UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Informática de Sistemas y Computadores

Comparativa de plataformas software para TinyML

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Computadores y Redes

AUTOR/A: Sanchez Acevedo, Juan

Tutor/a: Manzoni, Pietro

CURSO ACADÉMICO: 2023/2024

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA
DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y COMPUTADORES

Máster en Ingeniería de Computadores y Redes
Trabajo Fin de Máster

Comparativas de plataformas software para TinyML

Autor: *Juan Sebastián Sánchez Acevedo*

Director: *Pietro Manzoni*

Julio, 2024

AGRADECIMIENTOS

Como agradecimiento especial, reconozco el apoyo de mi director de trabajo de fin de máster Pietro Manzoni, el cual me guio y apoyó en la creación de este documento, correspondió a mis solicitudes de guía y accedió a dirigirme en este proyecto, en cuanto a mis compañeros agradezco a los ingenieros David Fernando Becerra, Jorge Mosquera y David Bustamante que me apoyaron no solo en la búsqueda de información referente a TinyML, sino también me brindaron su apoyo en la recolección de información de interés relacionada a las plataformas software para TinyML, de forma personal, agradezco a Estefanía Osorio y Laura Gómez, ya que me motivaron en las largas jornadas que pasé escribiendo, buscando información y creando mi trabajo de final de máster, que creyeron en mí y en la capacidad que tenía para afrontar este reto de emigrar de mi tierra natal para realizar mis estudios de posgrado.

También quiero aprovechar y agradecer a las directivas y todos los profesores que impartieron su catedra en cada una de las asignaturas vistas en el máster de ingeniería de computadores y redes, sin la información que me brindaron a lo largo de las cátedras, este proyecto no hubiese sido posible.

Por último, agradezco a las directivas del máster y a la Universidad Politécnica de Valencia por darme la oportunidad de cursar mis estudios de Posgrado en sus instalaciones y ser parte de este selecto grupo del cual me enorgullece ser parte.

RESUMEN

El presente documento recopila el análisis e investigación sobre las herramientas de software para la implementación de TinyML, la cual es una técnica de implementación de inteligencia artificial en el área de machine learning, mediante el despliegue de redes neuronales en dispositivos de bajos recursos, como placas de desarrollo con periféricos capaces adquirir información a partir de imágenes o sonidos, entre otros parámetros.

Las aplicaciones software para TinyML en las cuales se va a enfocar este documento son aquellas que cuenten con reconocimiento y bases de información solidas brindadas por la comunidad de las plataformas o datos relevantes obtenidos a partir de los sitios oficiales de las mismas. Se desarrollará un análisis y comparación de cada una de las plataformas Software y adicional a esto se presentan pruebas de uso de las plataformas más utilizadas mediante propuestas sencillas para profundizar en el funcionamiento de las tecnologías dispuestas para TinyML.

En cada entorno de desarrollo, se examinan los bloques de aprendizaje predispuestos por cada plataforma, las ventajas, las placas de desarrollo de bajos recursos compatibles y datos de interés relevantes de cada una de las plataformas software analizadas en el presente documento, cabe aclarar que las herramientas comparadas en este documento son de uso gratuito, con opciones de pago o de pago por lo cual uno de los enfoques importantes es entender las limitantes y ventajas que tienen estas plataformas. Así mismo se busca ofrecer una crítica constructiva, implementando pruebas hardware de algunas plataformas, en busca de una conclusión objetiva desde una perspectiva profesional, que busca concluir cuál de las plataformas software presentadas en este documento es la mejor para la iniciación y realización de proyectos de modelos de aprendizaje automática.

Palabras Clave: TinyML, Hardware, Software, Microcontrolador, Inteligencia artificial.

ABSTRACT

This document compiles the analysis and research on software tools for the implementation of TinyML, which is a technique for implementing artificial intelligence in machine learning, through the deployment of neural networks on low-resource devices, such as motherboards. development with peripherals capable of acquiring information from images or sounds, among other parameters.

The software applications for TinyML on which this document will focus are those that have recognition and solid information bases provided by the platform community or relevant data obtained from their official sites. An analysis and comparison of each of the Software platforms will be developed and in addition to this, use tests of the most used platforms are presented through simple proposals to deepen the operation of the technologies available for TinyML.

In each development environment, the learning blocks provided by each platform, the advantages, the compatible low-resource development boards, and relevant data of interest for each of the software platforms analyzed in this document are examined. It is worth clarifying that the tools compared in this document are free to use, with paid or paid options, so one of the important focuses is to understand the limitations and advantages that these platforms have. Likewise, it seeks to offer constructive criticism, implementing hardware tests of some platforms, in search of an objective conclusion from a professional perspective, which seeks to conclude which of the software platforms presented in this document is the best for the initiation and implementation of development projects. machine learning models.

Keywords: TinyML, Hardware, Software, Microcontroller, Artificial Intelligence.

INDICE DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

1 INTRODUCCIÓN

La comparación entre las plataformas de software, como TensorFlow, Edge Impulse, Efinix y las librerías desarrolladas, como Arduino Machine Learning, para proyectos que involucren el aprendizaje de redes neuronales y apliquen TinyML, es crucial para el desarrollo y exploración de nuevas metodologías de enseñanza. Además, permite aumentar el conocimiento sobre la inteligencia artificial y la profundidad de los proyectos realizables mediante la implementación de microcontroladores de bajo rendimiento, como la placa de desarrollo Arduino Nano 33 BLE. Esta última posibilita el desarrollo de sistemas de inteligencia artificial a bajo costo económico y computacional. También permite la integración o compatibilidad con múltiples entornos de desarrollo de software, abriendo las puertas tanto a desarrolladores en etapa de iniciación como a experimentados en el mundo de TinyML. Por otro lado, los entornos de desarrollo de Arduino IDE simplifican las tareas de aprendizaje, programación y despliegue de proyectos de inteligencia artificial.

Los costos de estos microcontroladores de bajos recursos facilitan que la implementación de nuevos proyectos sea factible y atractiva para la exploración de TinyML. Asimismo, la alta disponibilidad de dispositivos de bajos recursos que aceptan machine learning para Tiny, mejoran la capacidad de expansión de entornos de desarrollo de TinyML, como el caso de las aplicaciones de desarrollo software de redes neuronales y enfocadas en TinyML. Estos aplicativos son compatibles con microcontroladores capaces de aceptar algoritmos de inteligencia artificial, lo que resulta en una entrada sencilla al aprendizaje de inteligencia artificial debido a sus interfaces intuitivas en su mayoría de casos, por lo que se simplifica la creación de modelos de machine learning.

Otro aspecto por recalcar es la optimización de modelos que requieren poca capacidad de memoria y procesamiento, esencial para garantizar que el rendimiento de los modelos de aprendizaje de inteligencia artificial sea el adecuado. Esto se refleja tanto en el despliegue de aplicaciones mediante aplicaciones de software como lo son Edge Impulse, TensorFlow y Efinix y librerías como la Arduino Machine Learning Library, estas últimas tienen el objetivo de ser compatibles con el microcontrolador Arduino que soporten debido a su eficiencia en proyectos hardware con un consumo moderado de memoria. Las placas de desarrollo para TinyML también cuentan con múltiples sensores, como acelerómetros, colorímetros, micrófono, entre otros, que facilitan el desarrollo de modelos de aprendizaje de inteligencia artificial, por lo cual en resumen, se llevará a cabo un estudio y comparación de las plataformas software mencionadas en este escrito para deducir sus ventajas, desventajas, limitaciones, escalabilidad e interfaces de uso, los cuales son aspectos necesarios para la implementación de modelos de aprendizaje TinyML, posteriormente se realizará pruebas con plataformas software seleccionadas con base en la comparativa a realizar, con el fin de analizar qué ventajas y desventajas, como también que capacidad de expansión tienen estas en el mundo de TinyML.

1.1 MOTIVACIÓN

Este trabajo de final de máster se realiza a partir de la necesidad por parte del alumno de realizar una investigación en los temas de interés, en este caso el acercamiento al TinyML, el cual era objeto de interés debido a la gran expansión de los sistemas de inteligencia artificial, que por un lado han sido incentivados por parte del temario impartido por el profesorado del Máster en ingeniería de computadores y redes, así mismo el interés personal por el área del internet de las cosas (IoT), el cual representa un aspecto importante en la capacidad de despliegue de TinyML y por último la necesidad de plantear puntos de partida y que estudiantes en próximos ciclos académicos puedan retomar las bases de investigación plasmadas en este documento e indaguen en nuevas técnicas de implementación de TinyML, mediante proyectos de mayor complejidad los cuales puedan aprovechar el 100% de los aplicativos software que se enfocan en inteligencia artificial para TinyML.

1.2 OBJETIVOS

El objetivo general es elaborar una comparativa entre las diversas herramientas software que puedan ser utilizadas para soluciones de inteligencia artificial TinyML, en estas comparativas se reflejará cualidades particulares de cada una de las plataformas software obtenidas a partir de la investigación, por otro lado, al realizarse esta comparativa se deberá elegir las mejores plataformas de software para TinyML teniendo como base la información anteriormente descrita, con el fin de realizar ejercicio sencillos de soluciones de TinyML, en donde se analizarán resultados prácticos mediante un hardware que se pueda utilizar en las aplicaciones seleccionadas. En principio otro de los objetivos generales es realizar un estudio de las plataformas con el fin de dejar abierta la puerta a la evolución de proyectos de machine learning con TinyML, partiendo como base este proyecto de carácter investigativo que contendrá conceptos relacionados con inteligencia artificial, redes neuronales, machine learning, como también formas de uso de las plataformas software e implementación de hardware mediante estos aplicativos, para así generar a partir de estas soluciones proyectos de carácter práctico con mayor complejidad que den soluciones particulares e innovadoras a problemáticas relacionadas con implementación de TinyML.

1.3 OBJETIVOS ESPECIFICOS

- Realizar una comparativa de las diferentes plataformas software de TinyML
- Realizar elección de plataformas para realizar implementación Hardware mediante un microcontrolador de bajos recursos a elección del autor del trabajo de final de máster.
- Analizar las plataformas seleccionadas con el fin de obtener información ventajas, desventajas, facilidad de uso de interfaz y complejidad de estas.
- Realizar una guía detallada implementando un proyecto hardware de bajos recursos mediante el microcontrolador seleccionado por parte del autor de este documento con el fin de que funcione como bases de conocimiento sobre los softwares de TinyML para futuras aplicaciones.

1.4 IMPACTO ESPERADO

El trabajo de final de máster sobre comparativas de plataformas software para TinyML, pretende generar conocimiento desde cero sobre las herramientas que pueden llegar a usarse para la elaboración de redes neuronales, conceptos de machine learning esenciales y paso a paso de la creación de proyectos en las plataformas software que a criterio del investigador de este documento serán seleccionadas. Este documento pretende expandir el conocimiento del lector sobre lo que supone el machine learning y su utilización como TinyML, así mismo pretende ser punto de partida para elaboración de proyectos de carácter avanzado que contemplen algoritmos más avanzados que utilicen los modelos de aprendizaje automático que TinyML utiliza para la elaboración de sistemas de inteligencia artificial.

1.5 METODOLOGÍA

Después de revisar la información disponible sobre plataformas de software para TinyML, me he dado cuenta de que hay una notable escasez de datos, especialmente en lo que respecta a comparaciones entre distintas plataformas. A pesar de haber consultado fuentes como documentos oficiales y artículos de investigación, la información encontrada no solo es limitada sino también a menudo repetitiva. Esto presenta un desafío considerable, ya que buscamos aportar nuevos conocimientos y perspectivas frescas sobre las técnicas de TinyML.

Ante esta situación, he decidido adoptar un enfoque más detallado y personalizado. Planeo realizar una revisión exhaustiva de cada plataforma de software para TinyML que he encontrado. Esta revisión se enfocará en evaluar y presentar las capacidades únicas de cada plataforma, aunque he tenido que ser cuidadoso en el análisis de la información antes de incluirla en este documento.

Además, dada la falta de información comparativa y de desarrollo de aplicaciones de software para TinyML, propongo llevar a cabo un análisis profundo de las plataformas identificadas. Este análisis se concentrará en aspectos clave como la funcionalidad, la interfaz de desarrollo, la facilidad de uso y la compatibilidad con una variedad de dispositivos, incluyendo GPUs, placas de desarrollo de bajo recurso y dispositivos móviles.

Una vez completado este análisis, mi objetivo es realizar una comparación detallada de estas características. Posteriormente, como parte de este estudio, presentaré una implementación práctica de un modelo de aprendizaje TinyML en dos de las plataformas de software analizadas, utilizando una placa de desarrollo Arduino Nano 33 BLE Sense. Con esto, espero ofrecer una visión clara y útil sobre el estado actual y las posibilidades de las plataformas de software para TinyML

1.6 ESTRUCTURA

La estructura del trabajo del final de máster que encontrará a partir de este punto será:

- Capítulo 2 – Estado del arte (Necesario para que el lector tenga conocimiento de que trata el documento y entienda los conceptos utilizados en este trabajo de final de más), en este capítulo se encontrarán subcapítulos que tendrán relación a TinyML y las plataformas de software que implementan TinyML, como también contendrán comparativas de las mismas y por último se encontrara con una comparativa de las plataformas software investigadas, junto con la propuesta

para realizar el análisis de plataformas seleccionadas para implementación de hardware mediante una placa de desarrollo seleccionada por el autor del trabajo de final de máster.

- Capítulo 3 – Análisis del problema, En este capítulo se encuentra el desarrollo de las aplicaciones Hardware utilizando las herramientas de software que se encuentran en los aplicativos mencionados en el capítulo 2, los cuales emplean tecnología TinyML, en este apartado también se encontrarán los resultados de los proyectos realizados y el análisis de estos.
- Capítulo 4 – Conclusiones, también se encontrará un apartado que hable sobre la relación del trabajo desarrollado con los estudios cursados y trabajos futuros que pueden llegar a implementarse a partir de este trabajo de final de máster.
- Capítulo 5 – Referencias.

2 ESTADO DEL ARTE

Para entender el desarrollo del trabajo de comparativas de plataformas software para TinyML, antes se debe realizar una investigación de los temas que se abarca el proyecto de trabajo de final de máster, los cuales se presentarán a continuación en la metodología de cada uno de los elementos a implementar, como lo es la inteligencia artificial, enfocada en TinyML, el concepto de red neuronal, las diferentes plataformas que formarán parte de la comparativa en el desarrollo del trabajo investigativo, por último el microcontrolador seleccionado para los montajes hardware que se presentarán en este proyecto.

2.1 MACHINE LEARNING

Antes de hablar de TinyML se debe hablar de machine learning, este concepto hace referencia a las herramientas de inteligencia artificial, denominadas IA, que posibilitan el aprendizaje de máquinas teniendo como base los datos suministrados, estos datos suelen ser utilizados para predicción de eventos, clasificación de objetos, colores, sonidos, entre otros.

El concepto de machine learning se planteó con la finalidad de que las máquinas tuviesen la posibilidad de tomar decisiones dependiendo de la cantidad de datos y el nivel de procesamiento que la máquina posea. El machine learning da la capacidad a la máquina de enfrentar eventos, proponer soluciones y tomar decisiones por sí misma, con base al entrenamiento e indagación de experiencias de aprendizajes anteriores. Un ejemplo de machine learning es la predicción de condiciones climatológicas, con base en información de sensores y sistemas embebidos asociados a la máquina, las cuales alimentan la base de datos asociada al aprendizaje.

El aprendizaje de las machine learning se puede dividir en tres modos:

- Aprendizaje por refuerzo
- Aprendizaje supervisado
- Aprendizaje no supervisado

2.1.1 Aprendizaje por refuerzo

En el aprendizaje por refuerzo denominado (RL), la máquina tiene la capacidad de generar ensayos positivos y negativos, que con el tiempo logrará generar más casos positivos debido al aprendizaje de experiencias anteriores, adicionalmente los programas que se basan en el aprendizaje por refuerzo aprenden de cada una de las acciones realizadas en los ensayos, con el fin de realizar rutas de procesamiento de datos optimas. Estos tipos de aprendizaje se destacan por ser altamente utilizados en entornos con reglas específicas como los videojuegos, predicciones financieras y actividades que requieran de optimización de procesos, este tipo de aprendizaje destaca por minimizar la interacción humana ya que recolecta información en cada uno de los ensayos y alimenta la base de datos por sí misma.

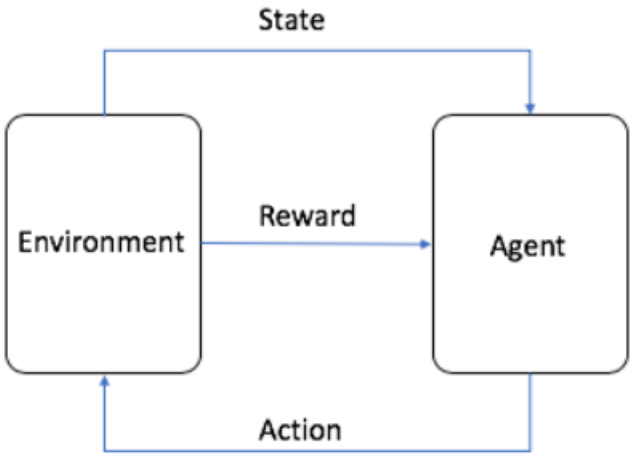

Figura 1 - Bucle de aprendizaje por refuerzo

Tomado de: Emaz, A. and Pérez-Ruiz, U. (1998b) ¿Qué es el aprendizaje mediante refuerzo?, Amazon. Available at: https://aws.amazon.com/es/what-is/reinforcement-learning (Accessed: 01 December 2023).

De acuerdo con la Figura 1 - Bucle de aprendizaje por refuerzo, mediante prueba y error y dependiendo del entorno en donde se despliegue los ensayos, el agente denominado también sistema autónomo, puede generar reglas condicionales para realizar toma de decisiones con el fin de obtener recompensas, las cuales son definidas por el programador de la machine learning, esto se realiza con el fin de obligar a la maquina a tomar decisiones de exploración de nuevas rutas o seleccionar situaciones ya conocidas debido a experiencias pasadas.

2.1.2 Aprendizaje supervisado

El aprendizaje supervisado, es al igual que el aprendizaje por refuerzo una clasificación de machine Learning, Este aprendizaje se basa en datos que cuentan con reconocimientos o etiquetas, esto con el fin de clasificar los datos que se utilizarán para el aprendizaje de la máquina, por otro lado, este tipo de aprendizaje cuenta con la cualidad de ser precisa en la predicción de eventos, debido a la clasificación de datos relevantes y no relevantes para los objetos de investigación.

El aprendizaje supervisado cuenta con formas de realizar estudios:

- Clasificación
- Regresión

En la clasificación se utiliza un algoritmo basado en precisión de datos en las pruebas realizadas, este indica dentro de una base de datos designada, parámetros específicos con el fin de realizar predicciones de etiquetados a los datos de forma acertada, por otro lado en el aprendizaje supervisado por el método de regresión, se utiliza para entender la relación de datos que son dependientes o independientes dentro de cada una de las pruebas realizadas, este aprendizaje utiliza las regresiones lineales, logísticas y polinómicas como base de los algoritmos de programación.

Este tipo de aprendizaje permite dar solución a un amplio espectro de problemas de la vida real, sean cotidianos o corporativos, un ejemplo claro de aplicación de este tipo de aprendizaje es el uso de las carpetas de los correos electrónicos, segmentando o categorizando la importancia de los correos recibidos, en subcarpetas designadas previamente, como Spam, prioritarios, entre otros.

2.1.3 Aprendizaje no supervisado

El aprendizaje no supervisado, a diferencia de su antagonista el aprendizaje supervisado , realiza estudios con base en datos que no requieren de etiquetas específicas, sino que en su lugar recibe datos de entrada definidos y a partir de estos, realiza estudios de patrones , este tipo de aprendizaje es empleado en problemas presentados en clústeres , en donde se solicita el hallazgo de inconvenientes comunes en múltiples dispositivos o en un conjunto de máquinas , por otro lado, este tipo de aprendizaje de machine learning es más económico de implementar y cuenta con un tiempo de enseñanza menor con respecto al aprendizaje supervisado.

Algunos de los retos que presenta el aprendizaje no supervisado son:

- Complejidad en el sistema computacional
- Falta de precisión
- Intervención humana

En la complejidad en el sistema computacional el principal inconveniente es el volumen de los datos de entrenamiento que deben ser ingresados en entrenamiento de la máquina, ya que al no utilizar datos con etiquetas especificas el sistema deberá adaptarse a los patrones que aparezcan con base en la información de entrada, además la falta de precisión de este aprendizaje se debe a la falta de etiquetas y reglas en los datos ingresados en cada uno de los ensayos, por último la intervención humana representa un retardo en tiempos y un enfoque diferente por cada revisor en las variables de salida de cada uno de los ensayos de entrenamiento del aprendizaje no supervisado.

2.2 REDES NEURONALES

Las denominadas redes neuronales son un método de aplicación de la inteligencia artificial (IA), los cuales permiten que las máquinas de procesamiento de datos permitan aprender parámetro y prever eventos próximos, un término acertado para referirse a las redes neuronales es el Deep Learning, este tipo de aprendizaje consiste en la utilización de nodos o dispositivos interconectados en una red semejante al cerebro de los seres humanos, de esta forma las redes neuronales al igual que el cerebro humano es capaz de realizar acciones, aprender de los errores y realizar correcciones, esto suele ser de suma utilidad en campos como la medicina o extracción de información específica de base de datos. De la mano de las redes neuronales se pueden acortar procesos de trabajo, mano de obra humana y comprensión de datos con la ayuda de aprendizajes automáticos. Los procesos en los que se enfocan las redes neuronales son:

2.2.1 Visión artificial

La visión artificial es la herramienta con la que cuentan las máquinas para realizar extracción de datos y procesamiento de información como imágenes o video, bajo este concepto se pueden distinguir objetos como lo hace el ser humano. Este tipo de aplicación funciona para reconocimiento de vehículos, señales de tránsito, usuarios en las zonas, mediante dispositivos como cámaras de videovigilancia.

2.2.2 Reconocimiento de voz

Así como los humanos el reconocimiento de voz en las redes neuronales se utilizan para distinguir personas, dado su acento, su lenguaje y tonalidad de voz, este tipo de redes neuronales suelen ser implementadas en dispositivos como asistentes virtuales y softwares de transcripción. Este tipo de aplicación suele ser enfocado en clasificación de llamadas, subtitulación de videoconferencias y transcripción de reuniones en aplicativos de comunicación corporativa como “Teams”.

2.2.3 Procesamiento de lenguajes

El PLN o procesamiento de lenguaje natural, es el uso de las redes neuronales para asimilar la lingüística del ser humano, en donde la maquina adquiere información a partir de datos que se encuentran en textos. Este tipo de aplicación de las redes neuronales se suele utilizar en servicio al cliente mediante chatbots, análisis de información como correos electrónicos para la organización y categorización de categorías mediante el análisis de contenido de emails.

2.2.4 Funcionamiento de las redes neuronales

Una red neuronal funciona de forma semejante al cerebro humano, copiando el concepto de las señales eléctricas que circulan por el cerebro humano, estas células artificiales trabajan de forma similar con el fin de realizar procesos complejos con el fin de dar soluciones a problemas corporativos o de la vida humana misma, estas son, módulos generados por software de aprendizaje automático llamados nodos, estos nodos se encargan de procesar la información y generar respuestas mediante cálculos matemáticos como regresiones lineales entre otros. Las redes neuronales al ser un sistema software cuentan con una arquitectura propia que se divide en tres capas, las cuales son:

- Capa de entrada
- Capa Oculta
- Capa de salida

En la capa de entrada, se realiza el ingreso de la información a analizar dentro de las operaciones matemáticas creadas por la red neuronal, el medio de adquisición de estos datos son los nodos o dispositivos de campo que permiten realizar el ingreso de información al sistema, estos se encargan de analizar y clasificar la información para posteriormente ser enviados a la capa oculta, la cual se encarga de analizar los datos anteriores de forma más profunda con el fin de encontrar patrones que conlleven a la solución del problema y una vez analizada la información recibida por la capa de

entrada, se envía a la capa de salida, la cual proporciona la información final en el formato que se desea, puede ser binario, booleano, respuestas en enteros o respuestas en strings, esto depende de la forma de codificación del algoritmo implementado en la red neuronal.

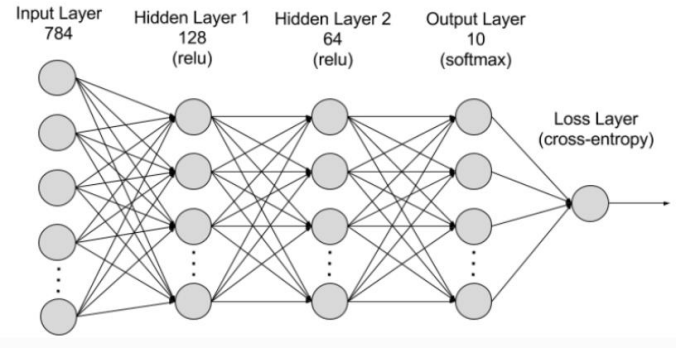

Figura 2 – Arquitectura de las redes neuronales

Tomado de: [6] "¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS". Amazon Web Services, Inc. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://aws.amazon.com/es/what-is/neural-network/

Por lo general las redes neuronales cuentan con más de una capa oculta y en cada una de estas se encuentran miles de neuronas artificiales que se conectan entre sí, por otro lado, las capas ocultas pueden obtener datos desde la capa de entrada o desde otras capas ocultas con el fin de procesar en mayor medida y depurar la mayor cantidad de información que no sea relevante para resolver el problema, por otro lado el peso de cada capa juega un papel importante en la jerarquía de la red neuronal, ya que una capa oculta más grande predomina sobre capas ocultas pequeñas generando así mayor influencia en cada uno de los nodos de la red neuronal. La Figura 2 – Arquitectura de las redes neuronales, muestra la topología de conexión utilizada en las redes neuronales.

2.2.4.1 Tipos de redes neuronales

Las redes neuronales se pueden catalogar de acuerdo con el flujo de información desde los dispositivos de entrada hasta los de salida, además las redes neuronales se pueden catalogar en tres tipos

- Redes neuronales prealimentadas
- Algoritmos de retro propagación
- Redes neuronales convolucionales

2.2.4.2 Redes neuronales prealimentadas

Este tipo de red neuronal se caracteriza por ser prealimentada de información antes de iniciar a captar datos nuevos de entrada, este tipo de aprendizaje de las redes neuronales se caracteriza además por procesar los datos obtenidos en una sola dirección, desde la capa de entrada hasta la capa de salida sin saltos, este tipo de red neuronal se enfoca principalmente en mejorar el rendimiento a largo tiempo.

2.2.4.3 Algoritmos de retro propagación

Las redes neuronales que utilizan el algoritmo de retroalimentación se basan en el uso continuo de bucles que mejoran el rendimiento de la red, por lo cual en cada iteración

se genera correcciones en el análisis de los datos y en los valores de salida de la red neuronal en busca de un solo camino correcto, una de las cualidades de este tipo de red es el uso de datos repetitivos mediante diferentes análisis afinando cada una de las salidas en los ensayos realizados. La red neuronal al utilizar iteraciones para retroalimentarse de información hace que cada nodo que pertenece a esta, realice el enrutamiento al siguiente nodo en la ruta de la red, al hacerlo se comprueba si es el nodo correcto y la información transmitida es la correcta, asignando un valor en peso para darles a cada uno de los nodos valores asignados, por lo cual entre más peso tenga el nodo, mayores iteraciones se realizarán por ese nodo, de este modo se descubre uno a uno los nodos pertenecientes a la ruta correcta.

2.2.4.4 Redes neuronales convolucionales

Las redes neuronales convolucionales se basan en la implementación de funciones matemáticas previamente programadas en el algoritmo, este tipo de redes neuronales se utilizan en mayor medida en la clasificación de imágenes al ser capaces de obtener información relevante de estas, logrando así el reconocimiento y categorización de las imágenes. En esta red neuronal se extrae información de cada capa oculta de la red, procesando características relevantes encontradas en cada una de las capas ocultas, como lo son el color, la figura, la forma, el tipo de bordeado, entre otros factores relevantes en el procesamiento de imágenes.

2.3 TINYML

Cuando se habla de TinyML, se hace referencia al rumbo en que los aprendizajes automáticos se dirigen, este tipo de aprendizajes hacen parte del ML (Machine learning), por lo cual la integración de Tiny, hace referencia a sistemas integrados de tamaño reducido.

TinyML, comprende toda la subrama de aprendizajes de machine Learning, los cuales tienen como objetivo principal detectar patrones en bases de datos y con base en ello generar rutas de predicción precisas, dando así solución a problemas cotidianos o corporativos, por otro lado TinyML, logra optimizar el procesamiento de datos a tamaños de kilobytes, esto sin depender de procesadores gráficos o de unidades de procesamiento, por otro lado TinyML, presenta beneficios en latencias, ancho de banda, seguridad y especialmente en el ahorro energético de los sistemas de aprendizaje automático.

2.3.1 Aplicaciones software para tinyml

Debido a que TinyML es un aplicativo de machine learning que se encuentra en exploración, la cantidad de dispositivos y aplicaciones que se encuentran en el mercado para realizar aplicaciones de redes neuronales mediante placas de desarrollo de bajos recursos para realizar proyectos TinyML, sin embargo de acuerdo con el avance tecnológico en el mundo del machine learning cada vez es mayor el interés por parte de las grandes industrias y el mundo académico para realizar aplicaciones de inteligencia artificial con dispositivos IoT, a esto se le conoce como tiny, la cual es una técnica de implementación de redes neuronales mediante aprendizaje automático en placas de desarrollo de bajos recursos, este tipo de placas de desarrollo a su vez son limitadas debido a que es un campo del machine learning aún en expansión, que supone una

desventaja, ya que el avance tecnológico no va a la par con las técnicas de aplicación de machine learning, a su vez esta problemática se ve reflejada en las aplicaciones software que se tienen destinadas para TinyML.

2.3.2 Latencia

Por lo general los datos que se transmiten de dispositivos a los servidores asociados a estos presentan una demora significativa debido a aspectos tanto de procesadores como de ancho de banda, potencia de transmisión entre otros factores, sin embargo, los aprendizajes

automáticos aplicados en TinyML realizan una optimización de estos tiempos, mediante predicciones inmediatas con un bajo coste informático, lo que produce una latencia significativamente menor en los retardos en la transmisión de datos.

2.3.3 Ancho de banda

La magnitud de datos que se envían desde dispositivos de campo hacia servidores asociados genera un uso de ancho de banda significativamente grande, afectando directamente al estado y calidad de la red del servicio, sin embargo, mediante TinyML, se realiza la independencia de la red de conexión, debido a que TinyML puede funcionar sin acceso a internet, esto directamente afecta al consumo de ancho de banda de la red, la cual presenta una disminución notable en la utilización de este.

