

Herramientas de Machine Learning para IoT

1st Marcos Darío Aranda

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Universidad Nacional de Catamarca
Catamarca, Argentina
marcos.aranda@ues21.edu.ar

2nd Eduardo Enrique Piray

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Córdoba, Argentina
edupiray@gmail.com

3rd Paola Inés Beltramini

Grupo de Internet de las Cosas - GloT
Universidad Nacional de Catamarca
Catamarca, Argentina
pbeltramini@tecn.unca.edu.ar

4th Ricardo Elian González

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Chaco, Argentina
eliangonzalez045@gmail.com

5th Gustavo Andrés Futo

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Buenos Aires, Argentina
gustavoandresfu@gmail.com

5th Martin Gianotti

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Buenos Aires, Argentina
lonasmartin@gmail.com

Resumen—El Machine Learning se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. En la actualidad el IoT permite la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, éste se ha convertido en uno de los términos más populares en la industria tecnológica, utilizando para ello dispositivos de bajo consumo. Poder lograr que estos dispositivos logren incorporar machine learning, podría ser de gran utilidad para el futuro. Es por ello que el presente trabajo se centra en identificar herramientas que permitan generar modelos de machine learning para la clasificación y detección de objetos, y luego ser implementado en dispositivos de bajo consumo.

Index Terms—Machine learning, IoT, Microcontroladores, Clasificación, Detección.

I. Introducción

El Machine Learning, es una rama de la Inteligencia Artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. Existe un tipo de aprendizaje llamado supervisado, el cual consiste en entrenar un algoritmo de Machine Learning con datos de entrenamientos etiquetados. Además con el uso de Internet de las Cosas (IoT - Internet of Things), que se entiende como la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, el mismo se ha convertido en los últimos años en uno de los términos más populares en la industria tecnológica gracias al enorme número de dispositivos que cuentan con la posibilidad de interconectarse [10]. Los dispositivos que cuentan con inteligencia artificial pueden ejecutar distintos procesos análogos al comportamiento humano, como la devolución de una respuesta por cada entrada (similar a los reflejos de los seres vivos), la búsqueda de un estado entre todos los

posibles según una acción o la resolución de problema mediante una lógica formal [1].

El objetivo de este trabajo es identificar cuales son las herramientas utilizadas para aprendizajes automáticos, que puedan ser utilizadas con un enfoque de código abierto para Machine Learning a gran escala para dispositivos de bajo consumo. [2]. Con el presente trabajo se busca contribuir a la clasificación y detección inteligente para dispositivos de bajo consumo en animales, para posteriormente el registro de información.

II. Marco Teórico

A continuación se realiza una breve descripción de las herramientas identificadas:

A. Aprendizaje Automático

En la detección de objetos se suelen emplear técnicas de Aprendizaje Automático o Machine Learning, esta área consiste en crear sistemas que pueden aprender por sí mismos, capaces de identificar una serie de patrones complejos a partir de datos de entrada. Este proceso se realiza mediante algún algoritmo de aprendizaje [3]. Existen cuatro tipos de aprendizaje automático:

- Aprendizaje supervisado: Se trata de un tipo de aprendizaje donde al sistema se le alimenta con datos de entrenamiento etiquetados. Una vez se ha completado el entrenamiento, se pueden introducir nuevos datos al sistema que no están etiquetados anteriormente, puesto que al estar entrenado el sistema es capaz de reconocer patrones para identificar los datos correspondientes. En este tipo de aprendizaje están los denominados problemas de clasificación y los de regresión [3].
- Aprendizaje no supervisado: En este tipo de aprendizaje se alimenta el sistema sin datos etiquetados.

Uno de los problemas más comunes de este tipo de aprendizaje automático es el de agrupamiento o clustering, en el que el sistema aprende a organizar los datos en distintos grupos, de modo que cada grupo contenga datos similares entre sí [3].

- Aprendizaje semisupervisado: Es una práctica que se encuentra en un punto medio entre el aprendizaje supervisado y el no supervisado. De esta manera, solo se utilizan un grupo mínimo de etiquetas. No obstante, la mayoría son grupos de datos no etiquetados, pues aumentan los costes, pero son útiles para cumplir con los objetivos. Aunque hay una supervisión sobre cómo actúa la máquina, no es un trabajo que se realiza a lo largo del trabajo con la máquina. Mientras que tendrás que etiquetar algunos resultados manualmente, otros serán propuestas automáticamente por el machine learning [11].
- Aprendizaje por refuerzo: Es una práctica que está basada en recompensar los comportamientos deseados, mientras que los no deseados serán penalizados. Es un proceso que está basado en la retroalimentación, pues la máquina aprenderá de las experiencias, a partir del desarrollo y rendimiento [11].

