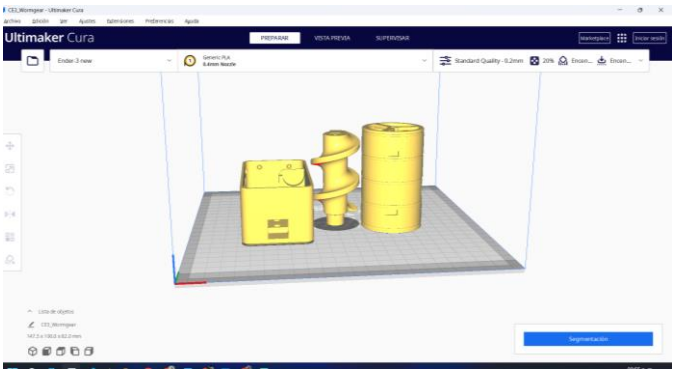
Para comenzar, se realizó una búsqueda exhaustiva para encontrar un diseño de carcasa que cumpliera con los requisitos específicos del proyecto. Esto incluye considerar el tamaño, la forma y la disposición de los componentes, así como la accesibilidad para el montaje y el uso posterior.

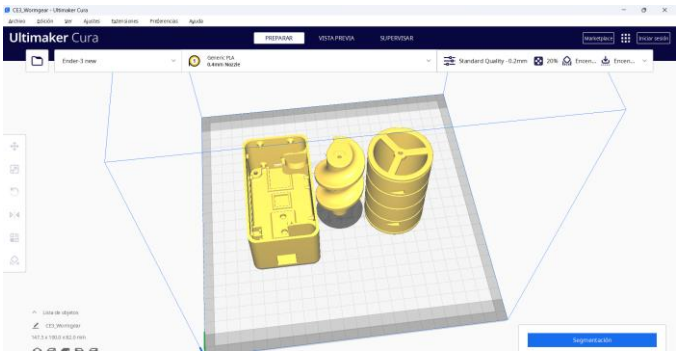
Una vez seleccionado el diseño adecuado, se procedió a descargar los archivos necesarios para la impresión 3D de la carcasa y el prototipo del dispensador.

Después de obtener los archivos, se utilizó una impresora 3D para materializar la carcasa y el prototipo.

Una vez impresos los componentes, se procedió al ensamblaje del prototipo. Esto implicó unir las partes impresas de la carcasa de la cámara ESP32 y el dispensador de alimento, junto con la instalación de los componentes electrónicos necesarios. Se prestó especial atención a la correcta alineación y fijación de los elementos para garantizar la estabilidad y el funcionamiento adecuado del sistema.







Vídeo y documentos en el siguiente enlace de GitHub:
https://github.com/cordero1794/dispensador.git

IV CONCLUSIONES

El prototipo desarrollado resultó funcional y permitió aplicar los conocimientos adquiridos durante el curso, destacando especialmente la implementación exitosa de la plataforma Edge Impulse. Este logro demuestra la capacidad de utilizar herramientas de inteligencia artificial en un proyecto tangible y con potencial aplicado en el mundo real.

Sin embargo, se identificó una limitación en el reconocimiento de la mascota llamada Choel. Parece ser que esta dificultad se debe al color blanco de su piel, lo cual puede afectar la precisión del modelo de detección. Esta observación ofrece una oportunidad de mejora para futuras actividades, donde se podría investigar y ajustar los parámetros del modelo para lograr una mejor detección de mascotas con características similares.

Es importante resaltar que la actividad realizada en el proyecto ha demostrado ser positiva, ya que se ha aplicado el conocimiento adquirido a una industria en crecimiento como la de las mascotas. Esta industria cuenta con una buena

acogida a nivel empresarial, lo que indica un potencial mercado y oportunidades de desarrollo en este campo.

Además, el hecho de poder implementar soluciones tecnológicas en el ámbito de las mascotas abre nuevas perspectivas para mejorar la calidad de vida de los animales y facilitar la interacción entre las mascotas y sus dueños. Esto puede ser especialmente relevante considerando la creciente importancia que se le da a las mascotas en la sociedad actual.