2.3.4 Seguridad

Una de las grandes ventajas de TinyML, es que no debe enviar por obligación los datos a servidores para revisión de la información obtenida en los dispositivos de campo, sino que en su lugar, al utilizar dispositivos que apliquen aprendizajes automáticos mediante TinyML, se reduce el riesgo que representa el envío de información mediante un servicio de red, esto a su vez genera un ahorro económico en la inversión de servicios externos y/o dispositivos que maximicen la seguridad de un entorno de red completo.

2.3.5 Ahorro Energético

El ahorro energético que se presenta con la llegada de TinyML se debe a la capacidad de adaptación de los sistemas y dispositivos que implementan aprendizajes automáticos en procesadores de bajos recursos, lo cual se ve reflejado no solo económicamente en la compra de sistemas de grandes requisitos, sino en la potencia de funcionamiento de estos, ya que requieren un menor costo energético.

2.3.6 Usos de TinyML

El uso de TinyML tiene grandes ventajas en cuanto a eficiencia y capacidad de respuesta a problemáticas mediante ya lo hablado el aprendizaje automático, además de los beneficios presentados anteriormente TinyML, cuenta con el apoyo de dispositivos IoT (Internet de las cosas), los cuales permiten la desconexión de servidores para el procesamiento de datos y toma de decisiones, esto a su vez crea un panorama más extenso en donde TinyML de la mano de IoT puede ejecutarse de forma exitosa.

Algunos de los campos de aplicaciones de TinyML son:

- Industria textil
- Agricultura
- Medicina

En el caso de la industria textil TinyML, permite el monitoreo de la maquinaria utilizada en este tipo de corporaciones, adicional al monitoreo constante y en tiempo real, un aspecto clave que aporta TinyML a la industria textil es la predicción de fallos en los dispositivos mediante equipos IoT que se encuentren instalados dentro de los equipos. Por otro lado, en el sector agrícola TinyML utiliza sus beneficios en pro de determinar condiciones climatológicas, con el fin de evitar sequias o quema de vegetación debido a bajas temperaturas en las madrugadas, adicional a esto puede monitorear parámetros en tiempo real del estado de la tierra y de la vegetación, con esto se puede obtener estadísticas de predicción de enfermedades en los cultivos. Por último, en el campo de la medicina TinyML, permite trabajar con dispositivos que controlen y monitoreen el estado de salud de pacientes no solo dentro de las instalaciones hospitalarias sino también en pacientes que puedan requerir monitoreo de tiempo completo con dispositivos como por ejemplo válvulas aórticas o pulmonares.

TinyML se centra en el despliegue de sistemas que cuenten con aprendizaje automático mediante modelos de redes neuronales (Deep learning), en donde se puedan prever situaciones, mediante soluciones de bajo consumo energético, de memoria, ancho de banda y red, lo cual permite que se presten servicios innovadores sin procesamiento en servidores externos o servicios en nube, que a su vez implica una reducción considerable en mano de obra especialidad y costos en los proyectos que implementa la inteligencia artificial como base de soluciones corporativas. Uno de los grandes retos de TinyML es el optimizar los algoritmos de programación usados con frecuencia en machine learning, lo cual representa bajo coste informativo y económico en aplicaciones como las mencionadas con anterioridad.

2.4 PROPUESTA

En principio el trabajo de final de máster se centrará en dar conocimientos básicos sobre temas principales como qué es machine learning, que es TinyML y demás términos que sean necesarios para la comprensión de las redes neuronales y el aprendizaje automático que estas realizan para dar solución a proyectos relacionados con TinyML y que utilicen dispositivos IoT en el proceso, luego se enfocará en dar entendimiento a cada una de las plataformas software que se encuentren activas a la fecha que apliquen aprendizaje automático para TinyML. en este punto se realizará una guía de que se puede encontrar en cada aplicación, desde inicios de sesión hasta tipos de proyectos que pueden crearse en estas, dando comprensión del alcance de cada una de estas, tipos de microcontroladores disponibles para trabajar con cada una de ellas y así mismo realizar comparativas entre estas definiendo que ventajas y desventajas tiene cada una.

En la siguiente figura se pueden observar todas las plataformas y librerías que cuentan con la capacidad de ejecutar aplicativos de inteligencia artificial mediante TInyML y sus respectivas placas de desarrollo:

Software/ Library / Framework	Developer	Supporting Platform	Hardware Platforms	Targeted Applications
TensorFlow Lite (TFL)	Google Brain Team	Android, iOS, Embedded Linux, Micro-controllers	Arduino Nano 33 BLESense, Sparkfun Edge, STM32F746 Discovery Kit , Adafruit Edgebadge, Adafruit TensorFlow Lite for Microcontrollers Kit, Adafruit Circuit Playground Bluefruit, Espressif ESP32-Devkitc, Espressif ESP-EYE, Wio Terminal: ATSAMD51, Himax WE-I Plus EVB Endpoint AI Development Board, Synopsys Designware ARC EM Software Development Platform, Sony Spresense	Image and Audio Classification, Object Detection, Pose Estimation, Speech and Gesture Recognition, Segmentation, Video Classification, Text Classification, Reinforcement Learning, On Device Training, Optical Character Recognition
Utensor	ARM	Android, iOS, Embedded Linux, Micro-controllers	Mbed, ST K64 ARM Boards	Image Classification, Gesture Recognition, Acoustic Detection and Motion Analysis
Edge Impulse	Zach Shelby And Jan Jongboom	Android, iOS, Embedded Linux, Micro-controllers	Arduino Nano 33 BLE Sense, Arduino Nicla Sense ME, Arduino Nicla Vision, Arduino Portenta H7 + Vision Shield, Espressif ESP32, Himax WE-I Plus, Nordic Semi Nrf52840 DK, Nordic Semi Nrf5340 DK	Asset Tracking and Monitoring, Human Interfaces, Predictive Maintenance
Nanoedge AI Studio	Cartesiam	Android, Linux	STM32 Boards	Anomaly Detection, Predictive Maintenance, Condition Monitoring, Asset Tracking, People Counting, Activity Recognition
Pytorch Mobile	Meta AI (Facebook)	Android, iOS, Linux CPU	NNAPI (Android), Coreml (iOS), Metal GPU (iOS), Vulkan (Android)	Computer Vision and Natural Language Processing
Embedded Learning Library (ELL)	Microsoft	Windows, Ubuntu Linux, Mac OS X	Raspberry Pi, Arduino, Micro: Bit	Image And Audio Classification
STM32Cube.AI	STMicroelectronics	Android, Linux	STM32 ARM CORTEX Boards	Anomaly Detection, Predictive Maintenance, Condition Monitoring, Asset Tracking People Counting, Activity Recognition
Autoflow	Daniel Konegen And Marcus Rüb	Android, iOS, Embedded Linux, Micro-controllers,	MCU, FPGA Boards, Raspberry Pi	Image Classification, Object Detection, Pose Estimation, Speech Recognition, Gesture Recognition
Apache Mxnet	Apache Software Foundation (ASF)	Linux	Raspberry Pi, NVIDIA Jetson	Image Classification, Object Detection, Pose Estimation, Speech and Gesture Recognition
ML Kit for Firebase	Google	Android, iOS	Mobile Devices	Facial Detection, Bar-Code Scanning, Object Detection

Figura 3 – Aplicativos Software para TinyML

Tomado de: Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., & López, O. L. A. (2023). TinyML: Tools, applications, challenges, and future research directions. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-16740-9

En primer lugar la Figura 3 – Aplicativos Software para TinyML, muestra algunos aplicativos que no son de acceso gratuito como lo es NanoEdge IA Studio, este aplicativo software es desarrollado por Cartesiam, el cual también cuenta con limitaciones de sistemas operativos de ejecución, junto con las tarjetas de desarrollo que acepta, por lo cual expansión es limitada debido a su compatibilidad con diferentes compañías en la industria de la inteligencia artificial, sin embargo deberá ser entendido que sus aplicaciones de redes neuronales son en su mayoría predicción de movimiento, detección de objetos y seguimiento de personas, por lo cual va enfocada a soluciones con dispositivos como cámaras de videovigilancia. Por otro lado existen bibliotecas como Utensor, AutoFlow, ELL, que funcionan en una amplia variedad de placas de desarrollo como FPGA, raspberry PI, ARM cortex y Arduino, entre otras, que funcionan para pocas aplicaciones de redes neuronales, como el reconocimiento de imágenes o sonido y clasificación de estos, Por otra parte, existen plataformas de carácter gratuito para desarrollares amateur, como Pytorch, TensorFlow y Edge Impulse, que a su vez

tiene una amplia gama de bloques de aprendizaje predeterminados para iniciar en el mundo de la inteligencia artificial mediante la técnica de TinyML, estos últimos tienen el mayor porcentaje de programadores de redes neuronales del mundo de TinyML, con un 59% de proyectos realizados mediante TensorFlow, 17% por Pytouch y 24% utiliza Edge impulse y los otros aplicativos software para TinyML, observados en la Figura 3 – Aplicativos Software para TinyML.

Por otro lado, las aplicaciones que los programadores de redes neuronales que más se utilizan en la actualidad son aplicaciones de visualización de imágenes, tanto reconocimiento como clasificación de esta, detección de movimiento en tiempo real y seguimiento de objetos y clasificación y reconocimiento de audio, a su vez las placas de desarrollo más utilizadas por los programadores de proyectos TinyML son Raspberry Pi pico, Arduino Nano 33 BLE sense, ESP32 y Nvdia Jetson Nano, las cuales tienen características de bajos recursos en términos de consumo, procesamiento, memoria, entre otras cualidades que son características ideales para despliegue de técnicas de aprendizaje automático en TinyML.

Luego de realizar esta comprensión de los aplicativos se realizará un desarrollo de aplicativos con hardware, en donde se dará elección a un dispositivo que cumpla con los requisitos mínimos para ser utilizado en las plataformas de software TinyML seleccionadas, esto se realizará con el fin de obtener valores resultados de uso comprendiendo las limitantes de cada una de las plataformas, por último se realizará una serie de conclusiones y parámetros de mejora con los que se pueda retomar este trabajo de final de máster en proyectos de TinyML con mayor complejidad partiendo de la documentación entregada.

3 DESARROLLO

En esta sección del documento de comparativas se realizó un análisis de cada una de las plataformas software para TinyML disponibles para el desarrollo de aplicaciones que aprendizaje automático, así mismo se realizaron pruebas de hardware con una placa de desarrollo de bajos recursos para completar el análisis de estos aplicativos, por lo cual esta sección está enfocada en el entendimiento y descripciones de cada una de las cualidades con las que cuentan las plataformas software que se documentan a continuación:

3.1 EDGE IMPULSE

Edge impulse es una plataforma que tiene como objetivo el desarrollo de algoritmos de aprendizaje automático para aplicación de Tiny Machine learning, mediante sistemas embebidos, placas de desarrollo y microcontroladores de bajos recursos. Esta plataforma de software cuenta con herramientas adecuadas para realizar proyectos de redes neuronales de la mano de la inteligencia artificial, estos pueden ser elaborados desde principiantes en el mundo del machine learning , como profesionales avanzados en la ejecución y puesto en marcha de redes neuronales, ya que Edge Impulse permite ingresar bases de datos, parametrizar variables y entrenar la red neuronal a partir de estos datos sin preocuparse de la codificación de complejos algoritmos de computación.

Figura 4 – Edge Impulse

Tomado de: “Edge Impulse”. Edge Impulse. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Con Edge impulse se puede realizar proyectos en poco tiempo, es una interfaz software de TinyML, intuitiva en donde se evidencia que se pueden crear sistemas de detección de objetos, reconocimiento de voz o procesamiento de gestos, este último se realiza mediante sensores como giroscopios o acelerómetros, estos sistemas solo requieren de una base de datos para realizar el entrenamiento del sistema TinyML, una vez se tenga esta información es momento de crear un impulse, la cual es la herramienta denominada por Edge Impulse para la creación y el entrenamiento del modelo de aprendizaje automático TinyML.

Así mismo Edge Impulse al ser una plataforma pionera en la implementación de inteligencia artificial mediante TinyML, está respaldada por grandes fabricantes de chips electrónicos, sensores, actuadores y microcontroladores, entre los más famosos está Arduino, ST electronics, Microchip y ARM.

Figura 5 – Marcas asociadas a Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Debido a que Edge Impulse cuenta con la confianza de grandes fabricantes como se puede observar en la Figura 5 – Marcas asociadas a Edge Impulse, se puede implementar modelos de aprendizaje automático en una amplia gama de microcontroladores como lo es **Raspberry Pi, Arduino 33 BLE, Cortex u OpenMV,** los cuales pueden ser utilizados como placas de desarrollo de bajos recursos, adicional el lenguaje de programación de estas tarjetas puede ser C++ o Java en algunos casos, los cuales son de conocimiento común en los grupos de trabajo de principiantes y profesionales de diseño de inteligencia artificial mediante machine learning.

3.1.1 Analisis Edge Impulse

En la página de Edge impulse se puede conseguir el listado de las placas de desarrollo que son compatibles con esta plataforma software de TinyML, estas cuentan con un firmware que permite la recopilar información de cada uno de los periféricos con los que cuentan las placas, Edge impulse es una de las plataformas que guía a los creadores de redes neuronales de inicio a fin, siendo una buena opción para el inicio en el mundo del machine learning. A continuación, el listado de placas de desarrollo que admite Edge Impulse para creación de redes neuronales:

ITEM	PLACA DE DESARROLLO
1	Alif Esemble E7
2	Arduino Nano 33 BLE Sense
3	Arduino Nicla Sense ME
4	Arduino Nicla Vision
5	Arduino Nicla Voice
6	Arduino Portenta H7 + Vision Shield
7	Avnet RASynBoard (Renesas RA6 and Syntiant NDP 120)
8	Expressif ESP32
9	Himax WE-I Plus
10	Infineon CY8CKIT-062-BLE Pioneer Kit
11	Infineon PSoc 62S2 Wi-Fi BT Pioneer Kit

ITEM	PLACA DE DESARROLLO
12	Nordic Semi nRF52840 DK
13	Nordic Semi nRF5340 DK
14	Nordic Semi nRF9160 DK
15	Nordic Semi Thingy:53
16	OpenMV Cam H7 Plus
17	Particle Photon 2
18	Renesas CK-RA6M5 Cloud Kit
19	Seeed Grove Vision AI Module
20	Silicon Labs xG24 Dev Kit
21	Silicon Labs Thunderboard Sense 2
22	Sony's Spresense
23	ST B-L475E-IOT01A
24	Synaptics katana EVK
25	Syntiant TinyML Board
26	TI CC1352P Launchpad
27	Raspberry Pi RP2040

Tabla 1 - Dispositivos compatibles con Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 5 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

De acuerdo con la Tabla 1 - Dispositivos compatibles con Edge Impulse, el número de placas compatibles con las herramientas de diseño de modelos de redes neuronales pertenecientes a Edge Impulse es de 27 placas de desarrollo, las cuales todas cuentan con distintos sensores instalados en la misma placa, como acelerómetros, colorímetros, cámaras en algunos casos y micrófono.

Así mismo Edge impulse cuenta con la practicidad de conexión de teléfonos inteligentes a la plataforma de desarrollo software, esta acepta la conectividad de todos los sistemas operativos Como IOS y Android, además cabe aclarar que el dispositivo móvil funciona como "Placa de desarrollo" al utilizar los periféricos con los que los móviles actuales cuentan, como cámara, micrófono y acelerómetro. Para realizar esta asociación la interfaz de Edge Impulse es intuitiva por lo cual realizar el emparejamiento no deberá de tomar mayor esfuerzo.

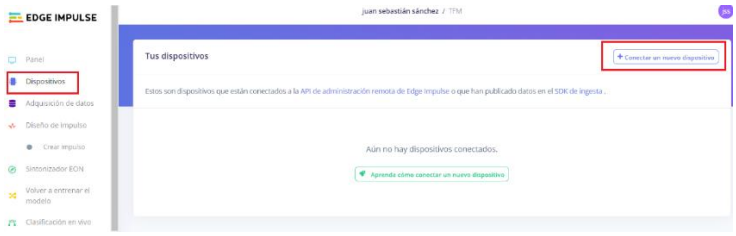

Figura 6 – Emparejamiento Dispositivos móviles – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 5 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

La conexión de dispositivos tanto telefónicos como las placas de desarrollo mencionadas con anterioridad se realizan mediante esta sección llamada *Dispositivos,* la plataforma tiene como ventaja que a simple vista es intuitiva y guía al usuario a las secciones importantes sin necesidad de ser un usuario avanzado de creación de redes neuronales de TinyML, por lo que es una aplicación ideal para iniciar el camino del machine learning.

Por otro lado Edge Impulse tiene la capacidad de recolectar datos desde múltiples dispositivos, estos pueden teléfonos celulares, CPU'S o placas de desarrollo, debido a que cuenta con una API que se utiliza para enviar información de un dispositivo asociado nuevo a Edge Impulse, este servicio está disponible en HTTP y HTTPS, sin embargo se requiere clave de autentificación para acceso a este servicio de Edge Impulse, adicional a esto la API de ingestión soporta los siguientes formatos de información:

- JPG, PNG para imágenes
- MP4, AVI para archivos de video
- WAV para archivos de audio
- CBOR/JSON formato de archivo que maneja Edge Impulse para la adquisición de datos
- CSV para bases de datos

Estos archivos una vez se suben al servicio de Ingesta de Edge Impulse, aparecerán automáticamente en el dashboard del proyecto de entrenamiento automático

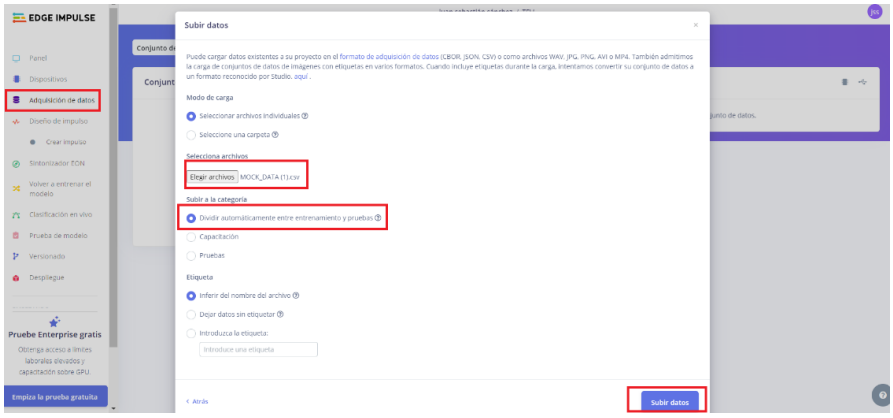

Figura 7 – Adquisición de datos – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 5 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

En la Figura 7 – Adquisición de datos – Edge Impulse, se observa nuevamente que la plataforma Edge Impulse está diseñada para guiar a los usuarios en la dirección correcta sin problemas de configuración de los aprendizajes automáticos, esto se convierte en una ventaja sobre otros aplicativos debido a la baja complejidad de los servicios que presta Edge Impulse, además de la gran variedad de tipos de archivos que la plataforma acepta, lo que repercute en la facilidad de creación de redes neuronales de diferentes tipos con una amplia gama de aplicaciones posibles.

Edge impulse cuenta también con herramientas de aprendizaje adaptables para plataformas como **Arduino IDE**, cuenta con kit de herramientas de **NVIDIA TAO**, **Omniverso de NVIDIA**, **Weights & Biases** y **Scailable**. En primer lugar Arduino y Edge Impulse cuentan con una asociación estratégica para que las herramientas de aprendizaje automático estén no solo disponibles dentro de Edge Impulse, sino también en la denominada Arduino Machine Learning Library, la cual se encuentra dentro de la aplicación de Arduino IDE, lo cual es una gran ventaja para programadores que conozcan lenguajes de programación como C/C++, por otro lado el kit de herramientas de NVIDIA TAO, se basa en el aprendizaje automático por transferencia, lo cual permite simplificar los tiempos de entrenamiento de modelos y mejora el rendimiento de los mismos, por lo cual genera un flujo de trabajo ideal para modelos con un entrenamiento previo que sean enfocados hacia situaciones reales sin la necesidad de alimentar al modelo con bases de datos grandes, por otro lado el Omnniverso de NVIDIA, es una plataforma de desarrollo de tiempo real multi GPU, la cual permite su escalabilidad y opera en plataformas de metaverso teniendo como base la tecnología de NVIDIA RTX, esta extensión se encuentra también disponible en Edge Impulse, lo que indica que la posibilidad de Edge Impulse para adaptar modelos sencillos y complejos mediante la asociación de las herramientas de modelo de inteligencia artificial mencionadas con anterioridad, se convierte en tareas triviales sin necesidad de contar con amplios conocimientos sobre el machine learning ni algoritmos de aprendizaje automáticos. Por ultimo Edge Impulse cuenta con dos asociaciones más las cuales son Weight & Biases y Scailable, en primer lugar Weight & Biases facilita el monitoreo de los modelos de aprendizaje automático, mejora la experiencia de trabajo en simultaneo de forma online y ayuda a la gestión de los datos, etiquetando y versionando cada uno de estos, por otro lado Scailable, permite realizar modelos de aprendizaje escalables con la posibilidad de conexión de cualquier dispositivo compatible, Scailable se emplea en su mayor parte en modelos que requieran tareas de visión de imágenes.

Así mismo Edge Impulse ofrece en su repertorio de proyectos, una serie de bloques de aprendizaje para los modelos de TinyML que se deseen crear, estos bloques se encuentran predefinidos en la plataforma y son los siguientes:

- Clasificación (Keras)
- Regresión (Keras)
- Detección de anomalías (K-means)
- Detección de anomalías (GMM)
- Clasificación de imágenes (Usando aprendizaje de transferencia)
- Detección de palabras clave (Usando aprendizaje de transferencia)
- Detección de objetos (Usando MobileNetV2 SSD FPN)

- Detección de objetos (Usando FOMO)

3.1.1.1 Clasificación (Keras)

La idea de una clasificación en una red neuronal es tomar datos de ingreso al sistema y generar una probabilidad sobre la clasificación de esta información, por lo cual se asignara a una clase de datos en particular, como ya se mencionó anteriormente en este documento, las redes neuronales cuentan con capas de entrada, capas ocultas y capas de salida, por lo tanto en función de la información analizada en las capas de entrada y ocultas, se generará un resultado, este es un proceso iterativo con el fin de que la red neuronal defina parámetros correctos para cada uno de los datos ingresados al modelo y con esto se realice una clasificación de los mismos de forma exitosa.

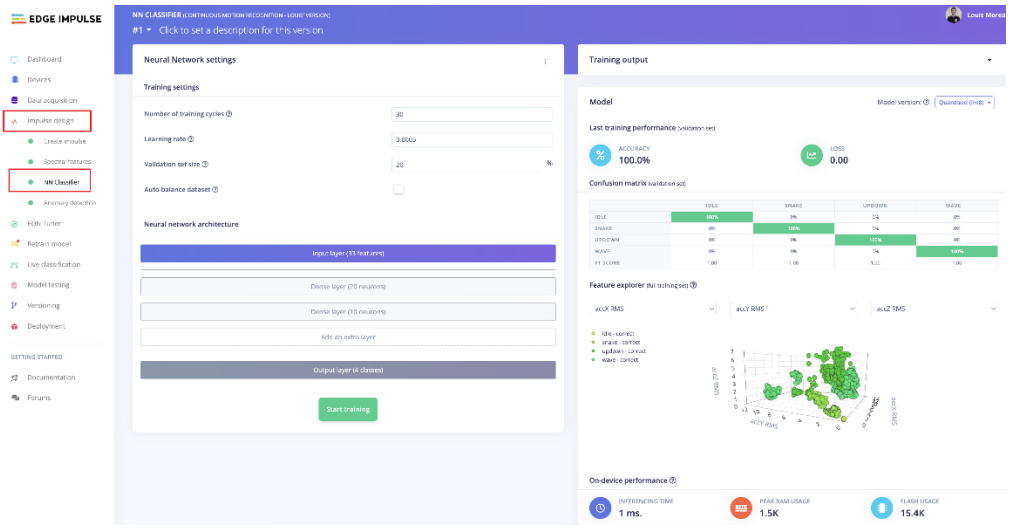

Figura 8 – Creación de red neuronal por clasificación Keras – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 5 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

En la página principal de Edge Impulse, en la sección *Impulse Design,* Se puede realizar el ajuste de la red neuronal que se desea crear, para este caso se deberá elegir *NN Classifier,* lo cual hace referencia al tipo de red neuronal por clasificación (Keras), la plataforma, así mismo se deberán de seguir los siguientes pasos para la configuración de la red neuronal dentro del apartado *Impulse Design*:

- **Número de ciclos de entrenamiento:** Este número corresponde a la cantidad de veces que la red neuronal procesa los datos de entrada.
- **Tasa de aprendizaje:** Este apartado se debe configurar con el fin de controlar cuanto se debe actualizar cada uno de los parámetros internos del modelo TinyML, esto se define básicamente en la velocidad con la que la red neuronal aprenderá sobre los datos ingresados.
- **Tamaño del conjunto de validación:** Hace referencia al porcentaje de información apartada del entrenamiento, este valor suele estar en 20% (valor estándar en la plataforma).
- **Conjunto de datos de equilibrio automático:** Ayuda a que el modelo sea robusto en caso de bases de datos con poca información.

La plataforma Edge Impulse determina valores predeterminados de entrenamiento para cada red neuronal, sin embargo, se desea realizar ajustes, en los títulos de los parámetros cuenta con burbujas de información lo cual crea un espacio de aprendizaje si el creador de la red neuronal no cuenta con la experticia suficiente para realizar estos ajustes con base en la experiencia.

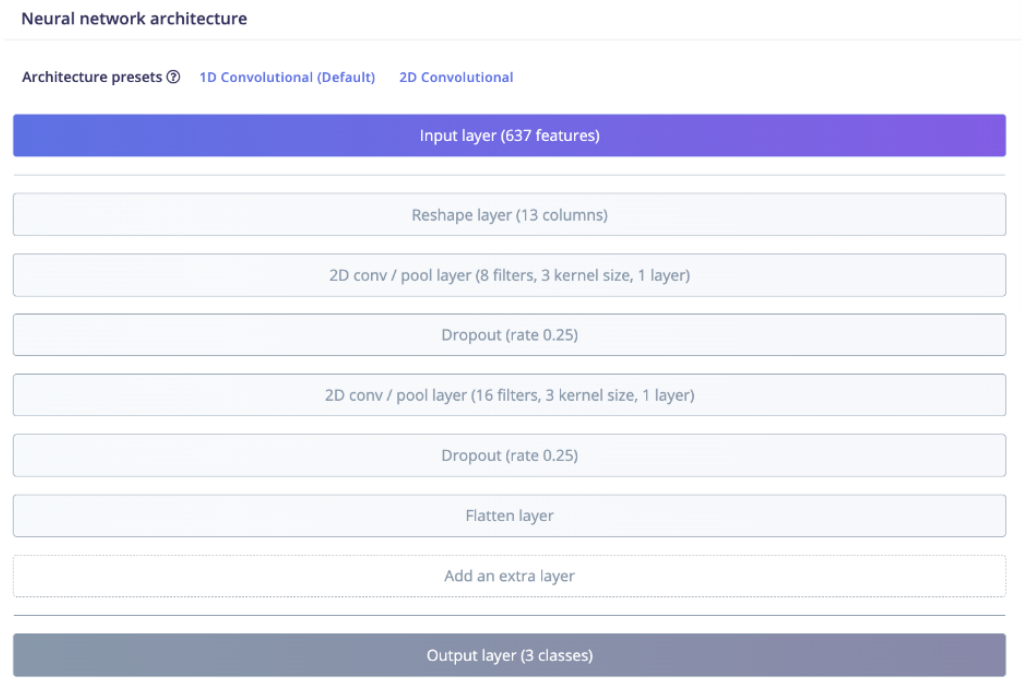

Figura 9 – Arquitectura de red neuronal – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Con base en lo anteriormente mencionado en este documento, Edge Impulse ofrece una configuración de arquitectura de red neuronal, estos parámetros deberán variar con base en dos características, el tipo de proyecto a realizar y el conocimiento de la configuración de redes neuronales del usuario, por lo que sí es un proyecto creado por principiantes Edge Impulse ofrece una variedad de configuraciones de la arquitectura de la red neuronal por clasificación. Esta toma como entrada la base de datos importada para realizar el entrenamiento y realiza el paso a paso de cada una de las capas que conforman a la red neuronal, para el caso de clasificación se tienen las capas de entrada, capa oculta y capa **Softmax**, la cual se encarga de clasificar cada uno de los datos y obtener una probabilidad de clasificación de datos.

Así mismo la plataforma ofrece un *Modo visual,* en donde se permite agregar capas a la red neuronal y generar mayor robustez en el modelo creado, las cuales analizarán repetitivamente los datos de entrada y salida de cada una de las capas ocultas, lo cual generará un patrón más exacto y un dato de salida en la capa de salida con mayor probabilidad de acierto.

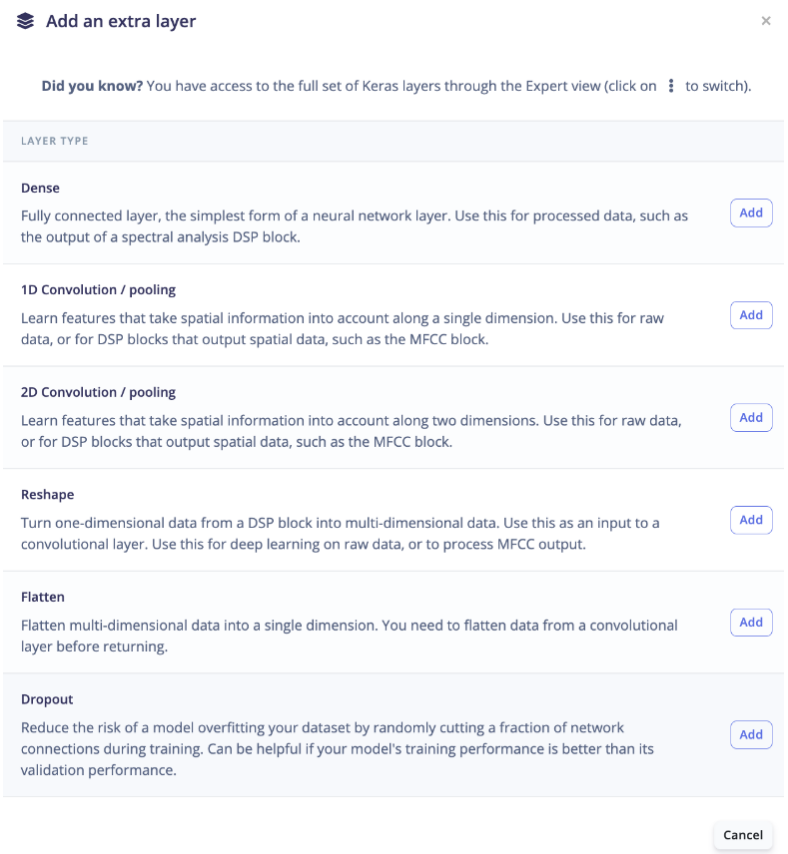

Figura 10 – Adición de capas extras en red neuronal – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Como se observa en la Figura 10 – Adición de capas extras en red neuronal – Edge Impulse, la plataforma de software para TinyML, ofrece también un apartado para agregar capas extras en los modelos de redes neuronales, esto es de suma utilidad para usuario con mayor experiencia en el uso del machine learning.