B. TinyML: Machine Learning

TinyML es una marca registrada de TinyML Foundation, esta asociación promueve el desarrollo de tecnología para aplicaciones de machine learning en dispositivos con capacidades de poco procesamiento [6]. Por tanto, hacer que los microcontroladores puedan ejecutar modelos de machine learning tiene el potencial para abrir nuevos escenarios disruptivos en todas las industrias. Otro aspecto relevante es el impacto ecológico que podría tener la adopción generalizada de las prácticas TinyML, que permitirían reducir el consumo de energía de cara al futuro [5].

C. Clasificación de imágenes

La tarea de identificar lo que representa una imagen se denomina clasificación de imágenes. Un modelo de clasificación de imágenes está entrenado para reconocer varias clases de imágenes. Por ejemplo, puede entrenar a un modelo para que reconozca fotografías que representen diferentes de animales. TensorFlow Lite proporciona modelos optimizados previamente entrenados que puede implementarse en aplicaciones móviles [7].

D. Detección de objetos

Un problema más complejo consiste en no solo decir si el objeto está o no presente en la imagen en cuestión, sino también en qué región de la imagen se encuentra dicho objeto [8].

III. Herramientas

A. TensorFlow Lite

Es un framework utilizado para implementación de modelos de aprendizaje automático que cuenten con poco poder de procesamiento entre ellos los dispositivos móviles, microcontroladores y dispositivos de IoT [4]. Las características más relevante de TensorFlow Lite son:

- Optimizado para el aprendizaje automático integrado en el dispositivo: No existe transferencia de datos con un servidor, logrando así disminuir la latencia. Los datos no salen del dispositivo. Con respecto a la conectividad no se necesita conexión a Internet, el tamaño del modelos y el binario son reducidos y además se obtienen mejoras en el rendimiento del consumo de energía.
- Compatible con plataformas iOS, Android, Linux embebido y microcontroladores. Soporta lenguajes Java, Swift, Objective-C, C++ y Python.
- Rendimiento para la aceleración de hardware y optimización de modelos.



Figura 1. Flujo de trabajo de TensorFlow Lite.

Con la incorporación del aprendizaje automático en microcontroladores, podemos potenciar la inteligencia de millones de dispositivos que usamos en nuestra vida, incluidos los electrodomésticos y dispositivos de la Internet de las Cosas, sin depender de hardware costoso ni de una conexión a Internet estable, que suele estar condicionada por restricciones de ancho de banda y energía, y que además, genera una latencia alta. Esto puede contribuir también a preservar la privacidad, ya que ningún dato sale del dispositivo [4]. TensorFlow Lite para microcontroladores se codificó en C++ 11 y requiere una plataforma de 32 bits. Se probó de manera exhaustiva con muchos procesadores basados en la arquitectura de la serie ARM Cortex-M y se trasladó a otras arquitecturas, como ESP32. El marco de trabajo está disponible como una biblioteca de Arduino. También puede generar proyectos para entornos de desarrollo, como Mbed. Es de código abierto y se puede incluir en cualquier proyecto de C++ 11 [9].

B. MediaPipe

Proporciona un conjunto de bibliotecas y herramientas para aplicar rápidamente técnicas de inteligencia artificial y machine learning. Permite conectar estas soluciones a las aplicaciones de manera inmediata, personalizarlas según necesidades y utilizarlas en varias plataformas de



Figura 3. Etiquetado de imágenes con labellmg.

desarrollo. MediaPipe es parte de un proyecto de código abierto [13], en la Fig. 2 se observa lo que incluye el paquete de soluciones [12].

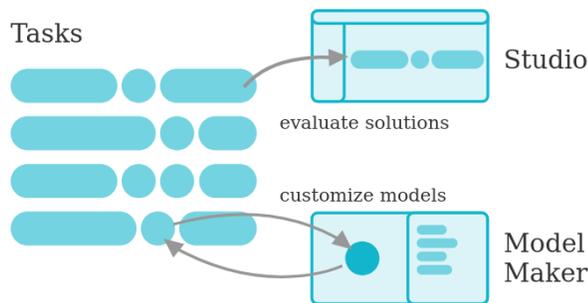


Figura 2. Paquete de soluciones de MediaPipe.