Así mismo Edge Impulse ofrece un modo experto en donde se emplea la librería Keras, la cual es una API de alto nivel, en donde se permite la definición, ajuste y entrenamiento de modelos de aprendizaje automático, sin necesidad de realizar ajustes de algoritmos de bajo nivel.

Figura 11 – Cambio de modalidad para expertos – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Este cambio de modalidad se realiza mediante la pestaña de *configuraciones de red neuronal,* así como se puede observar en la Figura 11 – Cambio de modalidad para expertos – Edge Impulse.

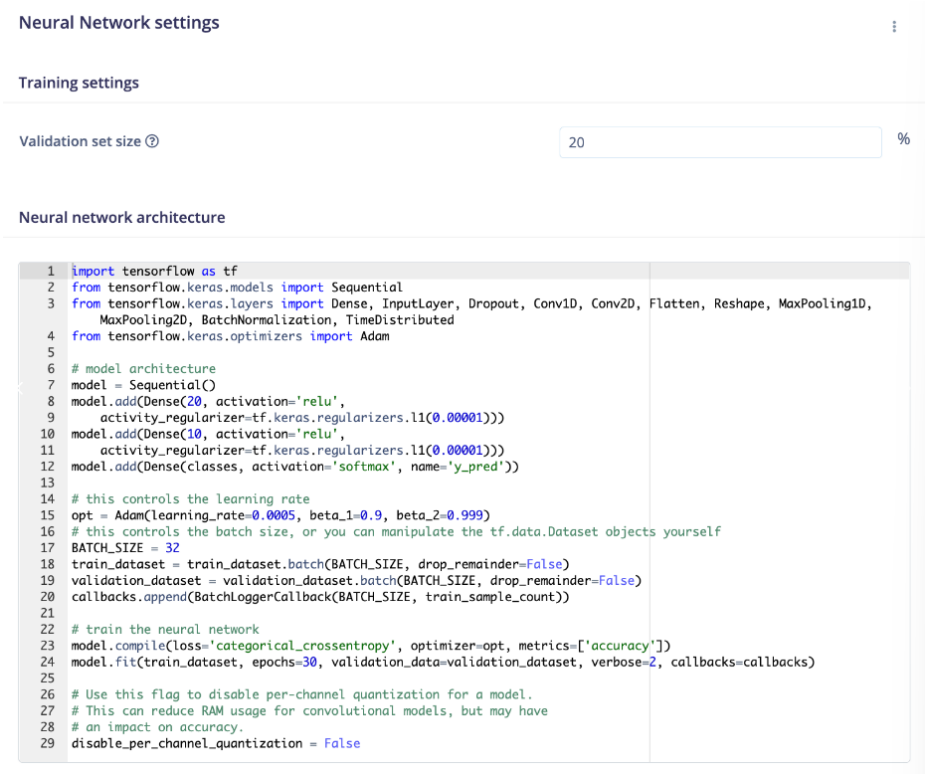

Figura 12 – Interfaz expertos Redes neuronales – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

De esta forma al realizar el cambio de modalidad de la red neuronal, se verá reflejado el código de alto nivel, en donde se puede modificar todos los parámetros de la red neuronal mediante algoritmos escritos en Python, tal y como se puede observar en la Figura 12 – Interfaz expertos Redes neuronales – Edge Impulse.

Una vez realizado los pasos anteriores en la configuración y entrenamiento del modelo de la red neuronal, se encuentra la sección de *Resultados de la formación,* este muestra los valores obtenidos en la salida del sistema, adicional a esto en la pestaña de *trabajos*¸ se pueden encontrar los datos obtenidos en todas las iteraciones en el tiempo de entrenamiento de la red neuronal, cabe mencionar que esta función solo está disponible en la versión de pago de Edge Impulse, la cual se enfoca en usuario corporativos y no en programadores independientes.

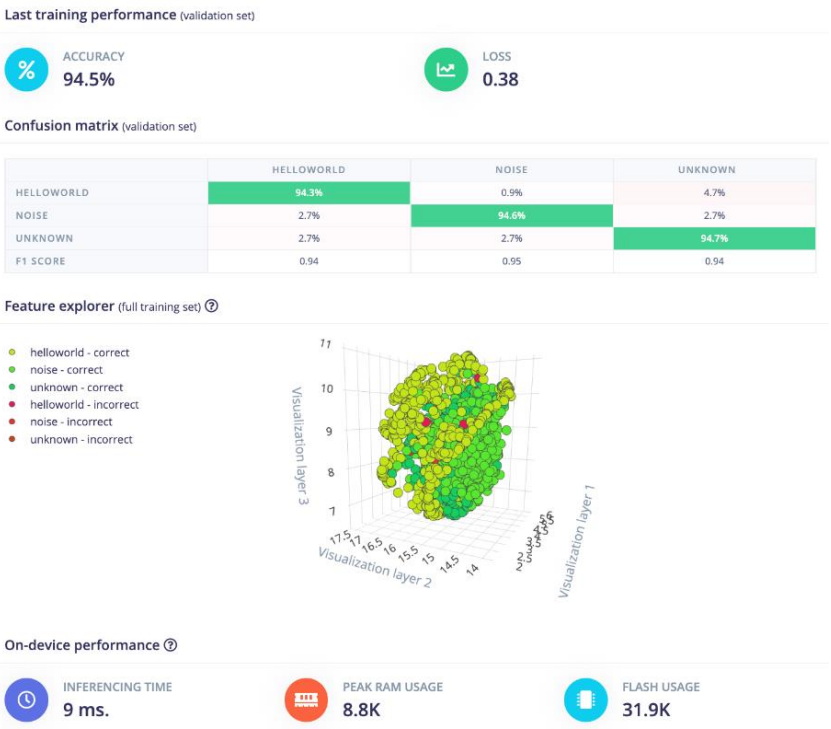

Figura 13 – Interfaz de validación de datos – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Por último, al realizar el entrenamiento del modelo de red neuronal hay una sección en donde se puede visualizar la información general del rendimiento del modelo, con el fin de evaluar los parámetros y validarlos por parte del creador del modelo. La forma en que se visualizan los datos es numérica/grafica como se puede observar en la Figura 13 – Interfaz de validación de datos – Edge Impulse.

3.1.1.2 Regresión (Keras)

Otro bloque de aprendizaje predeterminado por Edge Impulse es la red neuronal por regresión Keras, este tipo de aprendizaje automático es uno de los más comunes en modelos de redes neuronales, empleado el aprendizaje por supervisión de machine learning. En este tipo de modelos el entrenamiento se basa en la comprensión de variables independientes, la relación que estas tienen entre sí y el resultado que es una variable dependiente del sistema. Posteriormente el modelo tendrá la factibilidad de predecir resultados, con datos ingresados anteriormente, nuevos o datos ocultos.

Para que la red neuronal por regresión sea factible debe tener siempre una serie de parámetros previos que ayuden al etiquetado de la información ingresada al modelo de TinyML, por lo cual los siguientes parámetros deberán configurarse con antelación para que el aprendizaje automático por regresión sea efectivo:

3.1.1.3 Etiquetado

En primer lugar, el etiquetado realiza seguimiento de la información ingresada al sistema mediante valores numéricos, en donde cada dato tenga un peso determinado, dando mayor relevancia a la información de mayor peso.

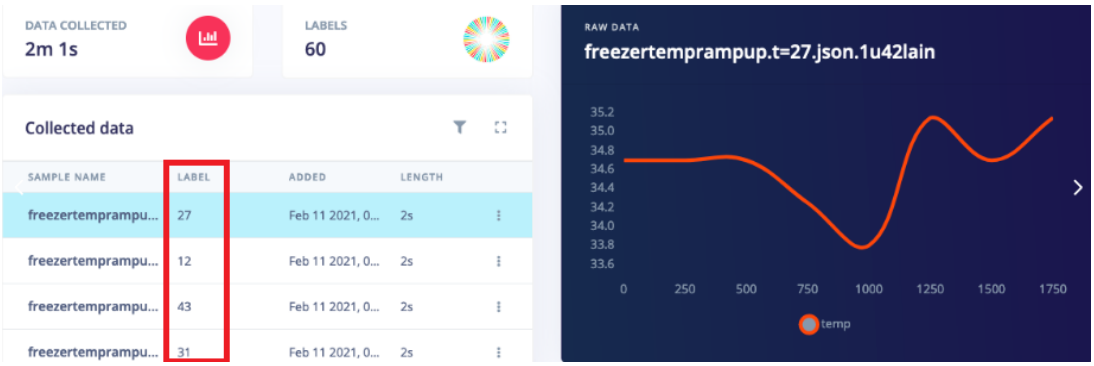

Figura 14 – Etiquetado red neuronal por regresión Keras – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Tal y como lo muestra la Figura 14 – Etiquetado red neuronal por regresión Keras – Edge Impulse, El modelo de red neuronal recolecta datos y en la casilla Label, le asigna un valor de etiquetado o peso a cada uno de los datos de entrada analizados.

3.1.1.4 Bloques de procesamiento

En el modelo de regresión Edge Impulse pone a disposición todos los bloques de procesamiento disponibles en la página para realizar la integración de señales a la base de datos entregada al modelo, pueden ser datos de audio, vibración, espectrogramas, o imágenes, entre otros tipos de formatos de información disponibles en la página.

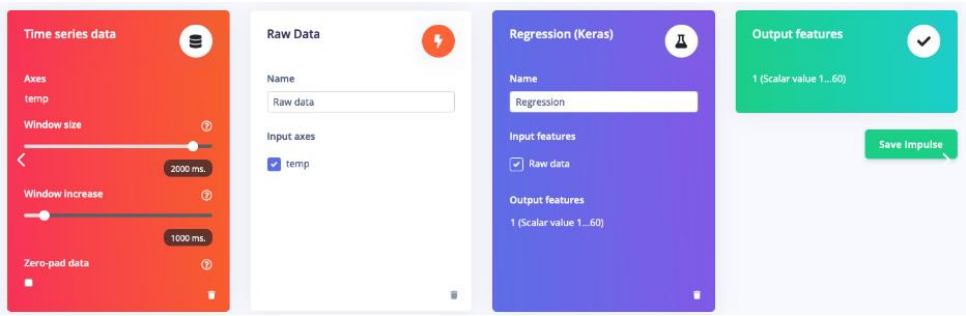

Figura 15 –Bloques de procesamiento en regresión Keras – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Como se puede observar en la Figura 15 –Bloques de procesamiento en regresión Keras – Edge Impulse, cuenta con la facilidad de optar por uno o más bloques de procesamiento dependiendo del tipo de datos que se requieran analizar, esto abre la posibilidad de extraer datos de todo tipo de sensores.

Con el modelo de regresión Keras, Edge Impulse invita que el programador de la red neuronal tenga la libertad de modificar la arquitectura de esta de forma visual o mediante código de alto nivel Keras, el cual como se mencionó para el modelo de red neuronal por clasificación se puede activar mediante la sección de configuración de la red neuronal. Por otro lado, este modelo de aprendizaje automático también cuenta con configuraciones predeterminadas por el equipo de Edge Impulso, tal como se tiene para

otros tipos de aprendizajes automáticos disponibles en la plataforma de software para TinyML.

Por último, el modelo de regresión se puede visualizar en la pestaña de *Prueba de modelo,* disponible en el dashboard del proyecto creado.

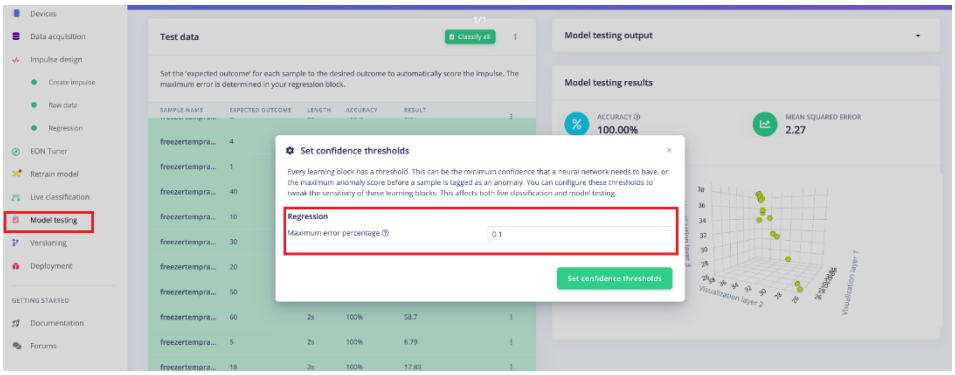

Figura 16 – Resultados de modelo de regresión Keras – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Como se observa en la Figura 16 – Resultados de modelo de regresión Keras – Edge Impulse, en la pestaña de *Model Testing,* Se puede realizar el ajuste del error máximo de los valores obtenidos mediante el aprendizaje por regresión Keras, este error se puede minimizar hasta 0.1, lo cual infiere en que es un modelo de red neuronal preciso.

3.1.1.5 Detección de anomalías (K-means)

Las redes neuronales cuentan con defectos con respecto al manejo de datos que no han procesado con anterioridad, esto se debe a que la red neuronal solo puede procesar patrones de información analizada con anterioridad en los entrenamientos realizados, por lo cual el modelo de aprendizaje para detección de anomalías es un método que analiza la información de un conjunto de datos y los agrupo mediante similitudes encontradas en K número de grupos predefinidos en el modelo de red neuronal. Mediante este sistema se puede configurar un umbral determinado con el fin de detectar anomalías en los datos.

Para realizar la configuración de los bloques de aprendizaje de detección de anomalías, Edge Impulso cuenta con los siguientes ajustes para definir umbrales que requieren los conjuntos de datos:

- Recuento de conglomerados: Hace referencia al recuento de K conjuntos de datos.
- Ejes: Hace referencia a los ejes que corresponden a las características de los datos entregados al modelo de aprendizaje.

Para realizar el proceso de aprendizaje de este tipo de modelo de red neuronal solo se debe ir al apartado de *Impulse Design,* y pulsar iniciar formación, cuando se desee realizar un análisis de los datos obtenidos en este mismo se verán los datos reflejados de forma gráfica.

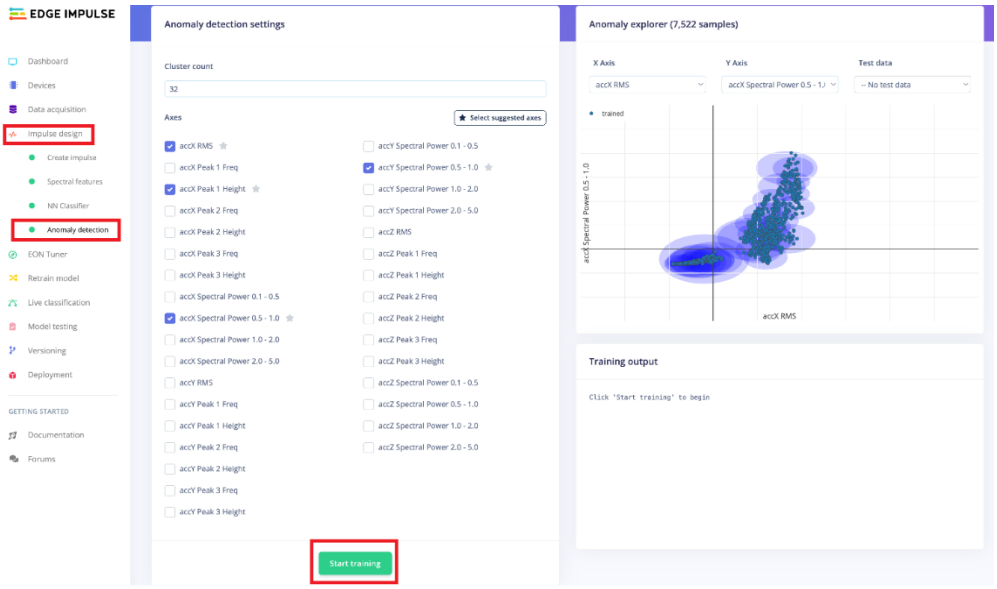

Figura 17 – Resultados de modelo de detección de anomalías (K-means) – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Una vez realizado el entrenamiento deberá de aparecer una distribución de puntos en medio de los ejes configurados con antelación, de acuerdo con esto, debería de aparecer datos en la gráfica de color naranja los cuales se asocian con anomalías encontradas por el modelo de la red neuronal.

3.1.1.6 Detección de anomalías (GMM)

El modelo de red neuronal GMM (Modelo de mezcla gaussiana), representa la distribución de datos del sistema como una mezcla de distribuciones gaussianas, Cada una de las mezclas de distribución gaussiana representa un segmento de datos los cuales son agrupados por el modelo debido a que cuentan con características semejantes. De acuerdo con la información anterior, se puede entender que el modelo de mezclas gaussianas funciona prediciendo que las muestras que se encuentran dentro de un conjunto de datos pueden ser modeladas utilizando las distribuciones gaussianas. Esto implica que detectar una anomalía mediante este modelo requiere de la identificación de puntos de datos que cuentan con una probabilidad baja de tener características similares a otros elementos dentro del mismo conjunto de datos.

El modelo de red neuronal de detección de anomalías GMM, cuenta con dos parámetros ajustables los cuales Edge Impulse pone a disposición del programador:

- Número de componentes: Este parámetro hace referencia al número de grupos de datos con los que cuentan los modelos GMM.
- Ejes: Al igual que el modelo de detección de anomalías K-means, los ejes representan las características con las que se definen los datos ingresados al modelo.

Figura 18 – Resultados de modelo de detección de anomalías (GMM) – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Al igual que en los resultados que se pueden obtener por el modelo de detección de anomalías K-means, el modelo de mezclas gaussianas, al terminar el aprendizaje automático, generará una gráfica en los ejes preestablecidos con anterioridad, en donde arrojará una serie de datos de colores azules, los cuales están dentro de los grupos de mezclas gaussianas, por otro lado los puntos que se encuentran fuera de estos conjuntos son etiquetados como anomalías de la información suministrada en el modelo y se denotarán de color naranja dentro de la gráfica obtenida. Tal y como se puede observar en la Figura 18 – Resultados de modelo de detección de anomalías (GMM) – Edge Impulse

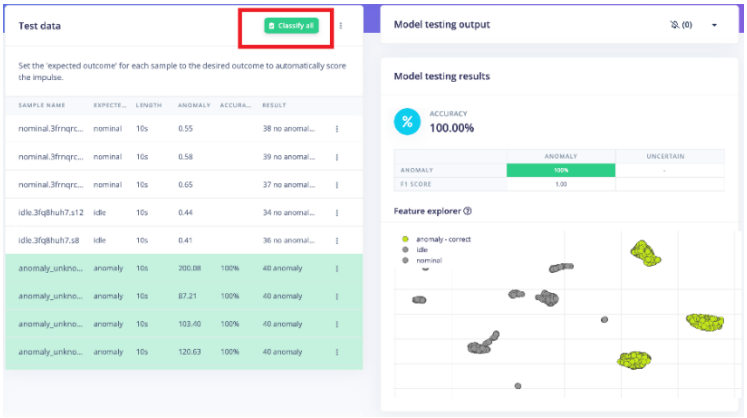

Figura 19 –Configuración de umbral de confianza (GMM) – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Para que el modelo quede bien ajustado se deberá realizar una última configuración, Edge Impulse cuenta con un valor predeterminado para configurar los umbrales de confianza del modelo GMM, este valor se encuentra en 1.00 de forma predeterminada, lo que debería bastar para que los datos que, si pertenecen al conjunto de datos GMM, se configuren como válidos y no como una anomalía, como se observa en la Figura 19 – Configuración de umbral de confianza (GMM) – Edge Impulse.

3.1.1.7 Clasificación de imágenes (Usando aprendizaje de transferencia)

Las redes neuronales basadas en el aprendizaje automático de transferencia, puede ser utilizadas para realizar clasificación de imágenes, siempre y cuando el conjunto de datos sea pequeño. El aprendizaje de transferencia se caracteriza por aprovechar datos aprendidos de un problema y extrapolarlos a problemas que se relacionen con los datos previos. Para este tipo de aprendizaje Edge Impulse creó un apartado en donde se puede elegir redes neuronales previamente entrenadas, para correlacionarlas a entrenamientos nuevos. Para este tipo de aprendizajes Edge Impulse utiliza arquitecturas de redes neuronales propias de MobileNetV1 o MobileNetV2, las cuales utilizan datos previamente entrenados como una red ImageNet.

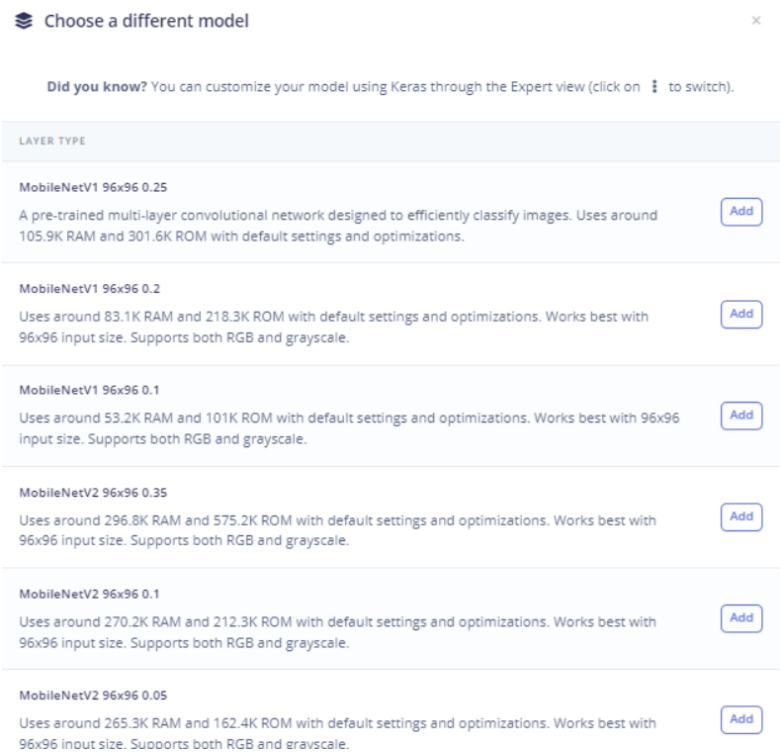

Figura 20 –Configuración de modelos MobileNet, Aprendizaje por transferencia – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Este tipo de aprendizaje viene con bloques de datos de entrada que van desde 96 x 96 pixeles a 320 x 320 pixeles, imágenes RGB o imágenes en escala de grises. El tipo de configuración y variedad de modelos se puede observar en la Figura 20 –Configuración de modelos MobileNet, Aprendizaje por transferencia – Edge Impulse.

3.1.1.8 Detección de palabras clave (Usando aprendizaje de transferencia)

El aprendizaje de transferencia también puede funcionar para la detección de palabras clave, este proceso se realiza mediante el análisis del audio como información de entrada al modelo de la red neuronal, esta utiliza las mismas características que se tienen en el aprendizaje de transferencia para el análisis de imágenes, por otro lado, la detección de palabras clave mediante este modelo es más eficiente que la aplicación de detección de imágenes, incluso si la información ingresada al modelo es pequeña.

Para esta aplicación Edge Impulse proporciona una serie de parámetros predeterminados, sin embargo, se deberá habilitar la función de equilibrio automático, lo cual evita el sesgo de los datos ingresados al modelo, por último, de ser necesario se deberá habilitar la opción de aumento de datos para que el conjunto de datos ingresados sea más grande, lo cual le da robustez al sistema TinyML de detección de palabras clave.

3.1.1.9 Detección de objetos (Usando MobileNetV2 SSD FPN)

El algoritmo de detección de objetos es uno de los modelos de aprendizaje más complejos de elaborar, debido a que construir un modelo desde cero sin tener una base de datos de entrada con una gran cantidad de información, no se asegura que el modelo generalice bien los datos y el entrenamiento puede llevar días en culminar los procesos de aprendizaje, por lo cual Edge Impulse optó por utilizar el aprendizaje de transferencia, lo cual aprovecha todas las características del modelo, realizando un re-entrenamiento en las capas superiores de la red neuronal, generando así modelos robustos y eficientes, por otro lado se deberá configurar la sección de *Impulse Design,* para realizar detección de objetos y utilizar el modelo MobileNetV2 SDD FPN-Lite 320x320.

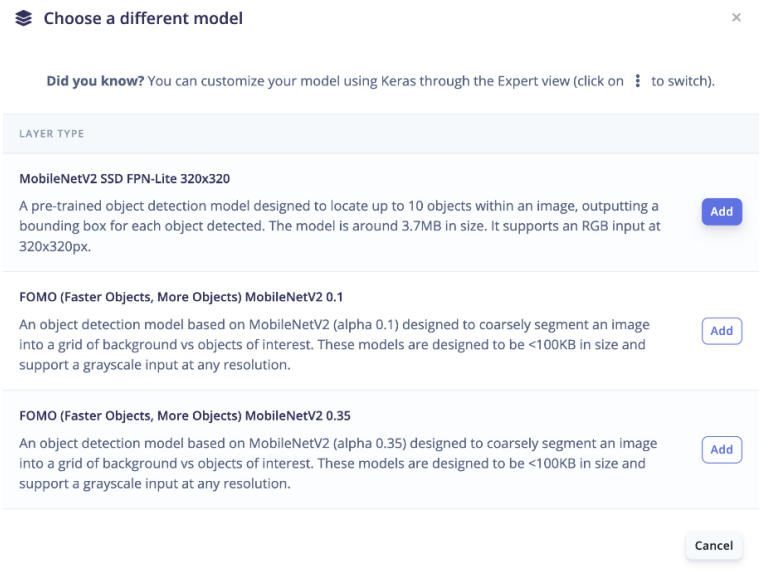

Figura 21 –Configuración de modelos MobileNet, para detección de objetos – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 6 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Una de las falencias de este modelo de detección de objetos es que solo funciona para la versión de Edge Impulse que corre en sistema operativo Linux, por lo cual se deberá evaluar la necesidad de generar esta red neuronal de acuerdo con los requerimientos que se desean saldar mediante esta aplicación de machine learning.

MobileNetV2, se basa en redes neuronales con bases de datos de entrada de imágenes con resolución de 320x320. Además, la red básica que utiliza mobileNet proporciona funciones apropiadas para la detección de objetos, ya que usa una capa de la red neuronal conectada en su totalidad y una capa softmax como última posición de la red neuronal.

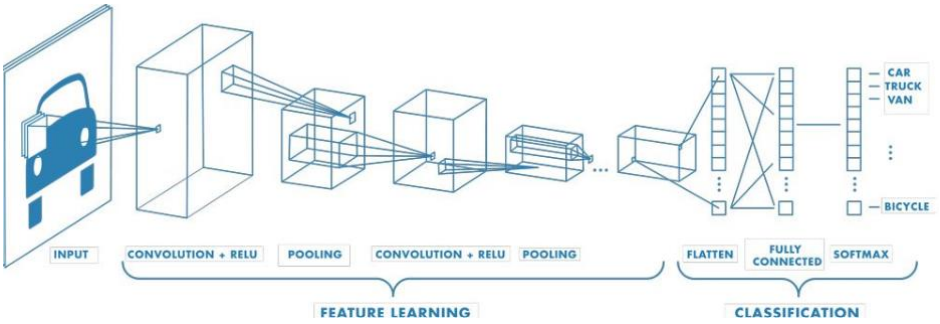

Figura 22 – arquitectura red neuronal MobileNet, para detección de objetos – Edge Impulse

Tomado de: "Edge Impulse Mathworks. Accedido el 7 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

En la Figura 22 – arquitectura red neuronal MobileNet, para detección de objetos – Edge Impulse, se observa la topología de una red neuronal MobileNetV2, la cual está compuesta por varias capas de convolución, las cuales aplican un filtro diferente a cada imagen que se encuentre dentro de la información entregada al modelo para el entrenamiento, por otro lado, la salida de cada capa es la entrada de la siguiente capa, por lo cual la salida final de la red neuronal será el objeto de interés, en el caso de la figura 21, será un vehículo automotor, descartando objetos como bicicletas.

3.1.1.10 Detección de objetos (Usando FOMO)

Por último, Edge Impulse ofrece un bloque de aprendizaje automático novedoso, que aplica detección de objetos a dispositivos con altas restricciones, este tipo de red neuronal permite realizar conteo de objetos, ubicación de estos en imágenes o el rastreo de objetos en tiempo real, a diferencia del procesamiento de modelos utilizados con MobileNet, FOMO tiene una potencia 30 veces mayor que este, por lo que realizar entrenamientos se vuelve una tarea más rápida y permite analizar más objetos.

Algunas características de la implementación de FOMO para realizar redes neuronales dirigidas a detección de objetos son:

- Detección de objetos a 30 FPS en placas de desarrollo de bajos recursos como Arduino NICLA Vision, en específico con la Raspberry pi 4 se alcanza una detección de objetos hasta de 60 FPS
- Utilización de memoria RAM de solo 245Kb
- Clasifica los objetos de imágenes y no clasifica como tal la imagen en una etiqueta única

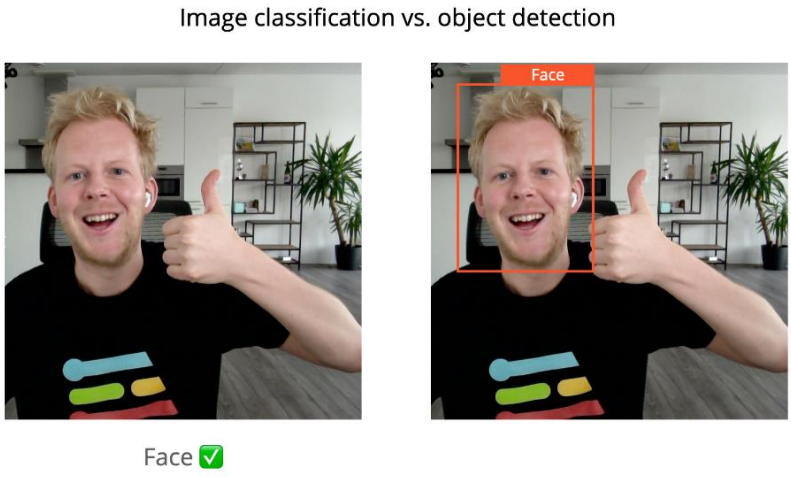

Figura 23 – Comparativa de detección de objetos vs clasificación de imágenes para detección de objetos – Edge Impulse

Tomado de: "Edge Impulse". Edge Impulse. Accedido el 7 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

Como se observa en la Figura 23 – Comparativa de detección de objetos vs clasificación de imágenes para detección de objetos – Edge Impulse, cuando se diseña con FOMO, es posible realizar la detección de objetos como caras o diferentes objetos que componen una imagen, a diferencia de la clasificación de imágenes que no reconoce los objetos de forma unitaria, sino que por lo contrario realiza el análisis de la imagen buscando patrones y le da una clasificación única a la totalidad de la imagen.

3.2 TENSORFLOW

TensorFlow es un software de aplicaciones para TinyML, se trata de una librería utilizada para realizar aprendizaje automático en machine learning, esta es desarrollo del equipo de investigación en inteligencia artificial de Google, este software propone construir redes neuronales con el fin de detectar patrones partiendo del concepto de razonamiento del ser humano.

TensorFlow es un sistema de código abierto originario de Google Cloud, por lo tanto, este se puede encontrar en la mayoría de los productos de esta sección de Google, debido a la compatibilidad de TensorFlow con estas plataformas, se puede decir que son ideales para el desarrollo de algoritmos de inteligencia artificial TinyML.

TensorFlow emplea el uso del aprendizaje automático para la realización de modelos de TinyML y de la misma forma que Edge Impulse se emplea para realizar tareas de verificación y validación de patrones mediante entrenamiento con datos de entrada. Este software se usa principalmente para evitar pérdidas de tiempo en el mantenimiento de infraestructuras , en caso contrario busca generar valor en los sistemas de inteligencia artificial aboliendo mano de obra humana, gestionando e implementado los procesos de aprendizaje automático recreados a partir de las variables ingresadas al modelo de la red neuronal, además TensorFlow brinda flexibilidad en las implementaciones que puede abarcarse en las redes neuronales y simplifica la ejecución y control de las mismas.

TensorFlow se puede implementar en aplicaciones, por ejemplo:

- Reconocimiento de imágenes.
- Análisis de mercado
- Diagnósticos médicos

Al igual que Edge Impulse estos tres campos son una muestra de la capacidad de la inteligencia artificial aplicada en TinyML, por ejemplo el reconocimiento de imágenes es de los casos más aplicados en TensorFlow, debido a que aplicando los diferentes tipos de aprendizaje automático para una red neuronal se puede evidencia parámetros y similitudes relevantes en las imágenes analizadas, este tipo de machine learning es altamente utilizado por corporaciones de seguridad privada con el fin de realizar procesamiento de señales y seguimiento vehicular y peatonal, por otro lado el análisis de mercado se enfoca más en las tendencias de marcas para generar publicidad y pautas de marketing en usuario con búsquedas específicas, la red neuronal llega a predecir los gustos y necesidades de cada uno de los usuario en una red social especifica y por último, el sector de la salud es uno de los campos con mayor potencial de aplicación de TinyML mediante TensorFlow, ya que debido a las cualidades de TinyML de desarrollar inteligencia artificial mediante sensores diminutos se pueden aplicar este tipo de soluciones en pacientes que requieran de monitoreo del estado de salud en tiempo real y control de dispositivos como válvulas pulmonares o válvulas aorticas.

3.2.1 Análisis TensorFlow Lite

TensorFlow cuenta con un conjunto de herramientas para la aplicación de TinyML, este se le denominó TensorFlow Lite, el cual es enfocado exclusivamente para aplicación de machine learning en dispositivos de bajos recursos, estos modelos de aprendizaje al igual que en el caso de Edge Impulse, se pueden implementar en dispositivos IoT o dispositivos móviles. Algunas características clave que se deben tener en cuenta en la implementación de modelos de bajos recursos mediante TensorFlow Lite son:

- **Optimización de aprendizajes automáticos integrados en equipos:** Al igual que Edge Impulse TensorFlow Lite ofrece soluciones que optimizan los modelos debido a que al utilizar TinyML, se producen reducciones de latencia, bajo consumo de red, bajo consumo de ancho de banda, mayor privacidad, menor consumo de energía. Estas características son propias de TinyML, más no de las plataformas Software de implementación de TinyML.
- **Compatibilidad con múltiples plataformas:** Al igual que Edge Impulse, TensorFlow Lite puede tener conectividad con sensores, microcontroladores, CPU, móviles que cuenten con sistemas operativos iOS, Android o Linux.
- **Compatibilidad con diversos lenguajes:** A diferencia de Edge Impulse, TensorFlow Lite acepta más tipos de lenguajes de programación, entre los que destacan Swift y Java, además recibe Objetive-C, junto C++ y Python, estos dos últimos también son aceptados por Edge Impulse.
- **Alto rendimiento:** TensorFlow Lite, maneja un sistema de aceleración de hardware y un sistema que optimiza los modelos de aprendizaje automático. Esta aplicación no se encuentra disponible para Edge Impulse, por lo cual es propia del aplicativo Software TensorFlow Lite.

3.2.1.1 Analisis TensorFlow Lite

para la creación de un modelo diseñado en TensorFlow Lite, se debe tener conocimiento que este tipo de proyectos se representa mediante un formato de modelo especial, el cual es eficiente y portátil, en el cual es conocido como *FlatBuffers.* Este es básicamente una biblioteca de serialización el cual funciona para diferentes tipos de lenguajes de programación, como Java, Javascript, Kotlin, C++, C#, Lua, PHP, TypeScript, Lobster, entre otros. Sin embargo, siempre se debe realizar la pregunta *¿Por qué FlatBuffers es ideal para los modelos de TinyML?* La respuesta a esta pregunta se obtiene con la explicación de las características que posee *FlatBuffers*, que cabe recalcar es de uso propio de TensorFlow Lite, por lo cual no se tiene información que otros aplicativos Software que creen modelos de aprendizaje para TinyML, puedan ser compatibles con esta librería.

3.2.1.1.1 Características de FlatBuffers en TensorFlow Lite

Flatbuffers al ser una librería de serialización propia de TensorFlow Lite fue creada por la compañía Google, con el fin de realizar el desarrollo de videojuegos u otros aplicativos que comprometan el rendimiento de los dispositivos, sin embargo, Se encuentra disponible actualmente como código abierto para realizar desarrollo de machine learning implementando esta librería, la cual cuenta con las siguientes características:

- **Acceso a datos serializados sin analizar:** Hace referencia a la forma de distinción de los datos que utiliza *FlatBuffers*, este representa los datos de forma binaria, de tal manera que pueda accederse directamente mediante un buffer sin analizar.
- **Eficiencia y velocidad de memoria:** *FlatBuffers* solo utiliza la memoria del buffer para tener acceso a los datos, por lo cual este tipo de librerías está dirigido a proyectos que requieran una velocidad de procesamiento superior a los demás y los tiempos de procesado de datos sea lo más corto posible. Esta característica puede variar con respecto a los lenguajes de programación utilizados.
- **Flexibilidad:** Se debe entender que *FlatBuffer* cuenta con la facilidad de ser compatibles con versiones anteriores a la librería, además no es necesario actualizar la base de datos del modelo para funcionar con *FlatBuffer*
- **Multiplataforma sin dependencia:** Nuevamente *FlatBuffer* puede ser utilizado en varias plataformas, mediante múltiples lenguajes de código como C++, Python, entre otros.

3.2.2 Dispositivos compatibles con TensorFlow Lite

TensorFlow Lite, es compatible con algunas placas de desarrollo de bajos recursos que cumplen las características de aplicación para TinyML, estas son:

ITEM	PLACA DE DESARROLLO
1	SparkFun Edge
2	Arduino Nano 33 BLE Sense
3	Kit de desarrollo STM32F746
4	EdgeBadge de Adafruit
5	Kit TensorFlow Lite de Adafruit para microcontroladores
6	Circuit Playground Bluefruit de Adafruit
7	ESP32-DevKitC de Espressif
8	ESP-EYE de Espressif
9	Wio Terminal: ATSAMD51
10	Placa de Desarrollo Himax WE-I Plus EVB Endpoint AI
11	Plataforma de Desarrollo de software Synopsys DesignWare ARC EM
12	Sony Spresense

Tabla 2 - Dispositivos compatibles con TensorFlow Lite

Tomado de: TensorFlow Lite para microcontroladores". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea].

De acuerdo a la Tabla 2 - Dispositivos compatibles con TensorFlow Lite, TensorFlow Lite es compatible con 12 placas de desarrollo de bajos recursos, para aplicaciones de TinyML, a diferencia de Edge Impulse que es compatible con 27 dispositivos, sin contar GPU's y dispositivos móviles con los sistemas operativos Linux, iOS o Android, lo que representa una notable diferencia entre las dos plataformas en cuanto a compatibilidad de dispositivo se trata, TensorFlow Lite se observa limitado en este aspecto.

TensorFlow Lite utiliza las funciones de delegación, las cuales cumplen el proceso de aceleración de hardware para aprovechar todos los recursos de las placas de desarrollo y dispositivos móviles, esto aprovecha de forma más eficiente la GPU y el procesador de señales digitales de estos dispositivos. TensorFlow Lite, utiliza como configuración predeterminada los núcleos de la CPU, sin embargo, estos en su mayoría no están optimizados para realizar procesos aritméticos que suelen ser utilizados en proyectos TinyML.

Por lo general se debería escribir códigos complejos para la optimización de la GPU de los dispositivos con el fin de ejecutar redes neuronales a través de interfaces personalizadas, sin embargo, este tipo de dispositivos cuentan con complicaciones debido a que cada acelerador es diferente, por lo cual se debería de realizar un código personalizado para cada uno de los procesadores en donde se requiera desplegar la solución de TinyML, por lo cual TensorFlow Lite, generó una API de delegado que resuelve este problema, actuando como puente entre el modelo y la optimización y aceleración de las GPU.

Las API delegados pueden escogerse de acuerdo con la plataforma en la cual se va a desplegar la red neuronal, estas pueden ser:

- Delegados por plataforma: Los cuales hacen referencia a las multiplataformas móviles como Android o iOS
- Delegados por tipo de modelo: De acuerdo con el tipo de modelo se deberá determinar un ancho de bits de datos para cada uno de los aceleradores, por lo cual TensorFlow Lite brinda la siguiente información para cuantificar el ancho de bits de datos

TIPO DE MODELO	GPU	NNAPI	HEXAGONO	COREML
Punto flotante (32 bits)	Si	Si	No	Si
Cuantificación float16 posterior al entrenamiento	Si	No	No	Si
Cuantificación del rango dinámico posterior al entrenamiento	Si	Si	No	No
Cuantificación de enteros posterior al entrenamiento	Si	Si	Si	No
Entrenamiento consciente de la cuantización	Si	Si	Si	No

Tabla 3 – API delegados - TensorFlow Lite

Tomado de: TensorFlow Lite para microcontroladores". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea].

- El tipo de delegado de hexagon se usa para acelerar modelos Android antiguos con procesadores Qualcomm Hexagon DSP.
- El tipo de delegado NNAPI se usa para acelerar modelos Android nuevos con sistema operativo Android 8.1 o superior
- El tipo de delegado COREML, se utiliza para dispositivos iOS que cuenten con la herramienta Neural Engine

3.2.2.1.1 Tipos de bloque de aprendizaje disponible en TensorFlow Lite

Para crear un modelo mediante TensorFlow Lite, la plataforma TensorFlow, pone a disposición una serie de modelos predeterminados al igual que lo hace Edge Impulse, sin embargo, en la plataforma TensorFlow Lite, se contemplan ejemplos que pueden ser utilizados como modelos de redes neuronales existentes, los cuales pueden contener bases de datos de entrada o no, por lo que se deberá realizar un análisis de este para la implementación de un proyecto nuevo.

3.2.2.1.2 Autocompletar

Los modelos de aprendizaje de redes neuronales de autocompletar son modelos de lenguaje LLM (Lenguaje extenso), las cuales tienen la función de generar texto a partir de grandes conjuntos de texto como base de la red neuronal. Este tipo de aprendizaje tiene utilidades de generación de texto, la respuesta de preguntas y traducción automática de textos como páginas web, etc.

Este tipo de modelos se deben ejecutar en Google Colab, debido a que este aplicativo de Software pertenece a Google, sin embargo esta plataforma es gratis para el desarrollo de proyectos de TinyML, mediante TensorFlow Lite, este último es diseñado para ejecutarse en placas de desarrollo de bajos recursos, para realizar este proceso el

modelo creado se deberá convertir en un modelo TensorFlow Lite, esto implicará en que el proyecto será más liviano, de bajo consumo y más rápido en el procesamiento de los datos.

Para realizar la conversión de modelos TensorFlow a TensorFlow Lite, se deberá utilizar la función de convertir modelos y posteriormente usar un intérprete de TensorFlow Lite, perteneciente a la misma plataforma de Software de TinyML,

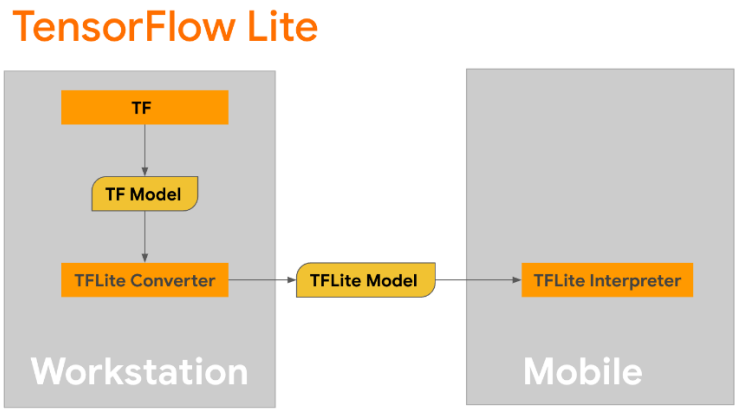

Figura 24 – Conversión modelos TensorFlow a TensorFlow Lite

Tomado de: "Autocompletar | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

De acuerdo con la Figura 24 – Conversión modelos TensorFlow a TensorFlow Lite, todos los modelos TensorFlow que deseen ser utilizados en dispositivos móviles o placas de desarrollo de bajos recursos, deberán realizarse mediante este proceso, en donde el modelo TensorFlow, será convertido a un modelo TensorFlow Lite, dentro de la misma estación de trabajo perteneciente al proyecto creado, que en el dispositivo deberá ser ejecutado en un intérprete de TensorFlow Lite.

Esta conversión se realiza mediante la función *generate (),* la cual se encarga de la conversión de los modelos TensorFlow, esta funciona al igual que *from_keras_model(),* posteriormente se deberá crear una función de inferencia con un valor de entrada y un modelo TensorFlow Lite.

```
def run_inference(input, generate_tflite):
    interp = interpreter.InterpreterWithCustomOps(
        model_content=generate_tflite,
        custom_op_registerers=
            tf_text.tflite_registrar.SELECT_TFTEXT_OPS
    )

    interp.get_signature_list()

    generator = interp.get_signature_runner('serving_default')
    output = generator(prompt=np.array([input]))


gpt2_lm.jit_compile = False
converter = tf.lite.TFLiteConverter.from_concrete_functions(
    [concrete_func],
    gpt2_lm)

converter.target_spec.supported_ops = [
    tf.lite.OpsSet.TFLITE_BUILTINS, # enable TFLite ops
    tf.lite.OpsSet.SELECT_TF_OPS, # enable TF ops
]
converter.allow_custom_ops = True
converter.target_spec.experimental_select_user_tf_ops = [
    "UnsortedSegmentJoin",
    "UpperBound"
]
converter._experimental_guarantee_all_funcs_one_use = True
generate_tflite = converter.convert()
run_inference("I'm enjoying a", generate_tflite)
```

Figura 25 – Función de inferencia TensorFlow a TensorFlow Lite

Tomado de: "Autocompletar | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

De acuerdo con la Figura 25 – Función de inferencia TensorFlow a TensorFlow Lite, se puede observar la sintaxis de código para la generación de la función de inferencia, la cual TensorFlow proporciona como código abierto, adicional a esto se deberá contar con una función que permita la optimización del modelo, para que funcione en aplicaciones de TinyML.

```
gpt2_lm.jit_compile = False
converter = tf.lite.TFLiteConverter.from_concrete_functions(
    [concrete_func],
    gpt2_lm)

converter.target_spec.supported_ops = [
    tf.lite.OpsSet.TFLITE_BUILTINS, # enable TFLite ops
    tf.lite.OpsSet.SELECT_TF_OPS, # enable TF ops
]
converter.allow_custom_ops = True
converter.optimizations = [tf.lite.Optimize.DEFAULT]
converter.target_spec.experimental_select_user_tf_ops = [
    "UnsortedSegmentJoin",
    "UpperBound"
]
converter._experimental_guarantee_all_funcs_one_use = True
quant_generate_tflite = converter.convert()
run_inference("I'm enjoying a", quant_generate_tflite)
```

Figura 26 – Función de cuantificación TensorFlow a TensorFlow Lite

Tomado de: "Autocompletar | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

Mediante la función observada en la Figura 26 – Función de cuantificación TensorFlow a TensorFlow Lite, se puede optimizar el tamaño del modelo para que cumpla con las características de bajos recursos de TinyML, adicional las variables del modelo original TensorFlow, con la función de cuantificación convierte todos los datos Float (Flotantes), que originalmente tienen un tamaño de 32 bits a variables enteras de 8 bits.

3.2.2.1.3 Clasificación de imágenes

El modelo de clasificación de imágenes funciona de la misma forma par TensorFlow y Edge impulse, utilizando también el aprendizaje por transferencia para las redes neuronales, este se enfoca en identificar las clases de imágenes existentes en la base de datos entregada al modelo con el fin de clasificarlas y añadirles una probabilidad a cada una y es muy útil en caso de reconocimiento de caras o formas como la de los animales.

tipo de animal	Probabilidad
Conejo	0.07
Hámster	0.02
Perro	0.91

Figura 27 – Ejemplo de clasificación de imágenes TensorFlow Lite

Tomado de: "Autocompletar | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

Como se muestra en la Figura 27 – Ejemplo de clasificación de imágenes TensorFlow Lite, cada uno de los valores de probabilidad hacen referencia a las salidas del modelo de aprendizaje por imágenes, estos valores cuentan con etiquetas individuales y la probabilidad de las salidas siempre dará 1.

Por otro lado, en este tipo de aprendizajes automáticos se pueden presentar resultados ambiguos, debido a que en el proceso de entrenamiento el modelo no logra distinguir los patrones de cada una de las imágenes y etiqueta los datos con probabilidades similares, por ende, en este tipo de casos se deberá optar por un tipo de modelo diferente al empleado para la clasificación de imágenes.

Etiqueta	Probabilidad
Conejo	0.31
hámster	0.35
perro	0.34

Figura 28 – Ejemplo de resultados ambiguos - clasificación de imágenes TensorFlow Lite

Tomado de: "Autocompletar | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

Como se observa en la Figura 28 – Ejemplo de resultados ambiguos - clasificación de imágenes TensorFlow Lite, las probabilidades de salida del modelo de aprendizaje son similares, por lo cual la red neuronal no es capaz de verificar cual es el patrón que cuenta con mayor similitud, por ende, en estos casos, el tipo de modelo requerirá un ajuste en los datos ingresados para el entrenamiento u optar por un tipo de modelo de red neuronal con mayor precisión.

TensorFlow Lite cuenta con las opciones de cambiar modelos de clasificación de imágenes, dado el caso que el modelo empleado cuente con resultados ambiguos, a diferencia de Edge Impulse, esta plataforma de software de TinyML, cuenta con tres tipos de arquitectura para modelos de clasificación de imágenes, estos son:

- MobileNet
- Inception
- NASNet

De los cuales MobileNet es uno de los modelos que TensorFlow y Edge Impulse comparten disponibilidad, por otro lado, Inception y NASNet, son modelos que se encuentran solo en TensorFlow Lite. Estos modelos de clasificación cuentan con la posibilidad de clasificar hasta 1000 tipos de imágenes, sin embargo, un punto a aclarar es que, si se desea realizar predicción de tipo y posición de objetos dentro de imágenes, este tipo de modelos no son aptos para desempeñar esta función, al igual que la predicción de composición de imágenes con contrastes.

3.2.2.1.4 Detección de objetos

Los modelos empleados por TensorFlow Lite para la detección de objetos, emplean el aprendizaje de detectores de disparo único, este modelo detecta la presencia y la posición de varios tipos de objetos en una misma imagen, este tipo de entrenamiento se puede utilizar con imágenes previamente ingresadas al sistema y posteriormente si se desea incluir nuevos datos, este es capaz de detectar los objetos con similitudes a los entregados en la base de datos inicial, generando listados de elementos encontrados con su ubicación espacial en las imágenes.

Este tipo de modelos de detección de objetos en los resultados de cada uno de los entrenamientos genera una tabla con asignación de índices, nombres y objetos detectados con su respectiva descripción.

Clase	Puntaje	Ubicación
Manzana	0.92	[18, 21, 57, 63]
Banana	0.88	[100, 30, 180, 150]
Fresa	0.87	[7, 82, 89, 163]
Banana	0.23	[42, 66, 57, 83]
Manzana	0.11	[6, 42, 31, 58]

Figura 29 – Ejemplo de resultados Detección de objetos TensorFlow Lite

Tomado de: "Detección de objetos | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

Como el ejemplo que se puede observar en la Figura 29 – Ejemplo de resultados Detección de objetos TensorFlow Lite, este tipo de modelos representa los datos de salida en una matriz con 3 secciones, la primera columna hace referencia al objeto detectado, la segunda columna es la probabilidad de que el objeto encontrado en la imagen sea en este caso, una banana, fresa o manzana y en la tercera columna es la posición en la cual encontró el objeto dentro de la imagen, esta última columna se lee de la siguiente forma:

[arriba, izquierda, abajo, bien]

Figura 30 – Lectura coordenadas - Detección de objetos TensorFlow Lite

Tomado de: "Detección de objetos | TensorFlow Lite". TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/auto_complete/overview?hl=es-419

Como se observa en la Figura 30 – Lectura coordenadas - Detección de objetos TensorFlow Lite, las coordenadas que aparecen en cada uno de los objetos encontrados mediante los resultados obtenidos del modelo se deberán leer como "arriba" hace referencia al valor superior que representa la distancia que existe entre el objeto y el borde superior de la imagen, así mismo con "izquierda", que hace referencia a la distancia que existe entre el borde izquierdo de la imagen y el borde del objeto encontrado, de esta forma se deberá analizar el resto de variables de estos datos obtenidos.

3.2.2.1.5 Estimación de poses

El modelo de red neuronal de la estimación de poses es la capacidad de poner definir las poses de una persona teniendo como punto de partida imágenes o videos, esto se logra realizando la estimación de posicionamiento de la anatomía del ser humano.

El modelo de estimación de poses, hace referencia a las técnicas que detectan contornos humanos, que determinan extremidades de estos, este tipo de modelos cuenta con una cámara como entrada de datos para el entrenamiento, y su salida o resultados obtenidos a partir de este modelo genera un listado de información indexada por cada uno de los puntos clave encontrados, utilizando puntuaciones entre 0 y 1, como probabilidades que indican que cada movimiento detectado, puede ser una extremidad del cuerpo humano.

TensorFlow Lite, proporciona dos modelos para la estimación de poses, estos son MoveNet y PoseNet, los cuales son propios de TensorFlow y este tipo de modelos es de implementación de TensorFlow, por lo cual plataformas de software como Edge Impulse no cuentan con este tipo de bloques de aprendizaje ya predeterminados dentro de su repertorio de herramientas para TinyML.

Identificación	Parte	Identificación	Parte
0	nariz		
1	ojo izquierdo	9	Muñeca izquierda
2	Ojo derecho	10	muñeca derecha
3	oreja izquierda	11	cadera izquierda
4	oreja derecha	12	cadera derecha
5	hombro izquierdo	13	rodilla izquierda
6	hombro derecho	14	rodilla derecha
7	codo izquierdo	15	tobillo izquierdo
8	codo derecho	dieciséis	tobillo derecho

Figura 31 – Estimación de poses – Identificación de extremidades TensorFlow Lite

Tomado de: "Estimación de poses | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/pose_estimation/overview?hl=es-419

TensorFlow Lite, cuenta ya con esta configuración para realizar la identificación de poses de las extremidades del cuerpo humano, por lo cual facilita el análisis de los datos obtenidos, como se muestra en la Figura 31 – Estimación de poses – Identificación de extremidades TensorFlow Lite. A su vez si es un video lo que se ha ingresado en el modelo este generará los puntos de interés en la base de datos generando una salida en video en tiempo real.

Figura 32 – Estimación de poses – Identificación de extremidades en video TensorFlow Lite

Tomado de: "Estimación de poses | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/pose_estimation/overview?hl=es-419

Tal y como se puede observar en la figura Figura 32 – Estimación de poses – Identificación de extremidades en video TensorFlow Lite, los puntos de interés o extremidades encontradas mediante el modelo de aprendizaje automático puede también en tiempo real seguir los movimientos mediante predicciones debido a las poses generadas por el ser vivo analizado en cada entrenamiento.

3.2.2.1.6 Reconocimiento de voz

Este modelo al igual que el reconocimiento de voz presentado por Edge Impulse para aplicativos TinyML, reconoce comandos de voz mediante palabras clave que se

etiquetan al inicio del entrenamiento de esta red Neuronal, por otro lado, TensorFlow Lite no cuenta con información de este modelo, sin embargo cuenta con una lista de ejemplos para realizarse despliegues en dispositivos iOS y Android Studio, por otro lado, al ingresar a estos ejemplos se observa que este tipo de aprendizaje automático está disponible para su despliegue en Raspberry Pi, sin embargo no se cuenta con más información sobre otro tipo de placas de desarrollo en las que se pueda realizar este tipo de modelos de red neuronal. Para más información se deberá indagar en las librerías de Github en donde se encuentran estos ejemplos. Cabe aclarar que el reconocimiento de voz mediante comandos es un modelo de red neuronal que es aplicable en TensorFlow Lite y no en Edge Impulse, el cual no cuenta con este tipo de bloques de aprendizaje predeterminado por parte de la plataforma software para TinyML.

3.2.2.1.7 Reconocimiento de gestos

El reconocimiento de gestos es un modelo de red neuronal que aplica el aprendizaje por transferencia, que al igual que la estimación de poses utiliza un sistema de puntuación e ID para etiquetar los gestos humanos, utilizando como dispositivo de campo las cámaras de las placas de desarrollo o de los dispositivos móviles, esta red neuronal está disponible para el despliegue en dispositivos Android, iOS y Rapberry Pi, sin embargo, al igual que el reconocimiento de comandos de voz, no se puede encontrar mayor información dentro de la plataforma Software para TinyML en cuestión, lo cual es una desventaja para usuarios principiantes que deseen emplear TensorFlow como plataforma Software para iniciar en proyectos de TinyML, para encontrar mayor información se deberá remitir a los ejemplos que la plataforma dispone a los usuarios en la plataforma GitHub. Por otro lado, cabe mencionar que la plataforma TensorFlow cuenta con una amplia variedad de bloques de aprendizaje de los cuales 15 de estos cuentan con información resumen propia de la plataforma para realizar los entrenamientos y que el programador logre captar la información del modelo de forma óptima.

3.2.2.1.8 Segmentación

La red Neuronal que aplica la segmentación se basa básicamente en dividir imágenes en conjuntos de pixeles u objetos de imagen, con el propósito de simplificar la imagen, para su análisis. Este proceso se realiza en dispositivos Android, iOS y placas de desarrollo con características de bajos recursos para aplicativos de TinyML.

Este tipo de red neuronal aplica el aprendizaje profundo el cual agrega etiquetas a los objetos divididos de la imagen inicial.

Figura 33 – Segmentación – Identificación de objetos - TensorFlow Lite

Tomado de: "Segmentación | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/segmentation/overview?hl=es-419

En la Figura 33 – Segmentación – Identificación de objetos - TensorFlow Lite, se puede observar, el resultado de la segmentación en una imagen, este tipo de red neuronal extrae objetos con base en las etiquetas que se han configurado por parte del programador de forma anticipada al entrenamiento del modelo. Este tipo de proyectos se realiza mediante los modelos DeepLabv1, DeepLavv2 y DeepLavv3, estos utilizan una agrupación espacial de la imagen y segmenta los objetos de interés con base en las características que son relevantes en la información de la imagen. Cabe mencionar, que este tipo de aprendizaje automático solo está disponible para dispositivos Android, mediante la interfaz de Android Studio, Dispositivos iOS y Raspberry Pi.

3.2.2.1.9 Clasificación de texto

El modelo de red neuronal para la clasificación de texto, realiza el etiquetado de párrafos en grupos predeterminados en función a la información que estos contengan, esto a su vez realiza predicciones positivas o negativas en cada uno de los párrafos, mediante la clasificación de cada uno de estos se logra convertir en una lista que contiene identificadores de palabras con base a un vocabulario predeterminado, de esta forma mediante el entrenamiento los resultados de salida del modelo tendrán una puntuación o probabilidad de que el párrafo ingresado como información de entrada a la red neuronal sea positivo o negativo.

Texto	Negativo (0)	Positivo (1)
Esta es la mejor película que he visto en los últimos años. ¡Lo recomiendo encarecidamente!	25,3%	74,7%
¡Qué desperdicio de mi tiempo.	72,5%	27,5%

Figura 34 – Clasificación de texto – TensorFlow Lite

Tomado de: "Clasificación de texto | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/text_classification/overview?hl=es-419#how_it_works

Como se observa en la Figura 34 – Clasificación de texto – TensorFlow Lite, la salida de información del modelo se verá reflejada en comentarios negativos y positivos, los cuales contarán con una clasificación mediante porcentajes, que son probabilidades de que el texto ingresado sea una crítica positiva o negativa, este tipo de bloque de aprendizaje no se encuentra dentro del repertorio de Edge Impulse, solo se encuentra en TensorFlow Lite. Por otro lado, cabe recalcar que una de las limitaciones con las que cuenta este tipo de modelos de aprendizaje automático, no admite lenguaje inglés y es apto para cuantificar las reseñas de película o recopilación de comentarios de estas por lo cual si se desea enfocar en otro tipo de proyecto se podría presentar falencias en la precisión del modelo.

3.2.2.1.10 Recomendaciones en el dispositivo

Las redes neuronales que se emplean para la recomendación de información se utilizan en su mayor parte en dispositivos móviles, para realizar recomendaciones de contenidos streaming de interés, productos en línea, aplicaciones, entre otro tipo de funcionalidades que están enfocadas al marketing digital. Este tipo de redes neuronales se puede aplicar con parámetros de recomendación predefinidos en el modelo.