- 1) Bibliotecas y recursos que proporcionan la funcionalidad principal para cada solución:
 - MediaPipe Tasks: APIs y bibliotecas multiplataforma para implementar soluciones.
 - Modelos de MediaPipe: son modelos previamente entrenados y listos para ejecutarse para usar con cada solución.
- 2) Herramientas que permiten personalizar y evaluar las soluciones:
 - MediaPipe Model Maker: personaliza modelos para soluciones con datos.
 - MediaPipe Studio: visualiza, evalúa y compara soluciones en el navegador Web.

IV. Desarrollo

A. Búsqueda de datos

Para este trabajo se han utilizado para realizar pruebas de entrenamiento un dataset de Open Images Dataset V7, el cuál es un servidor con aproximadamente 9 millones de imágenes con etiquetas, cuadros delimitadores de objetos, máscaras de segmentación de objetos, relaciones visuales y narraciones localizadas, se eligió la categoría Sheep. Posteriormente se procedió a la

descargar de categoría de las imágenes. Utilizando la herramienta OIv4 ToolKit se procedió a la descarga de la misma.

B. Etiqueta en imágenes

Las etiquetas se generan en un formato XML, por cada una de ellas se encuentra asociada cada imagen, se utilizó la herramienta labellmg para etiquetar cuadros delimitadores de objetos en imágenes.

C. Entrenamiento

Para realizar el entrenamiento se utilizó Google Colab, el cual es un servicio colaborativo, que permite escribir y ejecutar código de Python en el navegador, el cual cuenta con las siguientes características:

- No requiere configuración previa.
- Se puede acceder a una GPU sin costo alguno.
- Facilidad para compartir.

Como continuidad de este trabajo se espera validar el modelo y obtener resultados en dispositivos de bajo consumo utilizando MediaPipe.

V. Conclusiones

En este trabajo se describió dos herramientas de aprendizaje automático que pueden ser utilizadas para la implementación de modelos TinyML en dispositivos de bajo consumo. Se identificó que ambas herramientas son de código abierto y brindan un gran soporte con respecto a la documentación para la implementación de modelo.

Como trabajo a futuro se espera validar el modelo de clasificación y detección de objetos, en dispositivos con recursos limitados, como móviles, microcontroladores o dispositivos IoT.

Agradecimientos

Agradecemos a las autoridades de la Universidad Siglo 21 por el apoyo incondicional para el desarrollo de la investigación.

Referencias

- [1] Julián Pérez Porto and Ana Gardey: Definición de inteligencia artificial. Qué es, Significado y Concepto : <https://definicion.de/inteligencia-artificial/>: 28 11 2022
- [2] Michael Bowles: Machine Learning in Python® : Essential Techniques for Predictive Analysis. Published by John Wiley and Sons, Inc: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119150202>
- [3] Abu-Mostafa, Yaser S: Learning from data: a short course (2012)
- [4] TensorFlow Lite : <https://www.tensorflow.org/lite/guide?hl=es-419> : 02 06 2023
- [5] TinyML : <https://www.sngular.com/es/tinyml-machine-learning/>: 20 03 2023
- [6] Pete Warden and Daniel Situnayake: TinyML : Published by O'Reilly Media, Inc. : 2020
- [7] Clasificación de imágenes : <https://www.tensorflow.org/lite/examples>: 20 03 2023
- [8] Rozada Raneros, Saúl and others: Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning (2021)
- [9] TensorFlow microcontroller: <https://tensorflow.org/lite/microcontrollers> : 02 06 2023

- [10] Marcos D. Aranda, Paola I. Beltramini, Jesús E. Cano, Luis D. Villagrán, Juan P. Moreno, Sergio H. Gallina, Oscar Ariel Herrera Conegliano : IoT aplicado a la ganadería extensiva : RADI (PDTS) ISSN 2314-0925 : Volumen 17: 02 05 2021
- [11] Aprendizaje Automático : <https://www.keepcoding.io/blog/tipos-de-aprendizaje-automatico/> : 27 04 2023
- [12] Google. MediaPipe Solutions Guide: <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419> : 20 08 2024
- [13] Google. MediaPipe GitHub Repository. <https://github.com/google-ai-edge/mediapipe> : 20 08 2024