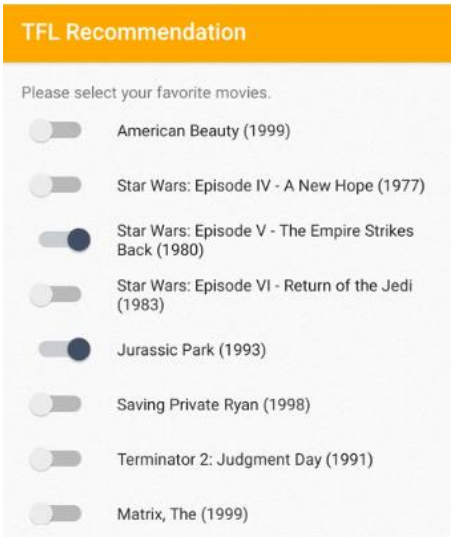

Figura 35 – Recomendación de contenido – TensorFlow Lite

Tomado de: "Recomendación de contenido | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/recommendation/overview?hl=es-419

Este tipo de modelos se puede realizar mediante la colaboración de colab o mediante ejemplos predeterminados y a disposición del usuario mediante la aplicación TensorFlow Lite, por lo cual está disponible para ejecutarse en dispositivos móviles de diferentes sistemas operativos y placas de desarrollo de bajos recursos. A su vez este modelo cuenta con una arquitectura de codificador dual que consiste en codificar el historial de búsqueda de los usuarios y etiquetar los temas de interés con el fin de realizar recomendaciones con base en la información de entrada del modelo de red neuronal, esta etiqueta representa la probabilidad de que el contenido sugerido sea de interés de cada uno de los usuarios por lo cual es el tipo de red neuronal más utilizado para realizar campañas de marketing para empresas, películas, disponibilidad de servicios en línea, entre otro tipo de enfoques que puedan requerir recomendaciones de contenido.

3.2.2.1.11 Uso de lenguaje natural

El modelo de la red neuronal que emplea el aprendizaje automático de lenguajes naturales de TensorFlow Lite, utiliza un modelo de codificación bidireccional, este modelo es un método que representa el lenguaje como una gama de tareas de procesamiento.

Este modelo utiliza preguntas como datos de entrada, para posteriormente devolver un dato de salida con una respuesta a las preguntas cuya información se basará en datos probabilísticos obtenidos a partir de preprocesamientos de la información de entrada anteriormente mencionada.

Esta red neuronal se puede aplicar en todos los dispositivos que aceptan TensorFlow Lite, sin embargo, es mayormente utilizado en enfoques de dispositivos móviles o aplicativos de inteligencia artificial, como lo es ChatGPT u otros aplicativos de información.

Para ejecutar este tipo de modelos se deberá contar con los siguientes datos para que el entrenamiento del modelo TinyML se ejecute de forma precisa:

- Pasaje (entrada): el pasaje o la información de entrada es la base de datos que tiene la red neuronal sobre datos que posiblemente el usuario pueda utilizar para hacer investigaciones posteriores, por ejemplo, información base encontrada en internet o información tomada de textos de literatura.
- Pregunta (entrada): este hace referencia a la interacción que se realizará por parte del usuario del modelo de red neuronal y el modelo mismo, pueden ser preguntas cortas o explicaciones extensas sobre temas de interés.
- Respuesta (Salida): la respuesta o la salida del modelo hace referencia a las oraciones que tengan mayor probabilidad de acierto para dar solución a las preguntas o explicaciones que el usuario del modelo requiera, nuevamente podrán llegar a ser respuestas cortas sin complicación alguna o respuestas largas que den explicación a más de una incógnita encontrada.

Con base en lo anteriormente descrito el modelo de pregunta y respuesta se despliega mediante MobileBERT, el cual es un modelo para el despliegue de proyectos como chatbots entre otras aplicaciones y su principal característica es que tienen una ejecución más rápida y de tamaño moderado con respecto a otros modelos de aprendizaje automático de este tipo. Cabe mencionar que TensorFlow Lite es la única plataforma de software gratuito que cuenta con este tipo de bloques de aprendizaje predeterminados para la realización de modelos de redes neuronales de este estilo.

3.2.2.1.12 Transferencia de estilo

Uno de los bloques de aprendizaje automático que tiene TensorFlow Lite más interesantes es el modelo de red neuronal que aplica la transferencia de estilo artístico, la cual da la posibilidad de crear nuevas imágenes con base a dos imágenes de entrada en el modelo de la red neuronal, en donde cada una tiene un objetivo específico que se deberá configurar antes de iniciar el entrenamiento del modelo, en donde una de las imágenes deberá tomar el papel de representar el estilo artístico que se quiere crear y la otra imagen será el contenido de esta.

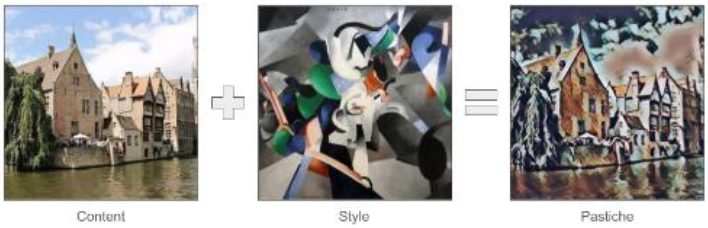

Figura 36 – Transferencia de estilo artístico – TensorFlow Lite

Tomado de: "Transferencia de estilo artístico | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/style_transfer/overview?hl=es-419

En la figura Figura 36 – Transferencia de estilo artístico – TensorFlow Lite, se observa como las dos imágenes de entrada una vez definida la funcionalidad en el modelo de aprendizaje, plasmas el tipo de imagen que se crea a partir de las dos bases de información de

entrada. Sin embargo, para comprender este tipo de aprendizaje se debe entender que este modelo consta de dos tipos de aprendizajes fusionados, los cuales son:

- Modelo de predicción de estilo: El cual emplea el modelo A-MoiblenetV2, el cual acepta imágenes de dimensiones desde los 100 x 100 pixeles
- Modelo de transformación de estilo: este es una red neuronal que aplica el estilo del contenido de las imágenes, y crea imágenes a partir del requerimiento del programador

Este tipo de bloque de aprendizaje es posible de ejecutarse en dispositivos móviles, iOS, Linux, placas de desarrollo incluyendo la raspberry Pi.

3.2.2.1.13 Respuesta inteligente

Este modelo de respuesta inteligente ofrecido en la plataforma TensorFlow Lite como uno de los bloques de aprendizaje predeterminado del aplicativo software para TinyML, ofrece sugerencias basadas en preguntas o entradas de texto realizadas en un chat, por lo cual es sumamente útil para empresas cuya demanda de atención al cliente es elevada y se puede hacer uso de Chatbots que den soluciones a temas generales o preguntas frecuentes por parte de los usuarios.

El modelo funciona generando sugerencias de respuestas a los mensajes que los usuarios ponen en el chat, por lo cual esta sería la información de entrada al modelo de respuesta rápida, una vez ingresada esta información el sistema arrojará una serie de respuestas con probabilidad de acierto no mayor a 1 entre las sugerencias desplegadas, que a su vez generaran una información extraída de la base de datos a partir de la solicitud de cada uno de los usuarios.

Este modelo cuenta con beneficios como:

- Rapidez: El modelo suele estar instalados en los aplicativos móviles o páginas web de los establecimientos por lo cual suele estar desconectado a internet, lo que mejora la velocidad de respuesta.
- Eficiencia: al ser un modelo con una base de datos limitada, incrementa la eficiencia de recursos de procesamiento al dejar una huella de memoria pequeña en el dispositivo en donde se despliegue el modelo de red neuronal.
- Privacidad: Al normalmente estar desconectado de internet, el modelo suele estar instalado directamente en los dispositivos finales, por lo que fugas de información suelen ser poco frecuentes en este tipo de aplicativos.

3.2.2.1.14 Superresolución

Este modelo de aprendizaje proporcionado por la plataforma de software TensorFlow Lite, realiza la recuperación de calidad utilizando los conceptos de Alta resolución (HR) y superresolución de imagen única (SISR). El modelo ESRGAN, utilizado por TensorFlow, convierte la imagen de baja resolución a imágenes de alta resolución con un factor de escala igual a 4, este valor es predeterminado por el modelo de aprendizaje, sin embargo, es voluntad del programador el cambio de estos parámetros predeterminados.

```
lr = tf.io.read_file(test_img_path)
lr = tf.image.decode_jpeg(lr)
lr = tf.expand_dims(lr, axis=0)
lr = tf.cast(lr, tf.float32)

# Load TFLite model and allocate tensors.
interpreter = tf.lite.Interpreter(model_path=esrgan_model_path)
interpreter.allocate_tensors()

# Get input and output tensors.
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

# Run the model
interpreter.set_tensor(input_details[0]['index'], lr)
interpreter.invoke()

# Extract the output and postprocess it
output_data = interpreter.get_tensor(output_details[0]['index'])
sr = tf.squeeze(output_data, axis=0)
sr = tf.clip_by_value(sr, 0, 255)
sr = tf.round(sr)
sr = tf.cast(sr, tf.uint8)
```

Figura 37 – Superresolución – TensorFlow Lite

Tomado de: "Superresolución | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/style_transfer/overview?hl=es-419

Mediante el código visualizado en la Figura 37 – Superresolución – TensorFlow Lite, se puede realizar el cambio de resolución de una imagen de entrada SISR, la cual cuenta con baja calidad a una imagen HR en la salida del modelo de aprendizaje automático. Cabe aclarar que este es el fragmento de conversión de imagen, si se desea ejecutar el código completo se deberá remitir al bloque de aprendizaje predeterminado en donde se contará con la opción de despliegue en diferentes plataformas, sin embargo, se recomienda que se utilice Colab, para la visualización y el claro entendimiento del proceso de conversión y la codificación del algoritmo de conversión, esto en caso de que el programador no sea experto en la aplicación de TinyML.

3.2.2.1.15 Clasificación de audio

La red neuronal que aplica la clasificación de audio en TensorFlow Lite, se basa en la representación de audios en la cual, el modelo se entrena para reconocimiento de eventos sonoros, como aplausos, carcajadas, escritura o sonidos de instrumentos como violines, trompetas, entre otras aplicaciones de este modelo.

La red neuronal de clasificación de audio utiliza en simultaneo el modelo de aprendizaje YAMNet el cual está disponible para TensorFlow Lite, este es un modelo que clasifica los audios originales en la entrada del sistema con el fin de realizar la entrega de resultados mediante un vector de puntuaciones que hacen referencia a las probabilidades de que los sonidos encontrados puedan ser de los objetos solicitados previamente al entrenamiento de la red neuronal.

Algunas limitaciones de YAMNet para la clasificación de audios son:

- No puede interactuar directamente con las salidas del sistema debido a que es difícil calibrar las clases de sonidos que se van a analizar en el modelo
- Puede tener discrepancia entre las entradas de audio y las tareas solicitadas a realizar
- Se debe realizar un ajuste de parámetros y calibraciones para que el modelo pueda ser utilizado en múltiples plataformas de desarrollo

Cabe aclarar que el sistema de despliegue de este tipo de modelos de aprendizaje que mejor se adapta a la clasificación de audio es Android, por lo cual TensorFlow Lite recomienda realizar este tipo de despliegues en esta plataforma.

3.2.2.1.16 Clasificación de videos

La clasificación de video es un aprendizaje automático proporcionado por TensorFlow Lite, con el fin de identificar acciones que se observan en tiempo real, este tipo de modelos utilizan como base una serie de datos de video en los cuales se configuran conjuntos de clases, que se catalogan como acciones o movimientos, este recibe como entrada cuadros de videos y genera resultados de salida clases a las cuales se les asigna una puntuación o probabilidad de que sea un movimiento especifico.

Este tipo de modelos de redes neuronales se puede configurar para reconocer acciones como aplausos, saludos, o acciones humanas como caminar o correr. Este tipo de red neuronal utiliza los modelos de aprendizaje de MoViNets, los cuales son optimizados para dispositivos móviles, adicional a esto, representan una optimización y eficiencia del uso y disposición de los conjuntos de datos de entrada para reconocer acciones de videos a gran escala.

Una de las grandes ventajas de este tipo de modelos es que recibe la información y la procesa en tiempo real, a medida que el video va transcurriendo el modelo identifica las clases de conjuntos de datos, devolviendo datos de salida mediante una tabla que combina el tipo de acción y la probabilidad de que esta sea la correcta.

Acción	Probabilidad
baile cuadrado	0.02
aguja de enhebrar	0.08
jugueteando con los dedos	0.23
Mano que saluda	0,67

Figura 38 – Clasificación de video – TensorFlow Lite

Tomado de: "Clasificación de video | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/video_classification/overview?hl=es-419

Como se observa en la Figura 38 – Clasificación de video – TensorFlow Lite, el modelo de clasificación de videos entrega en tiempo real el tipo de acción que se está presentando en la cinemática, asociada a una probabilidad no mayor a 1. Este tipo de modelos se pueden emplear en conferencias o pruebas deportivas con el fin de visualizar movimientos y evaluar el desempeño de las personas puestas a prueba.

Este tipo de modelo recibe secuencias de cuadras tipo video RGB como entrada y en general no se tiene un tamaño de la base de datos de entrada como límite sin embargo TensorFlow Lite recomienda que este sea compatible con la resolución y la velocidad de fotogramas de los videos entregamos para el entrenamiento del modelo.

3.2.2.1.17 Reconocimiento óptico de caracteres

El reconocimiento óptico de caracteres es un modelo de red neuronal que emplea, el modelo de detección de textos para realizar el análisis de los textos ingresados como entrada en la red neuronal y un modelo de reconocimiento de texto con el que se define los caracteres específicos dentro de esta aplicación.

Las entradas de los OCR, deberán ser un arreglo de caracteres no superior a 32 bits de información y un arreglo que reconozca este tipo de caracteres de igual tamaño, con esto, se inicia el entrenamiento del modelo y se obtienen valores de salida que reconocen los patrones de texto.

Este tipo de modelos funcionan de forma eficiente en la captura de imágenes y traslado de texto de las mismas a formato texto, realizando las tareas de transcripción con mayor eficiencia, minimizando así el error humano, cabe aclarar que este modelo de aprendizaje automático es propio para sistemas operativos Android, no cuenta con soporte para despliegue en placas de desarrollo ni sistemas operativos Linux, ni iOS, por lo cual representa una gran limitante para la expansión de proyectos de este tipo de redes neuronales.

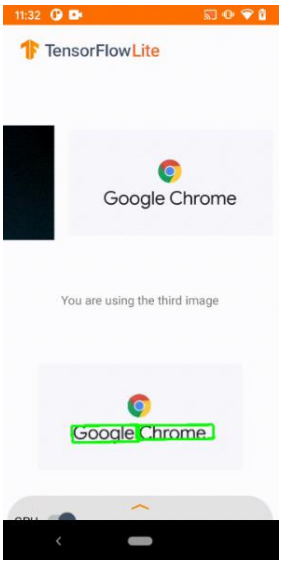

Figura 39 – Reconocimiento óptico de caracteres – TensorFlow Lite

Tomado de: "Reconocimiento óptico de caracteres | TensorFlow Lite". TensorFlow. Accedido el 10 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/examples/optical_character_recognition/overview?hl=es-419

Como se observa en la Figura 39 – Reconocimiento óptico de caracteres – TensorFlow Lite, el aprendizaje automático de reconocimiento óptico de caracteres reconoce en imágenes las letras que conforman las palabras, dando así una salida de en formato escrito en este caso la palabra que se encontró con patrones de caracteres es *Google Chrome,* de esta forma el reconocimiento óptico funciona para facilitar funciones de traducción de imágenes o reconocer patrones de interés en estas.

3.3 EFINIX

Efinix es una plataforma software que aplica aprendizajes automáticos para la aplicación de inteligencia artificial bajo las técnicas de TinyML. Esta plataforma utiliza un software llamado Efinity, propio del aplicativo para realizar diseños y despliegue de estos en placas de desarrollo FPGA, este software tiene compatibilidad con Windows, iOS, Ubunto (linux) y su placa de desarrollo predeterminada es FPGA TITANIUM, la cual cuenta con un alto rendimiento y bajo consumo de memoria, características fundamentales para la implementación de proyectos TinyML.

La plataforma de software para TinyML Efinix, cuenta con una amplia comunidad la cual desarrollo códigos abiertos para modelos TinyML Efinix con el acoplamiento del software TensorFlow, en su versión TinyML Tensorflow Lite, con el fin de cuantificar modelos estándar que utilizan bloques de aprendizaje automático derivados de tensorflow, junto con bibliotecas de funciones que permiten a los microcontroladores de bajos recursos sacar su máximo potencial en las aplicaciones de TinyML.

Por otro lado, la plataforma de Efinix para TinyML, utiliza las capacidades de instrucciones del núcleo Sapphire, esta plataforma utiliza este núcleo debido a que mejora de forma significativa las placas de desarrollo FPGA en cuanto a consumo energético y rendimiento de memoria de estas.

Efinix adopta la tecnología de RISC-V, lo que permite crear aplicaciones y algoritmos de inteligencia artificial aprovechando la facilidad de la programación de las tarjetas de desarrollo FPGA, lo cual permite mayor flexibilidad en cuanto a rendimiento y proyectos de modelos de aprendizaje basados en TinyML, además la plataforma de software ofrece técnicas propias de aceleración de hardware basado en FPGA, simplificando así el aprendizaje del lenguaje de programación VHDL, el desarrollo y diseño de proyectos.

En pocas palabras, Efinix combina el conocido entorno de desarrollo RISC-V ISA y aprovecha sus capacidades de comando personalizadas para operar dentro de un marco FPGA arquitectónicamente flexible.

A diferencia de muchos aceleradores de hardware, este enfoque no requiere herramientas ni compiladores de terceros. La aceleración también se ajusta mediante la aceleración de instrucciones de la máquina. Este es un nivel de granularidad que sólo es significativo para FPGA. El hecho de que los dispositivos de borde se puedan crear prototipos e implementar en la innovadora arquitectura de diseño FPGA de Efinix significa que la solución está preparada para el futuro. Los nuevos modelos y las arquitecturas de red actualizadas se pueden expresar en un entorno de software familiar y acelerarse en el nivel de instrucción personalizada usando solo una pequeña cantidad de VHDL.

Este nivel de separación de hardware y software permite que el 90% del modelo permanezca en software que se ejecuta en RISC-V, lo que reduce significativamente el tiempo de ejecución de tareas. La combinación de todos estos enfoques crea una solución eficaz que reduce la barrera de entrada para el uso de dispositivos periféricos. Los programadores de modelos de inteligencia artificial pueden acceder a capacidades de procesamiento integradas de clase mundial instanciadas en Efinix Quantum utilizando una línea de herramientas de última generación.

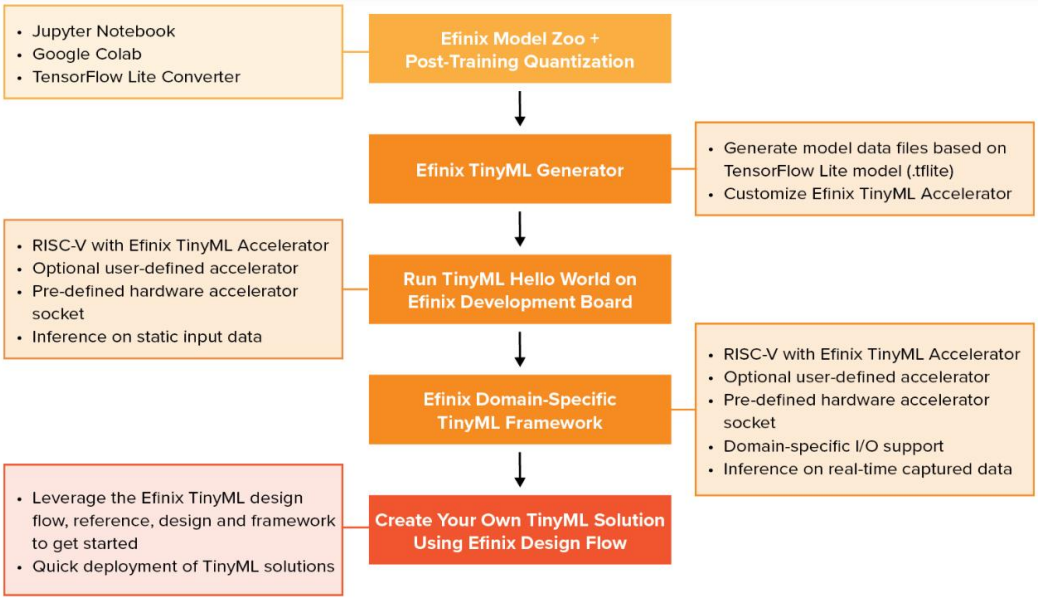

Figura 40 – Flujo de diseño Efinix

Tomado de: TinyML platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

Efinix brinda una línea de diseño de un extremo a otro que hace uso de implementaciones de TinyML en FPGA de Efinix. El flujo de diseño incluye los aspectos relevantes del diseño, desde el entrenamiento del modelo de redes neuronales con el uso de inteligencia artificial hasta la cuantificación del entrenamiento y la ejecución de inferencia en RISC-V. Efinix también brinda un acelerador TinyML personalizado que se adapta a los tipos de proyectos a implementar, a su vez También proporciona instrucciones para implementar TinyML en el marco flexible de Efinix. La información sobre el flujo de diseño que maneja Efinix para el diseño y despliegue de modelos de aprendizaje automático para TinyML, se puede observar en la Figura 40 – Flujo de diseño Efinix.

3.3.1 Analisis de Efinix

Efinix aborda todos los obstáculos potenciales al asumir el reto de que las tarjetas de desarrollo FPGA estén disponibles para el desarrollo de modelos de inteligencia artificial mediante TinyML, de forma intuitiva y fácil para programadores poco experimentados. Mediante el núcleo RISC-V Sapphire el cual es configurable en su totalidad por el programador mediante el uso de la GUI de Efinity; De esta forma, los programadores no tienen que conocer a fondo el lenguaje de programación VHDL el cual es el protagonista de la aplicación de RISC-V en las tarjetas FPGA, pudiendo así sacar provecho de manera efectiva a toda la programabilidad de lenguajes de programación C/C++. Esto permite a los grupos de investigación crear aplicaciones en bajos tiempos de respuesta mediante algoritmos implementados en software Efinix el cual se caracteriza por ser un aplicativo de alta velocidad de creación de modelos. Todos los periféricos y equipos requeridos pueden ser especificados, configurables y permiten crear instancias mediante el núcleo Sapphire, con el fin de brindar un SoC que pueda ser modificado a su 100%. Esta capacidad RISC-V cuenta con la capacidad de combinar múltiples núcleos (hasta cuatro núcleos) y capacidad de funcionar en modo individual, lo que ofrece un nivel de procesamiento superior en cuanto a rendimiento se trata para la aplicación en placas de desarrollo FPGA, así mismo cuenta con la capacidad de ejecutar aplicaciones en sistemas operativos Linux. La aceleración de hardware se simplifica

mediante la segmentación del hardware y software; Una vez que el programador del sistema realiza los ajustes necesarios del algoritmo programado en el software, se puede iniciar con la aceleración dentro de la placa de desarrollo FPGA mediante Efinix.

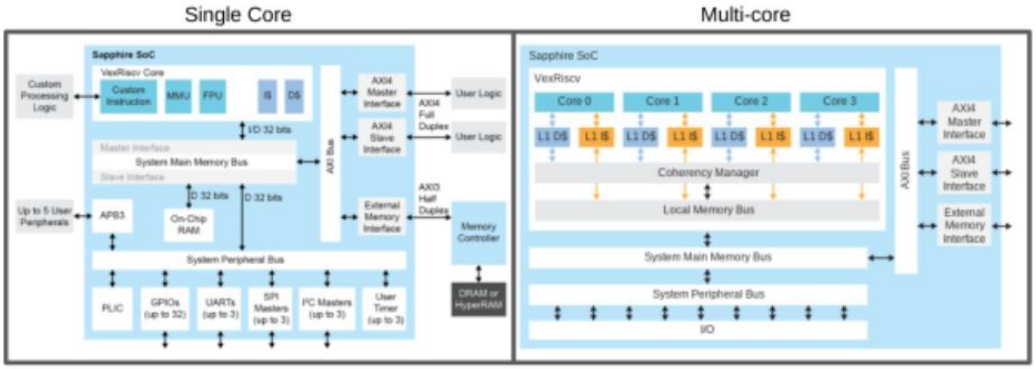

Figura 41 – Configuración de núcleo Sapphire RISC-V

Tomado de: TinyML Platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

La GUI (Interfaz gráfica de usuario) de Efinity permite a los programadores configurar el núcleo Sapphire RISC-V junto con todos los periféricos y dispositivos requeridos en lenguajes de programación familiares para un SoC completamente configurado. Esta configuración se puede expandir hasta cuatro núcleos Sapphire RISC-V. esta información se puede observar en la Figura 41 – Configuración de núcleo Sapphire RISC-V.

La Arquitectura del núcleo Sapphire RISC-V es única debido a que no todas las instrucciones están definidas. En cambio, se dejan algunas de estas para que los programadores las definan e implementen. Esto significa que puede crear unidades lógicas aritméticas (ALU) personalizadas que realizan funciones arbitrarias cuando lo solicitan, esto se le conoce como instrucción personalizada. Las instrucciones personalizadas definidas por el programador tienen la misma arquitectura que el resto de las instrucciones (por ejemplo, 2 registros de entrada, 1 registro de salida), dejando un total de 8 bytes de datos para procesamiento y 4 bytes devueltos al núcleo o núcleos RISC-V.

Sin embargo, la ALU está integrada en la placa de desarrollo FPGA y puede acceder a esta con el fin de realizar la extracción de información. Esto permite a los usuarios extender los datos más lejos de los 8 bytes, por otro lado, esto genera que la ALU sea se vuelva compleja, lo cual permite el acceso a información previamente compartida con la FPGA (como el acceso a los datos de un sensor de campo). Además, la aceleración de hardware, se puede utilizar la ALU, lo que da como consecuencia una mejora significativa en temas de velocidad de procesamiento en memoria. Efinix ha adaptado esta capacidad de entrenamiento personalizado a los proyectos de inteligencia artificial con su plataforma TinyML.

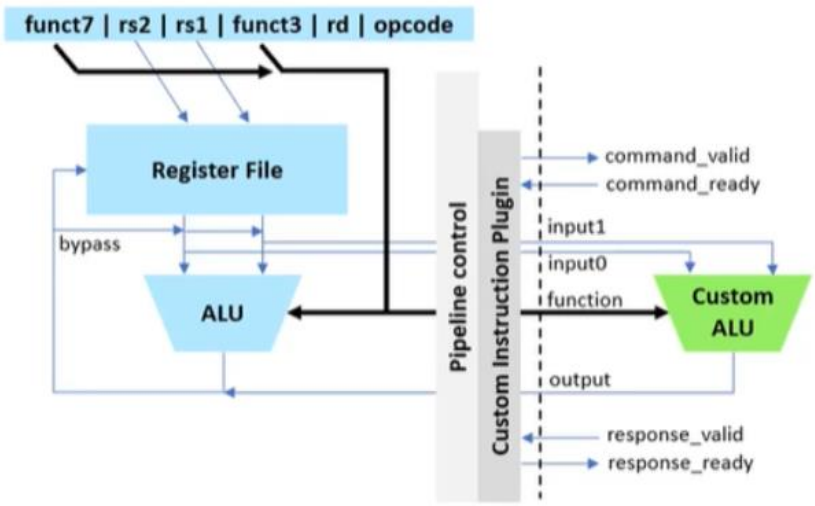

Figura 42 – Creación de ALU con RISC-V

Tomado de: Efinix (2023) How EFINIX is conquering the hurdle of hardware acceleration for devices at the edge, EE Times. Available at: https://www.eetimes.com/how-efinix-is-conquering-the-hurdle-of-hardware-acceleration-for-devices-at-the-edge

Como se observa en la Figura 42 – Creación de ALU con RISC-V, Se pueden crear ALU personalizadas con el núcleo Sapphire RISC-V, en el cual la configuración estándar incluye dos registros de origen (RS1 y RS2) de 4 bytes de ancho y un registro de destino (RD) de 4 bytes de ancho, esto es muy útil en proyectos de TinyML debido a que son elementos que cuentan con una baja carga de almacenamiento, por lo cual es eficaz la implementación de las ALU de este tipo en proyectos TinyML, con el uso de placas de desarrollo FPGA.

3.3.1.1 Aceleración de hardware con la plataforma TinyML

La plataforma TinyML de Efinix vuelve más veloz el proceso de aceleración de hardware, esto realiza mediante los cálculos utilizados en los modelos TensorFlow Lite y las cuales crean instrucciones personalizadas que optimizan su despliegue en aceleradores en la estructura de las FPGA.

Esto comprende los modelos de redes neuronales definidos por el software TensorFlow en el núcleo Sapphire RISC-V, aumentando su complejidad y su aceleración para que se ejecuten a velocidades de hardware que aprovechan las bibliotecas de código abierto perteneciente a TensorFlow Lite. Todo el flujo de trabajo de desarrollo se ha simplificado utilizando el flujo de herramientas Ashling, lo que hace que la configuración, creación y depuración de aplicaciones sea un proceso más sencillo de implementar.

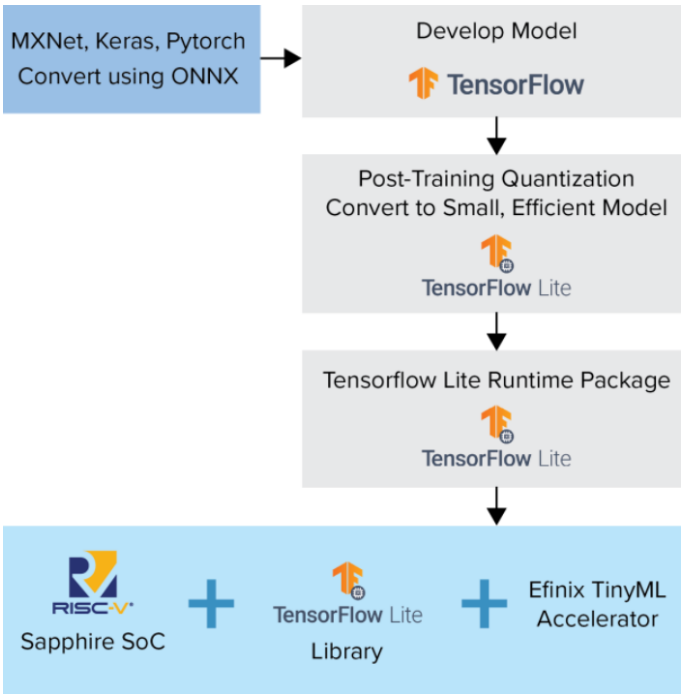

Figura 43 – Flujo de diseño de modelos Efinix TinyML

Tomado de: TinyML Platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

TensorFlow Lite genera una versión de los modelos estándar de tensorflow y aprovecha las funciones de las bibliotecas de este software para ejecutar mediante el núcleo Sapphire RISC-V, junto con el acelerador de Efinix TinyML tomando los modelos de bloques de aprendizaje automático, para extraer todo el potencial de las instrucciones personalizadas del núcleo Sapphire RISC-V y posteriormente realizar la aceleración en las placas de desarrollo FPGA.

La mayor parte de las librerías de instrucciones personalizadas de la plataforma de software Efinix TinyML son de libre acceso para los desarrolladores de proyectos e inteligencia artificial y estas se encuentran en Efinix GitHub como código abierto para implementar en El núcleo Efinix Sapphire, periféricos y todos los dispositivos requeridos para el diseño y desarrollo de proyectos de inteligencia artificial mediante modelos de redes neuronales. Por otro lado, La combinación de RISC-V, la estructura Efinix FPGA y las librerías disponibles en TensorFlow brindan la posibilidad de generar técnicas de aceleración que pueden ser divididas en los siguientes pasos:

- Paso 1: Ejecute el modelo TensorFlow Lite usando Efinity RISC-V IDE,
- Paso 2: utiliza el acelerador TinyML,
- Paso 3: acelerador de instrucciones personalizado definido por el usuario,
- Paso 4: Plantillas de acelerador de hardware.

3.3.1.2 TensorFlow lite usando Efinity RISC-V IDE

El "Paso 1" básicamente es un proceso estandarizado que a través de la GUI de Efinity los programadores pueden adquirir modelos de TensorFlow y desplegarlos mediante el software Efinix TinyML utilizando los núcleos del Sapphire RISC-V, sin necesidad de preocuparse por VHDL. Después del Paso 1, los programadores, en la mayoría de los casos, observarán que el rendimiento del algoritmo que están ejecutando no es óptimo y, por lo tanto, requiere aceleración.

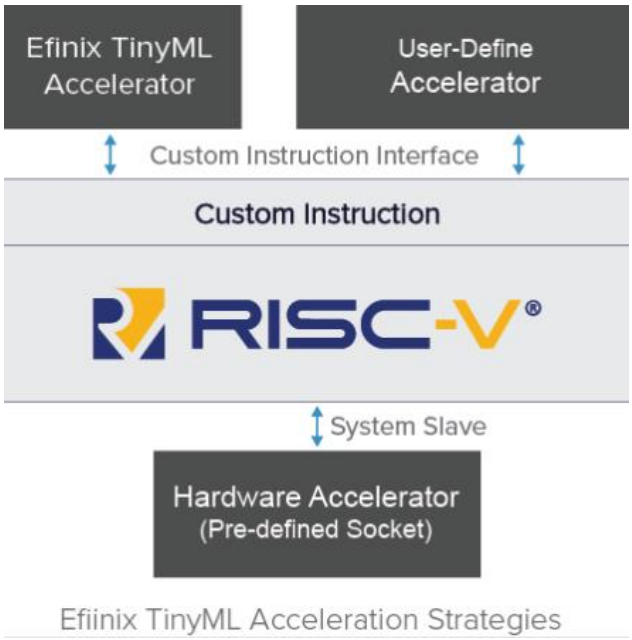

Figura 44 – Aceleración de hardware mediante Efinix TinyML

Tomado de: TinyML Platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

Como se observa en la Figura 44 – Aceleración de hardware mediante Efinix TinyML, Los usuarios de Efinix TinyML, puede combinar las entradas del modelo de red neuronal, en primer lugar, se utiliza como base el núcleo de RISC-V el cual al tomar modelos de TensorFlow Lite, se deberá acelera y optimizar, labor que los aceleradores de la plataforma Efinix toman el protagonismo.

3.3.1.3 Acelerador TinyML

El “Paso 2” facilita la creación y ejecución de instrucciones de convolución simple e instrucciones personalizadas de 2 registros de entrada y 1 registro de salida, mostrando una mejora de hasta 5 veces en la latencia del sistema. La latencia puede mejorar continuamente con el uso de instrucciones personalizadas que sean específicas para el aceleramiento a partir de funciones como, por ejemplo: ADD, MAXIMUM y MUL. Adicionalmente el “Paso 2” realiza la partición de hardware y software donde los programadores de los modelos de aprendizaje automático tienen la posibilidad de implementar bloques de aprendizaje perteneciente a los modelos de TensorFlow Lite, lo que se traduce en una interfaz intuitiva para la creación de instancias de instrucciones y con esto obtener modelos optimizados con una aceleración en masa para la ejecución de modelos en el núcleo Sapphire RISC-V.

3.3.1.4 Acelerador de instrucciones personalizado definido por el usuario

El “Paso 3” deja abierto a los programadores la posibilidad de crear sus propias instrucciones personalizadas, sin necesidad de utilizar plantillas de instrucciones que se encuentran en Efinix TinyML, lo que permite evolucionar constantemente las bases de diseño de proyectos en FPGA mediante este aplicativo Software y el uso de los núcleos de la RISC-V.

3.3.1.5 Plantillas de acelerador de hardware

El “Paso 4” sugiere que los elementos del modelo acelerados deberán ser incluidos dentro del proyecto de Efinix SoC con “zócalos” de aceleración. El zócalo del acelerador

cuántico permite a los programadores “dirigir” datos, recuperarlos y editarlos para generar bloques de información más pequeños o robustos.

El SoC Sapphire del núcleo RISC-V puede utilizarse para implementar un control general del modelo y del algoritmo a ejecutar, este último puede ser secuencial o flexible en su implementación. Como se ha mencionado el código implementado tanto en el hardware de la FPGA y el software de Efinix permite a los programadores optar por realizar cálculos en el procesador de la RISC-V o directamente en el hardware de la placa de desarrollo FPGA. En este tipo de plantilla de aceleración, el zócalo del acelerador de hardware predeterminado está conectado directamente con el controlador de la memoria DMA, al mismo tiempo que está conectado con una interfaz del SoC, con el fin de realizar transferencia y recepción de datos entre la CPU y el SoC, el cual puede emplear para realizar preprocesamientos o postprocesamientos de información. El controlador DMA agiliza la comunicación en serie de la memoria externa y los demás componentes del diseño del modelo de aprendizaje, mediante los siguientes aspectos:

- Almacenamiento de datos en la memoria externa.
- Envío y recepción de información hacia/desde el bloque de aceleración de hardware.
- Envío de información al sistema de postprocesamiento.

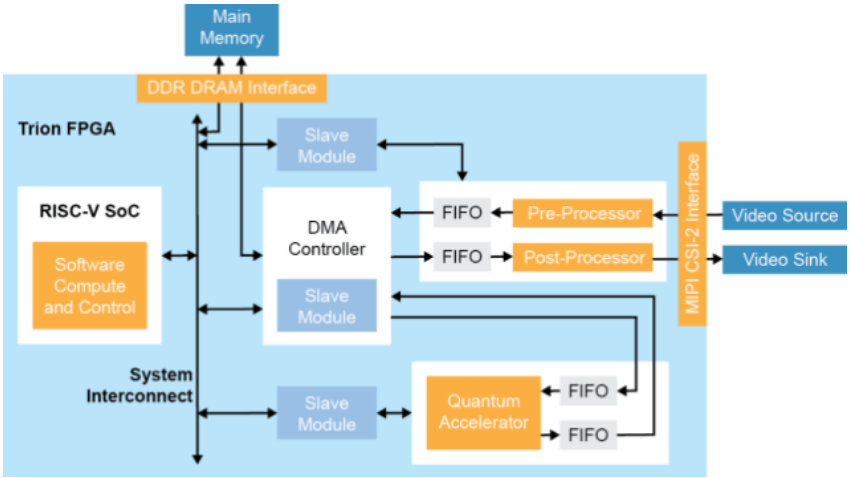

Figura 45 – Edge Vision Efinix TinyML

Tomado de: TinyML Platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

De acuerdo con la Figura 45 – Edge Vision Efinix TinyML, las plantillas de procesamiento y aceleración de hardware de Efinix TinyML, funcionan de manera bidireccional con los componentes básicos del diseño y a su vez funcionan de forma paralela generando procesos tanto en el controlador DMA, como en la RISC-V.

3.3.2 Placas de desarrollo Compatibles con Efinix TinyML

A diferencia de Edge Impulse y Tensor Flow, Efinix tiene un objetivo de compatibilidad de placas de desarrollo limitado, debido a que este aplicativo software para TinyML, solo se encuentra disponible para placas de desarrollo FPGA, las cuales son:

ITEM	PLACA DE DESARROLLO
1	Kit de desarrollo Titania Ti180 J484 - FPGA
2	Kit de desarrollo Titania Ti60 F225 - FPGA
3	Kit de desarrollo Trion T120 BGA576 - FPGA
4	Kit de desarrollo Trion T120 BGA324 - FPGA
5	Kit de desarrollo MIPI Trion T20 - FPGA
6	Kit de desarrollo Trion T20 BGA256 - FPGA
7	Kit de desarrollo Trion T8 BGA81 - FPGA
8	Placa de Desarrollo FireAnt T8 - FPGA

Tabla 4 – Dispositivos disponibles para uso de Efinix TinyML

Tomado de: Development Kits | Efinix, Inc. (s.f.). Efinix, Inc. https://www.efinixinc.com/products-devkits.html

Como se observa en la Tabla 4 – Dispositivos disponibles para uso de Efinix TinyML, La aplicación de software para TinyML, Efinix, cuenta con un sistema de desarrollo de inteligencia artificial enfocado solo en placas FPGA, por lo cual se ve limitado ante la expansión a otro tipo de sistemas y dispositivos, a comparación de Edge Impulse y TensorFlow, Efinix solo cuenta con 8 tipos de kits de desarrollo que son compatibles con las aplicaciones que se pueden modelar dentro del software Efinix TinyML.

Por otro lado, en este segmento no se hablará de los bloques de aprendizaje que Efinix puede tener a disposición del programador para la construcción de modelos de aprendizaje automático, debido a que como ya se ha mencionado en este documento anteriormente, Efinix adquiere las librerías de TensorFlow Lite, las cuentan con bloques de aprendizaje predeterminados para uso abierto de los usuarios de Efinix TinyML, por último, si se desea indagar sobre estos bloques de aprendizaje se deberá remitir al capítulo *Tipos de bloque de aprendizaje disponible en TensorFlow Lite* del presente documento, con el fin de retomar la información específica de cada uno de los bloques disponibles, cabe aclarar que al ser estos de código abierto, la plataforma software de TinyML Efinix, puede modelar redes neuronales a partir de todo el repertorio que TensorFlow Lite pone a disposición de los programadores para implementación de modelos de redes neuronales.

3.4 NANOEDGE AI STUDIO

NanoEdge AI Studio, es la herramienta de software que creó la compañía Cartesiam para aplicar las técnicas de TinyML en las placas de desarrollo ST32, pertenecientes a la misma empresa.

NanoEdgeAI Studio, es conocido por ser un ambiente de desarrollo para dispositivos portátiles con sistemas operativos Windows, Linux – Ubunto, este tipo de software de desarrollo para TinyML, cuenta con la característica de que cualquier programador que utilice este programa puede crear librerías para aplicación de TinyML mediante un entorno amigable e intuitivo. Como se ha mencionado con anterioridad, este tipo de programas suelen ser enfocados en aprendizajes automáticos para la detección de eventos, como anomalías, clasificación y regresión de imágenes, las cuales no solo se

pueden utilizar de forma individual, sino que a su vez se pueden combinar para generar una solución completa mediante inteligencia artificial.

Una de las cualidades de Studio (Abreviación para Nano Edge AI Studio), es la capacidad de desarrollar y buscar librerías para aplicación de modelos de TinyML, de acuerdo con las características de la placa de desarrollo a utilizar, lo cual optimiza y saca el mejor rendimiento de los microcontroladores de bajos recursos.

Con esta plataforma Sotfware para TinyML, se puede realizar aprendizajes iterativos con tiempos de respuesta de 30ms en microcontroladores como el ARM Cortex-M4, lo cual es un tiempo eficiente para aplicaciones de detección en tiempo real.

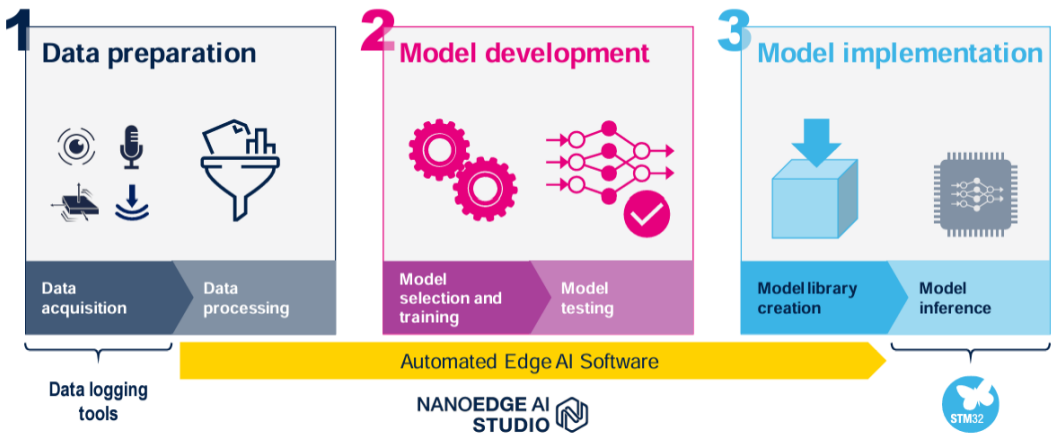

Figura 46 – Flujo de diseño NanoEdge AI Studio

Tomado de: Herramienta automatizada de aprendizaje automático (ML) para desarrolladores STM32. (s.f.). STM. https://www.st.com/en/development-tools/nanoedgeaistudio.html

Como se observa en la figura Figura 46 – Flujo de diseño NanoEdge AI Studio, Studio plantea una plataforma en donde se puedan adquirir datos de entrada para las redes neuronales, procesar esta información dentro de los modelos de aprendizaje automático, desarrollar el análisis de la información mediante las capas ocultas de los modelos de redes neuronales y utilización de librerías pre existentes en la plataforma o la posibilidad de crear las librerías a partir del modelo creado.

3.4.1 Análisis de NanoEdge AI Studio

NanoEdge AI Studio es un software proporcionado por ST para agregar fácilmente IA a cualquier proyecto integrado que se ejecute en cualquier brazo © Cortex M MCU.

Permite a los ingenieros integrados, incluso aquellos que no están familiarizados con la IA, encontrar casi sin esfuerzo el modelo de IA óptimo para sus necesidades a través de procesos sencillos.

Operado localmente en una PC, el software toma datos de entrada y genera una biblioteca de IA NanoEdge que incorpora el modelo, su preprocesamiento y funciones para una fácil integración en proyectos integrados nuevos o existentes.

La principal fortaleza de NanoEdge AI Studio es su benchmark que explorará miles de combinaciones de preprocesamiento, modelos y parámetros. Este proceso iterativo

identifica el algoritmo más adecuado adaptado a las necesidades del usuario en función de sus datos.

NanoEdge AI Library es una biblioteca estática de Inteligencia Artificial (IA) desarrollada originalmente por Cartesiam, para software C integrado que se ejecuta en Arm® Cortex ® microcontroladores (MCU). Viene en forma de un archivo. a precompilado que proporciona componentes básicos para implementar funciones inteligentes en cualquier código C. Cuando está integrada en microcontroladores, la biblioteca de IA NanoEdge les brinda la capacidad de "comprender" patrones de sensores automáticamente, por sí mismos, sin necesidad de que el usuario tenga habilidades adicionales en Matemáticas, Machine Learning o ciencia de datos.

Cada biblioteca estática de IA de NanoEdge contiene un modelo de IA diseñado para llevar capacidades de aprendizaje automático a cualquier código C, en forma de, por ejemplo, para aprender patrones de señales, detectar anomalías, clasificar señales o extrapolar datos. Funciones fácilmente implementables

Hay cuatro tipos diferentes de bibliotecas de IA NanoEdge, correspondientes a los cuatro tipos de proyectos que se pueden crear en NanoEdge AI Studio:

- Las bibliotecas Detección de anomalías (AD) se utilizan para detectar comportamientos anormales en una máquina, después de una fase inicial de entrenamiento in situ, utilizando un modelo dinámico que aprende patrones de forma incremental.
- Las bibliotecas clasificación de n clases (nCC) se utilizan para distinguir y reconocer diferentes tipos de comportamientos, anómalos o no, y clasificarlos en categorías preestablecidas, utilizando un modelo estático.
- Las bibliotecas Clasificación de 1 clase (1CC) se utilizan para detectar comportamientos anormales en una máquina, utilizando un modelo estático, sin proporcionar ningún contexto sobre las posibles anomalías esperables.
- Las bibliotecas Extrapolación (E) se utilizan para estimar un valor objetivo desconocido utilizando otros parámetros conocidos, utilizando un modelo estático (regresión)

Estas son las características más importantes de las bibliotecas de IA NanoEdge:

- ultra optimizado para ejecutarse en MCU (cualquier Arm® Cortex®-M)
- memoria ultra eficiente (1-20 Kbytes de RAM/memória flash)
- ultra rápido (inferencia de 1 a 20 ms en Cortex®-M4 a 80 MHz)
- inherentemente independiente de la nube
- ejecutarse directamente dentro del microcontrolador
- Se puede integrar en código/hardware existente.
- consume muy poca energía
- preservar la pila (solo asignación estática)
- no transmitir ni guardar datos
- no requieren experiencia en aprendizaje automático para su implementación

- Todas las bibliotecas de IA de NanoEdge se crean utilizando NanoEdge AI Studio.

Las bibliotecas de IA NanoEdge contienen una variedad de modelos de aprendizaje automático y cada uno de estos modelos se puede optimizar ajustando una amplia gama de hiperparámetros. Esto da como resultado una gran cantidad de combinaciones potenciales, cada una de las cuales está diseñada para un caso de uso específico (una biblioteca estática para cada combinación). Por tanto, se necesita una herramienta para encontrar la mejor biblioteca posible para cada proyecto.

NanoEdge AI Studio toma como parámetros de entrada del proyecto (como el tipo de MCU, RAM y tipo de sensor) y algunos ejemplos de señales, y genera la biblioteca NanoEdge AI más relevante. Esta biblioteca puede no estar entrenada (solo comienza a aprender después de integrarse en el microcontrolador) o estar previamente entrenada en Studio. En todos los casos, la biblioteca de IA NanoEdge puede inferir (detectar, clasificar, extrapolar) directamente desde el microcontrolador de destino.

La biblioteca de IA NanoEdge resultante es una combinación de tres componentes de software elementales:

- Algoritmo de preprocesamiento de señales (como FFT, PCA, normalización, reencuadre u otros),
- Modelo de aprendizaje automático (como kNN, SVM, redes neuronales, algoritmos de aprendizaje automático exclusivos de Cartesiam u otros),
- Hiperparametrización óptima para el modelo ML.

Cada biblioteca estática de IA de NanoEdge es el resultado de la evaluación comparativa de prácticamente todas las bibliotecas de IA posibles (combinaciones de tratamiento de señales, modelo ML y hiperparámetros ajustados), probado con los datos mínimos proporcionados por el usuario. Por lo tanto, contiene el mejor modelo posible, para un caso de uso determinado, dados los ejemplos de señales proporcionados como entrada.

El uso de NanoEdge AI Studio es un proceso iterativo por diseño: los usuarios pueden importar señales, ejecutar una evaluación comparativa, encontrar una biblioteca y comenzar a realizar pruebas. en menos de una hora.

Luego, dependiendo de los resultados obtenidos, se realizan cambios en los datos de entrada (calidad y/o cantidad de señales) y se reinicia el proceso, para obtener una mejor iteración de la Biblioteca NanoEdge AI.

En pocas palabras, NanoEdge AI Studio toma los datos del usuario como entrada (en forma de ejemplos de señales de sensores) y produce un archivo de biblioteca estática (.a) como salida. Este es un procedimiento iterativo sencillo y relativamente rápido. Sin embargo, Studio no proporciona ningún dato de entrada. El usuario necesita disponer de datos cualificados (de suficiente calidad y cantidad) para poder obtener resultados satisfactorios del Estudio. Estos datos pueden ser señales de sensores sin procesar o señales pretratadas y deben formatearse correctamente (ver más abajo). Por ejemplo, para la detección de anomalías en una máquina, el usuario debe recopilar ejemplos de señales que representen señales "normales". comportamientos en esta máquina, así como algunos ejemplos de posibles "anomalías". Este proceso de recopilación de datos es crucial y puede resultar tedioso, ya que se necesita cierta experiencia para diseñar la metodología correcta de adquisición y muestreo de señales, que puede variar

drásticamente de un proyecto a otro. Además, NanoEdge AI Studio no proporciona ningún código C listo para usar para implementar en su proyecto final. Este código, que incluye algunas de las funciones inteligentes de la biblioteca NanoEdge AI (como inicio, aprendizaje, detección, clasificación, extrapolación), debe ser escrito y compilado por el usuario. El usuario es libre de llamar a estas funciones según sea necesario e implementar todas las funciones inteligentes imaginables.

NanoEdge AI Studio se puede utilizar para generar bibliotecas de aprendizaje automático para diferentes tipos de proyectos, utilizando datos provenientes de uno o más sensores, posiblemente de diferentes tipos. Por lo tanto, es fundamental comprender qué tipo de proyecto crear para un caso de uso determinado y qué tipo de sensor.

3.4.1.1 Indicadores de desempeño

A continuación, se muestra la lista de indicadores secundarios involucrados:

- **Detección de anomalías**:
  - Precisión equilibrada (BA)
  - Margen funcional
  - RAM y requisitos de memoria flash
- **Clasificación de clase n**:
  - Precisión equilibrada (BA)
  - Exactitud
  - Puntuación F1
  - Coeficiente de correlación de Matthews (MCC)
  - Una medida personalizada que estima el grado de certeza de una inferencia de clasificación.
  - RAM y requisitos de memoria flash
- **Clasificación de 1 clase**:
  - Recordar
  - Una medición personalizada que tiene en cuenta el radio de la hiperesfera que contiene señales nominales y la recuperación obtenida en el conjunto de datos de entrenamiento.
  - RAM y requisitos de memoria flash
- **Extrapolación**:
  - R^2 (R-cuadrado)
  - SMAPE (Error porcentual absoluto medio simétrico)
  - RAM y requisitos de memoria flash

Los principales indicadores secundarios son "*Precisión equilibrada* " para la detección de anomalías y la clasificación de clase n, "*Recall*" para clasificación de 1 clase y "R^2" para extrapolación. Al igual que la puntuación, estas métricas se optimizan

constantemente durante la evaluación comparativa y se muestran a modo de información.

- Precisión equilibrada (detección de anomalías, clasificación) es la capacidad de la biblioteca para atribuir correctamente cada señal de entrada a la clase correcta. Es el porcentaje de señales que se han identificado correctamente.
- Recordar (clasificación de 1 clase) cuantifica el número de predicciones positivas correctas realizadas, de todas las predicciones positivas posibles.
- R^2 (extrapolación) es el coeficiente de determinación, que proporciona una medida de qué tan bien el modelo replica los resultados observados, en función de la proporción de la variación total de los resultados explicados. por el modelo.

Métricas adicionales relacionadas con las huellas de memoria:

- RAM (todos los proyectos) es la cantidad máxima de memoria RAM utilizada por la biblioteca cuando está integrada en el microcontrolador de destino.
- Flash (todos los proyectos) es la cantidad máxima de memoria flash utilizada por la biblioteca cuando está integrada en el microcontrolador de destino.

3.4.1.2 Herramientas integradas de NanoEdge AI Studio

Las Herramientas integradas en NanoEdge AI Studio para ayudar a la realización de un proyecto, son:

- El generador de registrador de datos, para crear código para recopilar datos en la placa de desarrollo con unos pocos clics
- El buscador de muestreo, para estimar la velocidad de datos y la longitud de la señal a utilizar en un proyecto en unos segundos.
- La herramienta de manipulación de datos para remodelar el conjunto de datos fácilmente

Figura 47 – Herramientas integradas a NanoEdge AI Studio

Tomado de: Automated Machine Learning (ML) tool for STM32 developers. (s.f.-a). ST. https://www.st.com/en/development-tools/nanoedgeaistudio.html

En la Figura 47 – Herramientas integradas a NanoEdge AI Studio, las herramientas integradas a NanoEdge, facilita la lectura de los datos obtenidos en los proyectos asociados a TinyML, a este tipo de herramientas se puede acceder desde la barra vertical izquierda mientras se está trabajando en el modelo de red neuronal.

3.4.1.3 Buscador de muestreo

La tasa de muestreo y el tamaño de muestra son dos parámetros que tienen un impacto considerable en los resultados finales proporcionados por NanoEdge. Una elección

incorrecta puede conducir a resultados muy pobres y es difícil saber qué frecuencia y tamaño de muestreo utilizar.

La herramienta está diseñada para ayudar a los usuarios a tomar decisiones informadas con respecto a la frecuencia de muestreo y la longitud de los datos, lo que lleva a un análisis preciso y eficiente de señales de series temporales en aplicaciones de IoT. Se necesita conjuntos de datos continuos registrados con la mayor velocidad de datos posible. Las señales en un conjunto de datos deben tener solo una muestra por línea, es decir, por ejemplo, 3 valores por línea si se trabaja con un acelerómetro de 3 ejes.

la herramienta remodela estos datos para crear buffers de múltiples tamaños (de 16 a 4096 valores por eje).

Al crear los buffers, la herramienta omite valores para simular datos registrados con una frecuencia de muestreo más baja. El rango de frecuencias probado va desde la frecuencia base hasta la frecuencia dividida por 32.

Con todas las combinaciones de tamaños de muestreo y tasas de muestreo, la herramienta aplica algoritmos de extracción de características para extraer la información más significativa de los buffers. Trabajar con funciones en lugar de con todos los buffers permite ser mucho más rápido.

3.4.1.4 Tipos de proyectos en NanoEdge AI Studio

A continuación, se describen los cuatro tipos diferentes de proyectos que se pueden crear utilizando Studio, junto con sus características, resultados y posibles casos de uso:

3.4.1.4.1 Detección de anomalías (AD):

Caso de uso: detectar anomalías en los datos utilizando un modelo dinámico.

Entrada del usuario: ejemplos de señales que representan estados nominales y estados anormales (usados solo para selección de biblioteca).

Resultado del estudio: biblioteca de detección de anomalías no entrenada que aprende de forma incremental, directamente en el microcontrolador de destino.

Detección de anomalías es el único tipo de proyecto que genera una biblioteca de IA NanoEdge capaz de aprender ejemplos de señales in situ, una vez integrada en el microcontrolador. Todos los demás tipos de bibliotecas sólo infieren en el microcontrolador.

Esta característica proporciona bibliotecas de detección de anomalías gran adaptabilidad, ya que la misma biblioteca se implementa en diferentes dispositivos (posiblemente monitoreando máquinas o máquinas ligeramente diferentes). que operan en diferentes condiciones ambientales, o que son susceptibles a perturbaciones) es capaz de entrenarse de manera diferente (aprender diferentes conocimientos) para adaptarse al comportamiento específico de su máquina objetivo.

Esto también significa que las bibliotecas de detección de anomalías pueden aprender de forma incremental sobre la marcha; el conocimiento se puede borrar, pero también se puede enriquecer en cualquier momento, simplemente aprendiendo ejemplos de señales adicionales que representen los nuevos comportamientos a aprender (por ejemplo, señales que representen nuevos regímenes nominales, posiblemente debido a un cambio en las condiciones operativas).

En proyectos de detección de anomalías (únicamente), el sensor Multisensor se utiliza para monitorear los "estados" de la máquina. que normalmente evolucionan lentamente en el tiempo. Estos estados pueden representarse mediante variables provenientes de fuentes de sensores distintas y/o el resultado de la agregación de buffers de señales en características artificiales de nivel superior.

Aquí, el formato de entrada es diferente:

- Cada línea representa una única muestra (posiblemente multivariable) en lugar de una señal completa.
- El número de valores por línea (igual al número de variables por muestra) no tiene por qué ser una potencia de dos.
- Las líneas no son independientes, por lo que el orden sí importa (las líneas no deben mezclarse).
- Normalmente, hay muchas más líneas en el archivo de entrada en comparación con el archivo "normal". caso (no "Multisensor"), ya que ahora solo hay una muestra por línea, en lugar de muchas muestras por línea.

Para la detección de anomalías, la pauta general es concatenar todos los ejemplos de señales correspondientes a la misma categoría. en el mismo archivo (como "nominal"). Como resultado, los puntos de referencia de detección de anomalías se inician utilizando solo 2 archivos de entrada: uno para todas las señales regulares y otro para todas las señales anormales.

- Las señales regulares corresponden al comportamiento nominal de la máquina, correspondientes a los datos adquiridos por los sensores durante el uso normal, cuando todo funciona como se espera. Incluya datos correspondientes a todos los diferentes regímenes o comportamientos que desee considerar como "nominales". Por ejemplo, al monitorear un ventilador, es posible que necesite registrar datos de vibración correspondientes a diferentes velocidades, posiblemente incluyendo los transitorios.
- Las señales anormales corresponden a un comportamiento anormal de la máquina, correspondiente a los datos adquiridos por los sensores durante una fase de anomalía.

Las anomalías no tienen por qué ser exhaustivas. En la práctica, es imposible predecir (e incluir) todos los diferentes tipos de anomalías que pueden ocurrir en su máquina. Simplemente incluya ejemplos de algunas anomalías que ya haya encontrado o que sospeche que podrían ocurrir. Si es necesario, no dude en crear "anomalías" manualmente.

3.4.1.4.2 Clasificación de 1 clase (1CC):

Para la clasificación de 1 clase, la idea es generar un único archivo que contenga todos los ejemplos de señales correspondiente a la clase única que se va a aprender. Si esta clase única contiene comportamientos o regímenes distintos, deben estar todos concatenados en 1 archivo de entrada. Como resultado, los puntos de referencia de clasificación de 1 clase se inician utilizando 1 archivo de entrada único. Las bibliotecas de clasificación de 1 clase son especialmente útiles cuando los tipos de anomalías que pueden ocurrir en una máquina de destino son difíciles de predecir o cuando no se puede proporcionar ningún ejemplo de señal que represente posibles anomalías.

Las variables que caracterizan a la clasificación de 1 Clase (1CC) son:

- Entrada del usuario: ejemplos de señales que representan solo estados normales (utilizados tanto para la selección de bibliotecas como para el entrenamiento de modelos).
- Resultado del estudio: biblioteca de detección de valores atípicos previamente entrenada que infiere directamente en el microcontrolador de destino.

3.4.1.4.3 Clasificación de clase n (nCC):

Las bibliotecas de clasificación de n clases se pueden utilizar, por ejemplo, para determinar qué tipo de anomalía está ocurriendo en una máquina (entre muchas posibles anomalías predeterminadas), o para detectar qué es el régimen/tipo de comportamiento actual de un equipo que tiene diferentes modos de operación.

Algunas variables características de la clasificación de clase n (nCC) son:

- Para la clasificación de n clases, todos los ejemplos de señales correspondientes a una clase determinada deben reunirse en el mismo archivo de entrada.
- Si alguna clase contiene comportamientos o regímenes distintos, deben estar todos concatenados en 1 archivo de entrada para esa clase.
- Como resultado, los puntos de referencia de clasificación de n clases se inician utilizando un archivo de entrada por clase.
- Entrada del usuario: ejemplos de señales que representan todos los diferentes estados (clases) que se esperan (utilizados tanto para la selección de bibliotecas como para el entrenamiento de modelos). Biblioteca de clasificación previamente entrenada que infiere directamente sobre el microcontrolador de destino.

3.4.1.4.4 Extrapolación (E):

La Extrapolación es estimación de un valor objetivo desconocido utilizando otros parámetros conocidos, utilizando un modelo estático. Así mismo es el único tipo de proyecto que genera una biblioteca ML capaz de evaluar un número (predecir el valor de una variable continua, utilizando un modelo de regresión matemática). Todos los demás tipos de bibliotecas solo generan clases discretas. Los proyectos de extrapolación asocian los parámetros conocidos a sus valores objetivo (utilizado tanto para la selección de bibliotecas como para el entrenamiento de modelos). El aprendizaje automático de extrapolación maneja una biblioteca de regresión previamente entrenada que infiere directamente sobre el microcontrolador de destino.

Los modelos matemáticos utilizados en las bibliotecas de extrapolación de inteligencia artificial de NanoEdge son modelos de regresión (no necesariamente lineales). Por lo tanto, se aplica toda la información general y las reglas que rigen la regresión.

NanoEdge AI Studio utiliza archivos de entrada proporcionados en proyectos de extrapolación para encontrar la mejor biblioteca de extrapolación de NanoEdge AI posible para entrenarla.

3.4.1.5 Preparación de datos de entrada en NanoEdge AI Studio

En comparación con los enfoques tradicionales de aprendizaje automático, que pueden requerir cientos de miles de ejemplos de señales para construir un modelo, NanoEdge AI Studio requiere conjuntos de datos de entrada mínimos (tan solo entre 50 y 100 ejemplos de señales, según el caso de uso). Sin embargo, estos datos deben ser

cualificados, lo que significa que deben contener información relevante sobre los fenómenos físicos que se van a monitorear.

Por lo cual algunos parámetros que se deberán de tener en cuenta son:

- La frecuencia de muestreo corresponde al número de muestras medidas por segundo. Para algunos sensores, la frecuencia de muestreo puede ser configurada directamente por el usuario (como los sensores digitales), pero en otros casos (como los sensores analógicos), es necesario configurar un temporizador para intervalos de tiempo constantes entre cada muestra.
- La velocidad a la que se toman las muestras debe permitir que la señal se describa con precisión, o "reconstruya"; la frecuencia de muestreo debe ser lo suficientemente alta para tener en cuenta las rápidas variaciones de la señal. Por lo tanto, surge naturalmente la cuestión de elegir la frecuencia de muestreo:
    - Si la frecuencia de muestreo es demasiado baja, las lecturas están demasiado alejadas
    - Si la frecuencia de muestreo es demasiado alta, podría afectar negativamente a los costes, en términos de potencia de procesamiento, capacidad de transmisión o espacio de almacenamiento.

A continuación, se presentan consideraciones generales para el formato del archivo de entrada, que se aplican a todos los casos:

- Se aceptan archivos .txt /.csv
- solo valores numéricos (sin contar los separadores) y sin encabezados
- separadores uniformes en todo el archivo: single space, tab.
- valores decimales formateados con un punto (.) y no comas (,).
- más de 1 muestra por línea (excepción: multisensor)
- menos de 16 384 (214) valores por línea
- el mismo número de valores numéricos en cada línea
- un mínimo de 20 líneas por eje del sensor (por ejemplo, para un acelerómetro de 3 ejes: 60 líneas es un mínimo)
- menos de ~100.000 líneas en total (generalmente, entre 50 y 1.000 son más que suficientes)

Así mismo, existen algunas reglas de formato específicas, dependiendo principalmente del tipo de proyecto creado en NanoEdge:

- Los proyectos de detección de anomalías, clasificación de 1 clase y clasificación de n clases siguen las mismas reglas generales.
- Los proyectos de extrapolación tienen una particularidad, porque necesitan incorporar los valores objetivo a partir de los cuales extrapolar.
- El multisensor es una gran excepción y se aplica solo a casos específicos en proyectos de detección de anomalías.

- En NanoEdge AI Studio, las líneas se tienen en cuenta de forma independiente e iterativa, por lo que deben representar una instantánea significativa en el tiempo de la señal a estar procesado. Por lo tanto, es fundamental establecer una frecuencia de muestreo coherente y un tamaño de búfer adecuado.
- cada línea para representar un ejemplo de señal único e independiente, compuesto de muchas muestras
- la frecuencia de muestreo de esta señal permanecerá constante durante todo el proyecto

3.4.1.6 Dispositivos Compatibles con NanoEdge AI Studio

Al igual que las plataformas software para tinyML que se han analizado en este documento NanoEdge AI Studio cuenta con una variedad de placas de desarrollo con las cuales se puede crear proyectos de redes neuronales aplicando TinyML, estas son las siguientes:

ITEM	PLACA DE DESARROLLO
1	STWIN SENSORTILE WIRELESS IND
2	NUCLEO-64 STM32F401RE EVAL BRD
3	NUCLEO-64 STM32F411RE EVAL BRD
4	NUCLEO-64 STM32G071RB EVAL BRD
5	NUCLEO-64 STM32G474RE EVAL BRD
6	NUCLEO-32 STM32L432KC EVAL BRD
7	NUCLEO-64 STM32L433RC EVAL BRD
8	NUCLEO-64 STM32L476RG EVAL BRD
9	STM32L4+ DISCOVERY KIT IOT NODE
10	DISCOVERY KIT WITH STM32L562ZE M
11	STM32 NUCLEO-64 DEVELOPMENT BOAR
12	STM32 NUCLEO-64 DEVELOPMENT BOAR
13	STM32 NUCLEO-64 DEVELOPMENT BOAR
14	STM32U5 DISCOVERY KIT FOR IOT

Tabla 5 – Dispositivos disponibles para uso de NanoEdge AI Studio

Tomado de: Automated Machine Learning (ML) tool for STM32 developers. (s.f.). St. https://www.st.com/en/development-tools/nanoedgeaistudio.html

En la Tabla 5 – Dispositivos disponibles para uso de NanoEdge AI Studio, se observa que NanoEdge AI Studio cuenta con 14 dispositivos compatibles con la plataforma software, a su vez al ser un producto proveniente de ST, la cual es una compañía referente en la industria de la automatización industrial, se entiende por qué solo las tarjetas STM32 son compatibles con esta plataforma de desarrollo de TinyML, es un caso similar al que ocurre con la plataforma Efinix TinyML, en donde el software se centraliza en un único tipo de placas de desarrollo con la diferencia de que NanoEdge AI Studio, cuenta con 6 referencias más con respecto a Efinix.

Studio está diseñado para poder trabajar con datos de sensores sin procesar que no necesariamente han sido preprocesados. Sin embargo, en los casos en que los usuarios ya tengan cierto conocimiento y experiencia sobre sus señales, se pueden importar señales preprocesadas.

Dependiendo del caso de uso del usuario, Studio necesita comprender qué formato de datos esperar en los archivos de entrada importados.

Hay 2 categorías principales de sensores seleccionables en Studio:

- Sensor genérico (n ejes), que es una generalización de algunos otros tipos de sensores, como acelerómetro de 3 ejes, magnetómetro de 1 eje, micrófono (1 eje), corriente (1 eje), y así sucesivamente. Este sensor cubre los casos de uso más típicos y se selecciona la mayor parte del tiempo.
- Multisensor (n variables), denominado "Multisensor", que es específico para proyectos de detección de anomalías.es completamente diferente, para los cuales el formato esperado casos de uso específicos, y diseñado para

3.4.1.6.1 Sensor Genérico

El sensor genérico del eje n (y todos los demás excepto el "multisensor") espera un buffer de datos como entrada, es decir, un ejemplo de señal representado por una sucesión de muestras instantáneas de sensores. Como resultado, este "ejemplo de señal" tiene una longitud temporal asociada, que depende de la frecuencia de muestreo (velocidad de datos de salida del sensor) y del número de muestras instantáneas que componen el ejemplo de señal (denominado tamaño del búfer, o longitud del búfer).

Este enfoque de sensor Genérico es el enfoque principal que se utilizará de forma predeterminada, ya que cubre todos los tipos (y combinaciones) de sensores posibles. todos los tipos de proyectos y la mayoría de los casos de uso.

Es especialmente relevante cuando los fenómenos físicos muestreados evolucionan "rápidamente" (como acelerómetro, sensor de corriente, etc.), utilizando velocidades de datos de salida superiores a unos pocos hercios (por ejemplo, de 10 Hz a 20000 Hz).

3.4.1.6.2 Sensor Multisensor

El sensor Multisensor, por otro lado, espera muestras instantáneas de datos. Este Multisensor, sólo disponible en proyectos de detección de anomalías, se utiliza normalmente cuando los fenómenos físicos muestreados evolucionan "lentamente" en el tiempo (como temperatura, presión u otros), o cuando no dependen explícitamente del tiempo. Un caso de uso típico de este sensor es el monitoreo de "estados de máquinas" representado por señales que forman características de nivel superior, resultantes de la agregación de múltiples variables, posiblemente provenientes de múltiples sensores. Por lo tanto, estas señales representan estados instantáneos y no señales que evolucionan en el tiempo.

3.5 LIBRERIAS PARA TINYML

Al mismo tiempo que las plataformas software para desarrollar modelos de aprendizaje automático para TinyML, existen librerías asociadas a compañías como Microsoft o entornos de desarrollo como Arduino, que cuentan con bloques de aprendizaje predeterminados para la implementación de modelos de redes neuronales, en esta sección se hablará sobre algunas de las librerías que cuentan con información en línea.

3.5.1 Arduino machine learning

Los microcontroladores que se utilizan en las tarjetas de desarrollo Arduino, son básicamente sistemas autónomos, que utilizan un solo chip, el cual es de bajo costo, estos son los procesadores en los cuales se enfoca la tecnología de Arduino para asociar TinyML y placas Arduino, sin embargo en la actualidad Arduino ha encontrado la forma de competir con grandes corporaciones que están asociando sus placas de desarrollo a la tecnología de TinyML, mediante placas de desarrollo como la Arduino Nano 33 BLE, la cual cuenta con un microcontrolador ARM Cortex-M4, que a diferencia de los chips usados en equipos móviles o computadores en la actualidad es un microchip de bajos recursos, sin embargo para la aplicación de TinyML se puede utilizar sin problema alguno. Las razones por las que este tipo de placas se puede emplear para machine learning es debido a que cuenta con características como:

- Bajo costo
- Funcionamiento con dispositivos IoT
- Eficiencia
- Aprendizaje automático

Las librerías de machine learning implementadas en Arduino IDE, son librerías que permiten el desarrollo de redes neuronales las cuales se pueden trabajar de la mano de versiones de TensorFlow Lite, Edge Impulse las cuales se encuentran diseñadas para trabajar con placas de Arduino que acepten algoritmos de inteligencia artificial y cuenten con sensores y microprocesadores con las características de TinyML.

3.5.2 librería de aprendizaje embebido - Microsoft

La biblioteca de aprendizaje embebido de Microsoft, También denominada ELL, es un proyecto de la compañía de código abierto, el cual fue diseñado para facilitar el desarrollo de modelos de aprendizaje automático en diferentes periféricos IoT, mediante TinyML.

Este sistema, es compatible con modelos de aprendizaje automático que cumplan con las características de redes neuronales abierta, generando un modelo como archivo .ell, en el cual se puede generar código ejecutable en dispositivos finales como pueden ser dispositivos móviles con sistemas operativos compatibles como Android o Linux, así mismo este tipo de librerías se puede ejecutar mediante código fuente C, C++, Python y Json, los dos últimos deberán ser importados a modelos con código fuente Python o Json.

3.5.3 Pytorch

Pytorch Mobile nace de la necesidad de ejecutar modelos de aprendizaje automático en dispositivos de campo con el fin de reducir las demoras en transferencia de datos, aumentar la privacidad de los servicios y el uso interactivo de estos elementos en diferentes soluciones tecnologías de la mano de IoT y la Inteligencia artificial.

Pytorch Mobile es una plataforma software que aplica TinyML en dispositivos periféricos, la cual permite implementar modelos de redes neuronales desde su entrenamiento hasta su implementación sin problema alguno, esto se menciona debido a que la

plataforma es una versión beta, sin embargo, proporciona un flujo ininterrumpido de trabajo desde el inicio hasta el fin del entrenamiento.

Pytorch Mobile simplifica el ambiente de investigación y producción para proyectos con prospección de despliegue en dispositivos móviles, con sistemas operativos iOS, Linux y Android, proporcionando así API's que solventan las de preprocesado de datos e integración necesaria de estos en periféricos móviles, incorporándolos de forma óptima a TinyML.

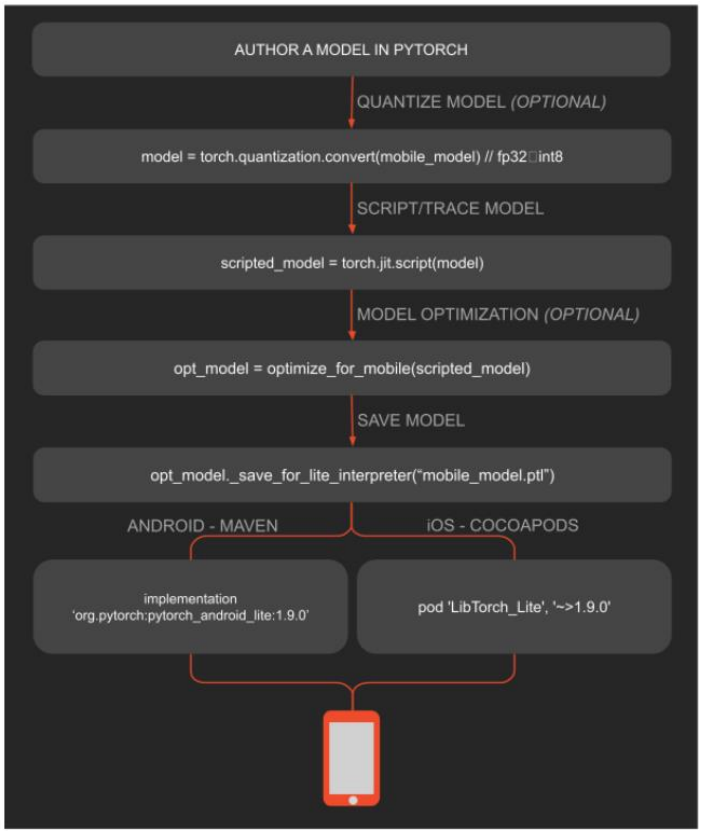

Figura 48 – Flujo de diseño Pytorch Mobile

Tomado de: Home. (s.f.). PyTorch. https://pytorch.org/mobile/home/

Mediante Pytorch el programador del modelo de red neuronal cuenta con la posibilidad de realizar implementaciones lineales sin la necesidad de ser un experto e programación de inteligencia artificial, el aplicativo software de tinyML está diseñado para ser un programa intuitivo y paso a paso realizara la conversión de los modelos, la optimización y el guardado de estos, a su vez para los diferentes sistemas operativos que pueden asociarse mediante dispositivos móviles, Pytorch cuenta con funcionalidades exclusivas para la implementación y despliegue de proyectos de modelos de redes neuronales basados en dispositivos de bajos recursos TinyML. Esta información se puede observar en la Figura 48 – Flujo de diseño Pytorch Mobile.

3.5.4 Utensor

Utensor es una librería de aprendizaje automático con cualidades como baja latencia, bajo consumo de almacenamiento y memoria, el cual fue diseñado por TensorFlow, con el objetivo de ser incluido en dispositivos ARM para su optimización.

Esta biblioteca cuenta con tiempos de ejecución mínimos y es una herramienta que se puede manejar offline, cuya parte da una mayor privacidad a los usuarios de este tipo de librerías, adicional esta librería es de uso libre y su código se encuentra disponible en Github para descarga, esta librería contiene un repositorio con implementaciones de ejemplo de modelos de redes neuronales que apliquen bloques de aprendizaje automático enfocados en placas de bajos recursos

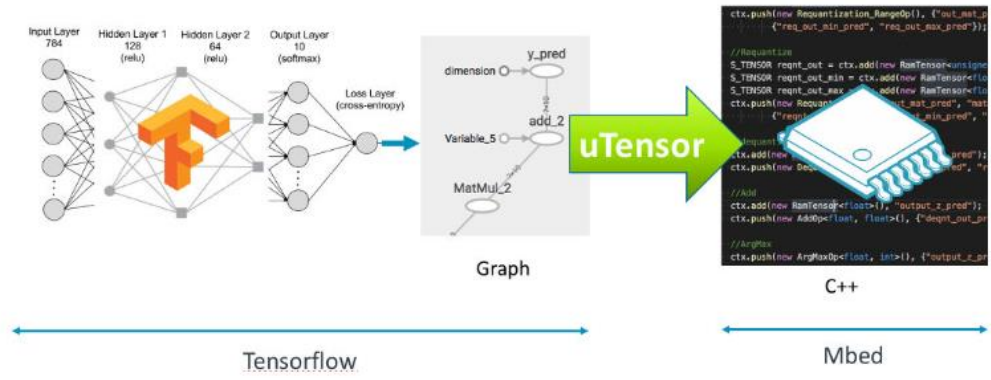

Figura 49 – Flujo de Implementación Utensor

Tomado de: GitHub - uTensor/uTensor: TinyML AI inference library. (s.f.). GitHub.
https://github.com/uTensor/uTensor

De acuerdo con la Figura 49 – Flujo de Implementación Utensor, se debe construir y entrenar un modelo en el aplicativo de software para TinyML, TensorFlow, el cual se tomará como base para Utensor, este último adquiere el modelo y genera archivos .cpp y .hpp que contienen el código fuente en leguaje de programación C++, el cual es necesario para la inferencia del modelo de red neuronal.

3.6 COMPARATIVA FINAL

Para completar el proceso de comparación y análisis de las plataformas de software con funciones para TinyML, se procederá a realizar una comparativa mediante aspectos relevantes que se encontraron a lo largo del análisis realizado para cada aplicativo software, de tal forma que se pueda interiorizar los puntos a tocar y evaluar de forma critica los siguientes aspectos:

3.6.1 Interfaz gráfica

3.6.1.1 Edge impulse

A diferencia de las otras aplicaciones analizadas en este documento, Edge Impulse cuenta con una interfaz gráfica completa en donde el programador sin importar el nivel de conocimiento que tenga sobre Inteligencia Artificial o TinyML, puede moverse en el entorno del aplicativo de Software de TinyML con facilidad y más importante aún, comprendiendo las diferentes secciones con las que la aplicación cuenta, desde la creación de un modelo de red neuronal, la visualización de los datos, el emparejamiento de los dispositivos o hasta el tipo de bloque de aprendizaje que la plataforma tiene a disposición de los usuarios.

3.6.1.2 TensorFlow Lite

De la misma forma que se maneja en Edge Impulse TensorFlow Lite, contempla una interfaz gráfica completa sin embargo, no es tan intuitiva como la de Edge Impulse en donde se puede ubicar con facilidad las herramientas de interés para la creación de modelos de redes neuronales, en el caso de TensorFlow Lite cuenta con pestañas que indican las herramientas de interés, sin embargo y a criterio del creador de este documento, se debe de haber trabajado con sistemas de programación de inteligencia artificial para entender que herramientas son las necesarias y como se debe modelar las redes neuronales y como ejecutar los bloques de aprendizaje que tensorFlow Lite ofrece como código abierto a los programadores y creadores de soluciones de TinyML.

3.6.1.3 Efinix

Efinix a diferencia de Edge Impulse y TensorFlow Lite, las cuales son plataformas software con su interfaz gráfica en una página web, Efinix es un aplicativo que se debe descargar al escritorio del ordenador de forma que se trabaja mediante una aplicación instalable y no desde la nube, sin embargo, este aspecto no es relevante con el entendimiento de la plataforma, por otro lado, lo que sí es relevante es la poco intuitivo que se presenta el aplicativo, debido a que no cuenta con guías claras en el despliegue inicial de la aplicación, presentándose así como una plataforma Software TinyML dirigida a programadores experimentados, ya que su complejidad no es apta para desarrolladores que deseen iniciar en el mundo del Tiny Machine Learning.

3.6.1.4 NanoEdge AI Studio

NanoEdge AI Studio, es la aplicación Software para TinyML más compleja de utilizar en cuanto a interfaz gráfico, debido a que así mismo como se presenta con Efinix, este aplicativo cuenta con una API instalable en escritorio, por lo cual se trabaja con recursos del ordenador en donde se desee realizar el modelo de la red neuronal y no con equipos conectados a la nube, como se presenta con TensorFlow Lite que cuenta con la ayuda de Colab para el despliegue de entrenamiento de modelos de redes neuronales o como Edge Impulse que utiliza sus propios recursos para generar el entrenamiento de los modelos de inteligencia artificial, por lo cual esta interfaz gráfica perteneciente a NanoEdge AI Studio presenta falencias debido a su complejidad, no es un aplicativo intuitivo, está diseñado para programadores expertos en el manejo de placas de desarrollo STM32

3.6.2 Desarrollo de prototipos

3.6.2.1 Edge impulse

Para el desarrollo de prototipos la plataforma Edge Impulse se presenta como la mejor opción para diseñar modelos de aprendizaje automático, no solo por su amplia variedad de bloques de aprendizaje automático predeterminados, listos para entrenamiento, sino por la facilidad de importar datos a esta plataforma de software TinyML, la cual es intuitiva y esta desarrollada para que programadores amateur o profesionales puedan realizar desarrollo de modelos de aprendizaje de forma rápida y sin mayor problema.

3.6.2.2 TensorFlow Lite

Para desarrollo de prototipos TensorFlow Lite a comparación de Edge Impulse, no solo se presta para diseñar modelos de aprendizaje automático en placas de desarrollo de bajos recursos, sino que a su vez, está enfocado en realizar implementación de aprendizaje automático en dispositivos móviles y dispositivos con sistemas operativos como iOS y Android, con el fin de desarrollo de API y no solo la creación de modelos de

aprendizaje profundo, por lo cual esta aplicación va más allá del simple hecho de analizar datos y entrenar modelos de redes neuronales, sino también aplicarlos en desarrollos móviles en dispositivos inteligentes de telefonía.

3.6.2.3 Efinix

Efinix no es una plataforma que sea enfocada directamente al desarrollo de prototipos de modelos de aprendizaje automático, sin embargo, este cuenta con la asociación del aplicativo con TensorFlow Lite, con el fin de expandir las herramientas de desarrollo de la plataforma, cabe aclarar que este panorama se ve limitado a las condiciones de compatibilidad que TensorFlow Lite tenga con las placas de desarrollo que Efinix TinyML maneja, las cuales son placas de desarrollo FPGA de bajos recursos, por lo que este tipo de características deberán ser explotadas en soluciones específicas con objetivos particulares.

3.6.2.4 NanoEdge AI Studio

El caso de NanoEdge AI Studio es similar a lo que se tiene con Efinix, sin embargo la plataforma software tiene un enfoque de desarrollo de prototipado basado en placas de desarrollo STM32, por lo cual cuenta con exclusividad de aplicaciones de modelos de aprendizaje debido a que en primer lugar esta es una plataforma paga, por lo que en la comunidad y soporte de esta plataforma es pequeño con respecto a las otras aplicaciones software para TinyML, por otro lado, al ser un aplicativo pago, se tiene muy poca información para realizar modelos de aprendizaje automático nuevos e innovadores y por último la complejidad de desarrollo de prototipos en NanoEdge AI Studio presume una complejidad grande debido a la programación de las placas de desarrollo STM, las cuales requieren un conocimiento basto en programación y en este microchips ARM.

3.6.3 Bloques de aprendizaje

3.6.3.1 Edge impulse

En el caso de los bloques de aprendizaje, Edge Impulse es una de las plataformas más completas de las plataformas software para TinyML, que se han analizado en este documento, debido a que maneja ocho tipos distintos de bloques de aprendizaje, los cuales varían entre clasificación de imágenes, detección de objetos, detección de sonidos, escritura y detección de anomalías en bases de datos, adicional Edge Impulse cuenta con una característica clave, la cual es que está enfocada en diferentes tipos de profesionales del machine learning, ya que cuenta con funcionalidades como modelos predeterminados con secciones de ajuste de parámetros sencillos o con la librería Keras, la cual es enfocada para el uso total de la red neuronal y el cambio de parámetros, mediante algoritmos de programación, esta función está enfocada en programadores de alto nivel de experiencia en machine learning e inteligencia artificial, cabe aclarar que este enfoque es exclusivo de Edge Impulse, lo que lo vuelve una plataforma de software para TinyML ideal para iniciar en los modelos de redes neuronales.

3.6.3.2 TensorFlow Lite

TensorFlow Lite es la plataforma de software para TinyML, que cuenta con la mayor cantidad de bloques de aprendizaje disponibles para el uso gratuito de programadores principiantes o expertos con un total de 19 bloques de aprendizaje predeterminados en los cuales se pueden desplegar soluciones de TinyML, en dispositivos iOS, Android y placas de desarrollo de bajos recursos, las cuales van desde raspberry pi hasta placas

arduino. TensorFlow Lite está enfocado en usuarios que requieren de la posibilidad de modificar completamente el código fuente de los bloques de aprendizaje provisionados por TensorFlow Lite, los cuales se encuentran en su mayoría en carpetas de Github, posibilitando el acceso a todas las librerías de los modelos de redes neuronales, sin embargo, una de las desventajas de TensorFlow Lite, es la dirección en que la plataforma de software para TinyML dirige su información, ya que se debe de tener un conocimiento solido en aplicaciones de redes neuronales y un enfoque preciso sobre como deberá usar cada uno de los modelos disponibles, debido a que TensorFlow Lite no es una plataforma destinada al aprendizaje desde cero de usuarios que deseen entrar en el campo de la inteligencia artificial mediante TinyML, sino que al contrario busca explotar conocimientos previos de cada uno de los usuarios y llevarlos a un nivel mayor mediante aplicaciones con librerías keras entre otras librerías que brindan la posibilidad de programar desde cero o hacer cambios avanzados en las configuraciones de los modelos de redes neuronales.

3.6.3.3 Efinix

En el caso de Efinix se cuenta con las mismas herramientas que TensorFlow Lite cuenta para el aprendizaje de modelos de redes neuronales, debido a que Efinix a diferencia de TensorFlow o Edge impulse, maneja la aplicación de modelos de aprendizaje automático en placas de desarrollo FPGA, sin embargo, los modelos de código abiertos dispuestos por tensorFlow Lite son optimizados para que funcionen en las placas de desarrollo que la plataforma software para TinyML – Efinix, de igual forma este tipo de segregación de placas de desarrollo presenta la desventaja de al ser un enfoque específico para FPGA, carece de tanto de información de despliegue de información en sistemas FPGA y su comunidad para la búsqueda de ayudas y foros online son limitados, por lo que Efinix es una plataforma con un alto nivel de complejidad en cuanto a la aplicación de bloques de aprendizaje, ya que aunque estos últimos tengan el mismo enfoque de despliegue de modelos de redes neuronales, la aplicación en FPGA es más compleja ya que se debe utilizar VDHL como lenguaje de programación de cada uno de los bloques, por lo que aumenta el nivel de dificultad de aplicación de TinyML, volviendo a Efinix una de las plataformas con mayor complejidad de uso para usuarios amateur, dando a entender que es dirigida a programadores expertos en el área del machine learning y FPGA.

3.6.3.4 NanoEdge AI Studio

Por último NanoEdge AI Studio es una plataforma que tiene solo cuatro enfoques de bloques de aprendizaje, los cuales son detección anomalías, clasificación de elementos de 1 clase, clasificación de n clases y extrapolación, por lo cual es la plataforma más limitada en cuanto a aplicaciones de bloques de aprendizaje automático, así mismo es la plataforma software que al igual que efinix está enfocada en un solo tipo de placas de desarrollo, las cuales son las STM32, sin embargo, a diferencia de Efinix, TensorFlow Lite y Edge Impulse, esta plataforma de software es completamente paga, por lo cual limita aún más el conocimiento y aplicación de modelos de redes neuronales, ya que al ser un sistema pago, la comunidad y foros que explican el manejo de los bloques de aprendizaje son diminutos, por lo que se debe referir a los manuales e instructivos que se puedan encontrar en la documentación dentro de las páginas oficiales de ST32.

3.6.4 Información en página

3.6.4.1 Edge impulse

En cuanto a información que se puede encontrar en la plataforma software, Edge Impulse a manera personal es la página que mejor modera la información que es pertinente para el uso de los bloques de aprendizaje predeterminados, el uso general y la asociación de dispositivos para la captación de datos que se utilizan para el aprendizaje automático de los modelos de redes neuronales, ya que es concisa y clara para el entendimiento y puesta en marcha de proyectos de inteligencia artificial. Por otro lado, Edge impulse al estar enfocada en usuarios de todo tipo de conocimientos es de las plataformas que mejor segrega la información por habilidades, ya que informa de la posibilidad de realizar despliegues de modelos de forma amateur mediante configuraciones predeterminadas, explicadas de forma que un programador iniciado en el área del machine learning pueda entender el concepto y los parámetros modificables o también brinda la opción de generar sistemas de despliegue en donde el programador experto pueda profundizar en el código fuente del modelo de red neuronal al punto de llegar a hacer cambio en bajo y alto nivel, dando flexibilidad al usuario de cambiar parámetros que en la mayoría de casos solo se ajustan para obtener información específica en despliegues de modelos particulares, todo esto de la mano de una base de información analizada y entregada en cada sección de la creación de modelos de aprendizaje automático.

3.6.4.2 TensorFlow Lite

TensorFlow Lite es una de las aplicaciones que si bien cuenta con mayor información disponible para que el usuario analice y verifique si los modelos de redes neuronales dispuestos como código abierto que se encuentran en la plataforma software de TinyML, se adaptan a las necesidades que los programadores necesitan cubrir, cuenta con un problema y es el de no filtrar la información que desea entregar a los usuario de esta plataforma, ya que no cuenta con el enfoque de brindar información seccionada por niveles de conocimiento, sino que al contrario, está enfocada en plasmar la mayor cantidad de información sobre las características de los modelos y no tiene presente el hecho de que es información difícil de comprender, otro aspecto a recalcar, es que la plataforma TensorFlow Lite, tiene todos sus bloques de aprendizaje en la plataforma GitHub, por lo cual para usuario que no cuenten con experiencia en el manejo de TensorFlow Lite, es de mayor complejidad el entendimiento que la plataforma brinda, por lo cual TensorFlow, Aunque cuente con la mayor cantidad de información sobre modelos de redes neuronales, no es la mejor plataforma para programadores que cuenten con poca experiencia en machine learning que deseen realizar modelos de aprendizaje automático para TinyML.

3.6.4.3 Efinix

En la plataforma Efinix se encuentra información en página resumida sobre el funcionamiento de los aceleradores de hardware para TinyML desarrolladas en FPGA, sin embargo, no hay información relevante sobre la implementación de los bloques de aprendizaje que se pueden implementar dentro de las placas de desarrollo FPGA, no se encuentra información sobre el entendimiento de cómo se puede aplicar los bloques de aprendizaje de TensorFlow Lite, dentro de Efinix TinyML, por otro lado, tampoco se ven bloques de información que den mayor entendimiento sobre como implementar la aceleración de hardware de las FPGA para utilizarlas de forma que puedan ser utilizadas para implementar modelos de aprendizajes automáticos, en plataforma solo se muestra una sección corta sobre la interacción de Efinix y TensorFlow Lite, por lo cual, Efinix es

la plataforma Software para TinyML, que cuenta con menos información en página para el despliegue de modelos de redes neuronales, por lo cual en este aspecto Efinix, es la plataforma que genera mayores inconvenientes en la aplicación de proyectos de TinyML en FPGA.

3.6.4.4 NanoEdge AI Studio

NanoEdge AI Studio, al ser una plataforma de pago cuenta con restricciones de información que se puede encontrar dentro de la plataforma de software para TinyML, sin embargo, la información que se encuentra dentro de las páginas oficiales de ST, está muy bien organizada, enfocando al usuario exactamente en el objetivo que la plataforma quiere, el cual es la aplicación de modelos de redes neuronales desplegadas en placas de desarrollo STM32, por lo cual, toda la información que se encuentra son referentes al uso de los bloques de aprendizaje automático que se pueden desplegar en estas placas de desarrollo, así mismo, la información sobre las placas que se pueden usar, las particularidades de cada una y los trabajos en los cuales se emplean con normalidad, también se encuentran en las páginas de ST, por lo cual, NanoEdge AI Studio a pesar de ser una plataforma de pago, cuenta con información concisa y organizada para que programadores de medio y alto conocimiento sobre despliegues de aplicaciones mediante machine learning, puedan entender de forma precisa como implementar bloques de aprendizaje automático, cabe aclarar que de manera personal, esta aplicación no es apta para dar los primeros pasos en el mundo de la inteligencia artificial.

3.6.5 Guía paso a paso

3.6.5.1 Edge impulse

Edge Impulse es la plataforma mejor posicionada en esta comparativa debido a que es una plataforma software enfocada al aprendizaje desde cero, debido a esto cada sección de la plataforma de software está explicada en su totalidad, dando al usuario la posibilidad de comprender cada acción que se realiza en el paso a paso de la configuración de las redes neuronales, cabe aclarar que este proceso inicia desde el log in de la aplicación, hasta que la plataforma realiza el aprendizaje y el despliegue del modelo de inteligencia artificial, por otro lado Edge Impulse, como ya se ha mencionado en este documento anteriormente está enfocado para programadores de todos los niveles de conocimiento por lo cual las guías paso a paso están diseñadas para que programadores expertos o programadores de bajo nivel de conocimiento en el machine learning tengan guías descriptivas y concisas de cómo aplicar y sacar el máximo provecho a las herramientas que la plataforma de software para TinyML tiene a disposición de forma gratuita.

3.6.5.2 TensorFlow Lite

Tensorflow Lite es una plataforma que no cuenta con específicamente una guía paso a paso de cómo manejar y poner en marcha los bloques de aprendizaje automático que tiene a disposición de los usuarios de la plataforma, en su lugar, TensorFlow Lite, se apoya en la comunidad y repositorio de GitHub, para generar códigos de uso abierto, para que los usuarios extraigan el contenido necesario de estos repositorios y los apliquen en cada uno de los proyectos que deseen crear, por ende, esta plataforma carece de tener explicaciones profundas de cómo realizar un modelo de aprendizaje automático desde cero, lo cual genera inconvenientes para los usuario que no tengan conocimiento en despliegue de estos proyectos de inteligencia artificial. Por otro lado, en la plataforma de GitHub, TensorFlow Lite pone a disposición de los programadores las herramientas necesarias para realizar proyectos de TinyML, sin embargo, son

documentos con explicaciones básicas que no profundizan en mayor medida como generar emparejamiento con dispositivos o como desplegarlos en plataformas como Colab, por lo que es una plataforma compleja de utilizar y requiere un nivel medio alto de conocimiento en machine learning y programación para su uso.

3.6.5.3 Efinix

La plataforma de Software Efinix, según esta comparativa es la herramienta para aplicación de proyectos TinyML, con peor desempeño en las guías de uso, por un lado al ser un aplicativo de escritorio cuenta con un sistema de guía paso a paso precario en donde solo explica los puntos más importantes, pero no indica un paso a paso de cómo realizar un proyecto de modelos de aprendizaje automático desde cero, por lo cual para el uso de esta herramienta el programador de la inteligencia artificial deberá de tener previo conocimiento de la plataforma de software, así mismo deberá de tener un nivel de programación en VDHL medio – alto, ya que las placas de desarrollo FPGA, no son los dispositivos más intuitivos del mercado, por lo cual esto representa un reto para usuario que deseen optar por el uso de Efinix como herramienta para realizar acercamientos al machine learning, esto último deja a Efinix como la herramienta menos recomendable para generar modelos de redes neuronales desde cero siguiendo guías entregadas por parte de los desarrolladores de la plataforma para TinyML.

3.6.5.4 NanoEdge AI Studio

En el caso particular de la plataforma de desarrollo software para proyectos de TinyML, NanoEdge AI Studio, se cuenta con varias guías detalladas de como generar proyectos de inteligencia artificial mediante bloques predeterminados y entregados por parte de ST, la cual es la compañía encargada de la administración de este aplicativo, que al igual que Efinix es una plataforma instalable de forma local en los computadores, sin embargo, a diferencia de Efinix o TensorFlow Lite, la plataforma NanoEdge AI Studio cuenta con una base de datos y con guías paso a paso detalladas para que los usuarios que si bien deben contar con una experticia mayor al promedio en programación de microcontroladores Cortex que utilizan las placas de desarrollo STM32, sean proyectos sencillos de implementar, por lo cual, la plataforma NanoEdge AI, solo por debajo de Edge Impulse es la segunda mejor aplicación de la comparativa realizada en este documento en cuanto a guías paso a paso se trata.

3.6.6 Nivel de dificultad

3.6.6.1 Edge impulse

Edge Impulse es de las plataformas Software para TinyML, que cuenta con la menor complejidad en cuanto al uso de esta, ya que como se ha mencionado con anterioridad Edge Impulse está enfocada en ser una plataforma para programadores con pocos conocimientos de la ejecución y despliegue de modelos de redes neuronales, por tal motivo este software, cuenta con buenas guías paso a paso, información en página y cuenta con un nivel de intuición alto, lo cual genera un mayor entendimiento a los usuarios sin importar el nivel de conocimiento en programas de machine learning o de programación en sí, por lo cual en esta comparativa se reconoce que Edge Impulse es la plataforma más sencilla, por otro lado, otro aspecto relevante es la conexión de dispositivos, el software para TinyML, cuenta con un dashboard fácil de entender en donde genera la conexión de dispositivos de forma fácil, con pocos de pasos a seguir y con compatibilidad con dispositivos móviles de distintos sistemas operativos y cuenta con gran variedad de placas de desarrollo de bajos recursos que son compatibles con Edge Impulse.

3.6.6.2 TensorFlow Lite

En el caso de TensorFlow Lite, aunque es una de las aplicaciones pioneras en el mundo del machine learning para Tiny, no es la plataforma de software con el nivel de complejidad más baja, ya que, en primer lugar esta no cuenta con un concepto de guías paso a paso definidas en general, cuenta con demasiada información sobre bloques de aprendizaje la cual no cuenta con un orden dirigido para que el usuario pueda realizar el despliegue de estos modelos de aprendizaje automático, sino que brinda demasiada información que puede saturar al lector, por lo cual, para el uso de TensorFlow Lite, se debe tener conocimientos previos en el uso de herramientas como Colab o de plataformas de gestión de información como GitHub, adicional del uso y programación de aplicaciones en Android Studio o iOS, esto sin contar que el nivel de conocimiento en programación debe ser medio – alto, por lo cual hace a TensorFlow Lite una aplicación más compleja de usar si se contrasta con Edge Impulse, sin embargo, se resalta que TensorFlow Lite si se cumple la mayoría de estos requisitos mencionados con anterioridad, se convierte en una plataforma con un nivel de dificultad baja, adicional a esto esta plataforma Software cuenta con una comunidad grande en donde se puede buscar información como guías paso a paso detalladas que pueden dar solución a las problemáticas que pueden tener los usuarios que no cuenten con un nivel alto de conocimiento en el área de TinyML.

3.6.6.3 Efinix

Efinix para TinyML, es una de las herramientas que cuentan con menor información para realizar proyectos de modelos de redes neuronales, por lo cual, Efinix es una de las aplicaciones con mayor complejidad de uso, en primer lugar, Efinix es una aplicación de escritorio, que aunque es un aplicativo de uso gratuito, no cuenta con explicaciones claras de cómo realizar despliegues de proyectos o de cómo realizar la aceleración de hardware mediante FPGA, elemento característico de la aplicación de Efinix para TinyML, adicional a esto en la página no se encuentran guías paso a paso claras de como generar proyectos ni tampoco se encuentra mucha información en foros, por lo cual genera un nivel de dificultad mayor si lo comparamos con plataformas software como Edge Impulse o TensorFlow Lite, así mismo al ser una plataforma Software enfocada en FPGA, genera un mayor nivel de dificultad sobre los demás aplicativos, debido a que FPGA utiliza lenguaje de programación VDHL, el cual es poco conocido y solo pocos programadores conocen y tienen un nivel aceptable para ejecutar programas en este tipo de tarjeta de desarrollo, por todo lo anteriormente mencionado, Efinix se cataloga como la plataforma con mayor nivel de dificultad para desplegar proyectos de modelos de aprendizaje automático, esto sin importar el nivel de conocimiento del programador.

3.6.6.4 NanoEdge AI Studio

Para NanoEdge AI Studio el nivel de dificultad que se percibe para esta plataforma de software de TinyML, se presenta en el conocimiento de las placas de desarrollo STM32, ya que estas cuentan con características especiales como la configuración de pines, configuraciones de reloj y demás parámetros que permiten realizar emparejamiento con sensores externos como opciones de conexión por I2C y demás características que son propias de estas tarjetas, adicional a esto al ser una plataforma de pago, cuenta con explicaciones en la aplicación, la cual así como se presenta en Efinix, es un aplicativo instalable en escritorio, sin embargo a diferencia de Efinix, esta plataforma Software cuenta con la documentación necesaria para que se genere un entendimiento más claro

de como iniciar desde cero el despliegue de proyectos de modelos de aprendizaje automático, por lo cual en este comparativo, NanoEdge AI Studio, se posiciona en tercer lugar de dificultad por debajo de Edge Impulse y TensorFlow Lite y por encima de Efinix para TinyML.

3.6.7 Comunidad y soporte

3.6.7.1 Edge impulse

La comunidad y soporte de la aplicación de software para TinyML Edge Impulse, es de las comunidades que mayores foros tiene, además las bases de datos como GitHub cuentan con secciones de preguntas y respuestas para problemas comunes que puede presentar Edge Impulse, por otro lado, el soporte de la misma plataforma al ser Gratuita, se pensaría que debe ser precaria, sin embargo Edge impulse se enfoca en toda la comunidad brindando soporte mediante sus bases de información que en la mayor parte de los casos cuenta con la documentación necesaria para solucionar incógnitas o problemas, sin embargo, debido a que la comunidad de TensorFlow, junto con su soporte es más grande, mismo caso que NanoEdge AI Studio, posiciona a Edge Impulse como la plataforma Software en esta comparativa como la tercera aplicación para TinyML con mejor comunidad y soporte por debajo de TensorFlow Lite y de NanoEdge AI Studio.

3.6.7.2 TensorFlow Lite

TensorFlow Lite es la mejor plataforma software para TinyML en cuanto a comunidad y soporte se trata, esto se debe a que la compañía creadora de esta herramienta es Google, por lo cual, se cuenta con muchos usuarios que utilizan los bloques de aprendizaje predeterminados que se encuentran en la plataforma, así mismo tiene bases de datos y foros como Github, en donde constantemente se sube información actualizada por parte de la misma plataforma generando soporte a los usuarios de TensorFlow, adicional a esto los mismo programadores que pertenecen a la comunidad de desarrollo de modelos de redes neuronales en TensorFlow Lite, aportan información de interés para la comunidad, con la solución de problemas o nuevos bloques de aprendizaje, por ende, TensorFlow Lite es la plataforma idónea para la investigación de modelos de redes neuronales.

3.6.7.3 Efinix

Efinix al ser una de las nuevas aplicaciones que a diferencia o TensorFlow Lite, no cuenta con el respaldo de compañías como Google o Edge Impulse que lleva más tiempo en el mercado, es una aplicación que a la fecha es poco reconocida en el área del machine learning, sin embargo cuenta con un factor crucial y es el objetivo de la plataforma que es el de implementar técnicas de TinyML, en placas de desarrollo FPGA, las cuales tienen una comunidad y soporte no solo en páginas como Intel o sitios oficiales de FPGA, sino que también cuentan con amplios foros de información para el diseño de proyectos en estos dispositivos, sin embargo, al no tener una comunidad sólida y que su soporte en página es difícil de encontrar, posicionan a Efinix para TinyML, en la última posición en la comparativa de plataformas softwares de este documento.

3.6.7.4 NanoEdge AI Studio

En cuanto a NanoEdge AI Studio, es la segunda plataforma en esta comparativa que cuenta con mejor comunidad y soporte den página, ya que, al ser una empresa con años de trayectoria en los mercados, los programadores que cuentan con el conocimiento adecuado para la implementación de proyectos con técnicas de TinyML mediante las

placas de desarrollo STM32, además, la plataforma NanoEdge AI Studio, cuenta con bases de datos en donde se encuentran guías paso a paso de como instalar el aplicativo en computadores, iniciar proyectos bajo los bloques de aprendizaje automático disponibles, conocimiento de las herramientas que se encuentran dentro de la plataforma, entre otros aspectos primordiales en el momento de iniciar un nuevo despliegue de modelos de redes neuronales, esto sumado a que la comunidad de ST, cuenta con foros en línea en donde se exponen nuevos proyectos con código abierto para libre uso, junto con experiencias y posibles soluciones a inconvenientes que puedan presentarse en el despliegue de soluciones bajo la técnica TinyML, por lo cual NanoEdge AI Studio se sitúa en segunda posición por debajo de TensorFlow Lite, en cuanto a comunidad y servicio de soporte se trata.

3.6.8 Costos

A continuación, se presenta la Tabla 6 – Costo de trabajo en plataformas Software para TinyML, en donde se observan los valores promedio del uso de las plataformas software para TinyML, esta investigación se realizó mediante conversaciones directamente sostenidas con agentes de las compañías de pago como Edge Impulse y NanoEdge AI Studio.

PLATAFORMAS	COSTOS
EDGE IMPULSE	Edge impulse es una plataforma gratuita para desarrolladores que quieren generar modelos de aprendizaje sin enfoques industriales, sin embargo, cuenta con una sección enfocado a empresas, por lo cual para este caso se realizó la investigación presupuestal sobre el valor mensual de obtener todos los beneficios de Edge impulse como, desarrollo de modelos de redes neuronales ilimitados, posibilidad de trabajo colaborativo de mínimo 10 personas, máximo 30 personas, soporte técnico por parte de la plataforma 24 horas, 7 días a la semana, entre otros beneficios con los que cuentan los planes premium de Edge Impulse, lo anteriormente mencionado tiene un valor de 250 USD al mes.
TENSORFLOW LITE	TensorFlow Lite es una plataforma gratuita para desarrollo de sistemas TinyML, por lo cual el costo de trabajar con la plataforma es de 0 USD al mes, por otro lado, las placas de desarrollo que son compatibles con TensorFlow Lite rondan valores desde los 55 USD.
EFINIX	Efinix es una plataforma Gratuita para desarrollo de sistemas TinyML, por lo cual el costo de trabajo de la plataforma es de 0 USD al mes, sin embargo, cabe mencionar que las placas de desarrollo que son compatibles con esta plataforma rondan valores desde los 230 USD.
NANOEDGE AI STUDIO	NanoEdge AI Studio es una plataforma que requiere una licencia de activación de usuario para el uso de la plataforma, esta licencia es de carácter vitalicio por lo cual tiene un costo que ronda los 175 USD por usuario,

	aunque actualmente esta licencia se encuentra gratis para usuarios nuevos.

Tabla 6 – Costo de trabajo en plataformas Software para TinyML

3.7 PRUEBA DE HARDWARE

Con el fin de realizar pruebas en las plataformas software en las que se puede generar proyectos bajo la técnica de TinyML, se optó por analizar cada una de estas y buscar una placa de desarrollo compatible con estas aplicaciones, así mismo se llegó a la conclusión que Edge impulse y TensorFlow Lite son las plataformas más reconocidas cuentan con la placa de desarrollo Arduino Nano 33 BLE Sense v2, como dispositivo en común, adicional Efinix al ser una plataforma que utiliza los modelos de aprendizaje automático que facilita TensorFlow Lite como código abierto y NanoEdge AI Studio es una plataforma que requiere licenciamiento, se realizó dos implementaciones de pruebas básicas de hardware en Edge Impulse y TensorFlow Lite con el fin de poner en práctica la técnica TinyML en dispositivos de bajos recursos.

3.7.1 Prueba de Hardware para TinyML en TensorFlow Lite

En primer lugar, se realizó una aplicación de bloques de aprendizaje automático de movimiento, el objetivo de esta red neuronal es la visualización de movimiento similares a latigazos y movimientos laterales, mediante el uso de los sensores acelerómetro y giroscopio con los que la placa Arduino Nano 33 BLE Sense v2, cuenta, por lo cual se realizó el siguiente código:

```
while (samplesRead < numSamples) {
    if (IMU.accelerationAvailable() && IMU.gyroscopeAvailable()) {
        // Leer datos de aceleración y giroscopio
        IMU.readAcceleration(aX, aY, aZ);
        IMU.readGyroscope(gX, gY, gZ);
        samplesRead++;      // Imprimir los datos en formato CSV
        Serial.print(aX, 3);
          Serial.print(','); // ... Repetir para todas las variables de aceleración y giroscopio
          Serial.println();
          if (samplesRead == numSamples) {  // Agregar una línea vacía si es la última muestra
          Serial.println();     }
          }
           }
```

Figura 50 – Código de captura de datos de sensores Arduino Nano BLE

En la figura Figura 50 – Código de captura de datos de sensores Arduino Nano BLE, se observa el código para captura de datos del acelerómetro y giroscopio, una vez finalizado esto, se debe copiar los datos obtenidos a un archivo .CSV para la implementación en el código de aprendizaje obtenido en TensorFlow Lite y el cual deberá ser ejecutado en la plataforma Colab perteneciente a Google, adicional a esto se deberán cargar los archivos a la plataforma Colab y ejecutar todas las líneas de código que se encuentran en esta plataforma.

Una vez terminadas de ejecutar cada una de las líneas de código del bloque de aprendizaje, la plataforma creará un fichero nuevo con el nombre *Model.h,* este ejecutable es un archivo el cual, se deberá ejecutar en la plataforma de Arduino.

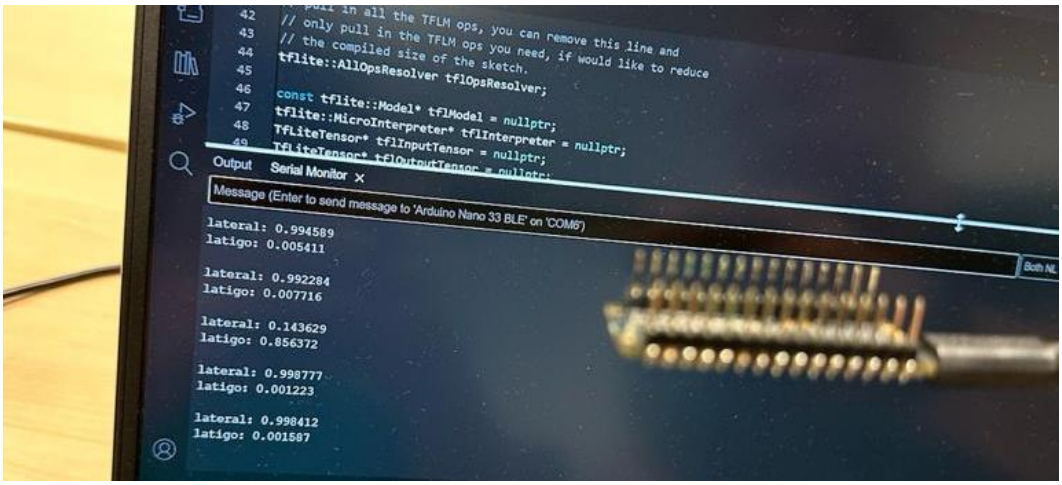

Figura 51 – Output de modelo de aprendizaje en TensorFlow Lite

Una vez se ejecuta el *model.h,* se deberá abrir un monitor serial para la visualización de los datos, estos deberán de mostrarse en pantalla como se observa en la Figura 51 – Output de modelo de aprendizaje en TensorFlow Lite. Si se realiza un movimiento lateral la plataforma dará valores entre 0 y 1, los cuales hacen referencia al porcentaje de que este movimiento sea un látigo o un movimiento lateral.

De esta forma se realiza la implementación de los bloques de aprendizaje de TensorFlow Lite, se realizó la implementación, desde cero y se observó que los únicos requisitos para la elaboración de este modelo de red neuronal es el conocimiento previo de programación en C++ para la captación de datos previos por parte de los sensores utilizados, el uso de la plataforma Arduino y conocimientos básicos del funcionamiento de la plataforma Colab, adicional a esto cada uno de los ejemplos que tiene TensorFlow Lite a disposición de los usuarios cuentan con todos los códigos abiertos para implementación de este estilo de proyectos, por lo cual TensorFlow Lite se entiende como una plataforma de nivel medio de dificultad para programadores de inteligencia artificial, así mismo se concluye que no es una plataforma intuitiva, ya que para el desarrollo del proyecto se debió investigar en la misma plataforma que herramientas son las necesarias para el desarrollo del ejercicio de Hardware.

3.7.2 Prueba de Hardware para TinyML en Edge Impulse

Por otro lado, la prueba de Hardware para TinyML que se desarrolló en Edge impulse contó con herramientas más didácticas e intuitivas para el desarrollo del proyecto de modelo de red neuronal, en este caso en particular se optó por utilizar el periférico de micrófono incorporado en la placa de desarrollo Arduino Nano 33 BLE sense v2, con el fin de realizar un ejercicio de bloques de aprendizaje automático para reconocimiento de voz.

En primer lugar, se debe realizar la conexión de la placa de desarrollo utilizada en el entorno de programación como se observa en la Figura 52 – Adquisición de datos en Edge Impulse.

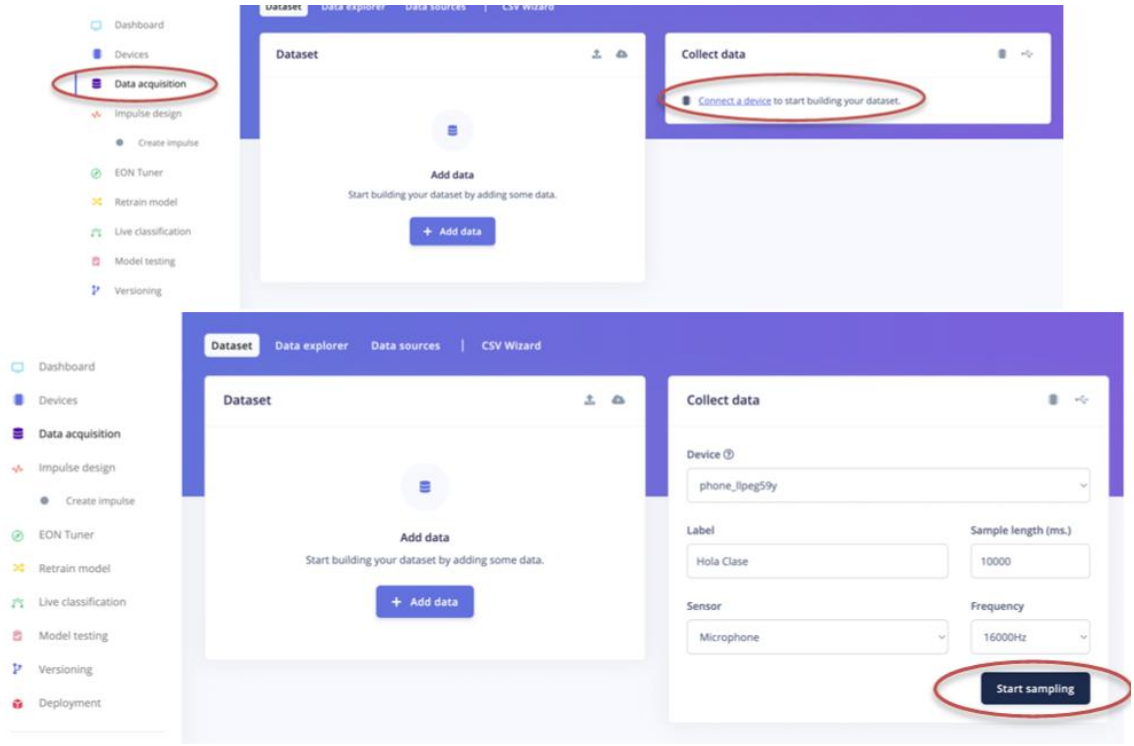

Figura 52 – Adquisición de datos en Edge Impulse

En este caso se debe de tener en cuenta que, para bloques de aprendizaje automático enfocados en la detección de sonidos, los datos deberán ser una recopilación de alrededor 5 a 10 minutos de pruebas para cada una de las secciones que se desea implementar.

Una vez adquiridos los datos se deberá subir el *Dataset,* al proyecto se deberá iniciar con el entrenamiento del modelo de red neuronal como se observa en la Figura 53 – Entrenamiento de la red neuronal.

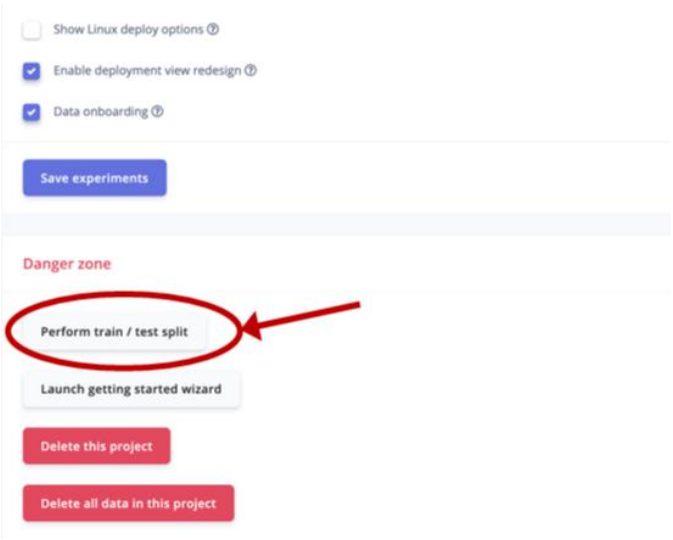

Figura 53 – Entrenamiento de la red neuronal

Este entrenamiento deberá ser de alrededor al 80% o superior, con el fin de que el modelo de red neuronal esté bien ajustado y pueda captar la probabilidad de los sonidos con exactitud.

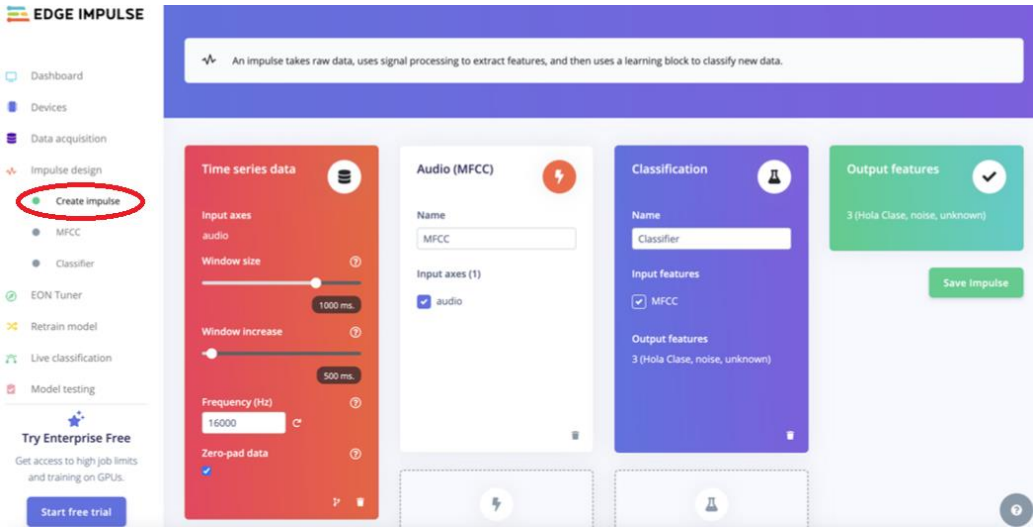

Figura 54 – Configuración de la red neuronal

Como se observa en la Figura 54 – Configuración de la red neuronal, En la sección *Create Impulse,* se deberá configurar los parámetros del bloque de aprendizaje, para este ejercicio son los que se muestran en la figura anterior.

Finalmente creado el impulso en la plataforma se deberá exportar el archivo *Model.h,* No sin antes visualizar que el modelo está correctamente configurado y el aprendizaje de este fue exitoso, tal y como se observa en la Figura 55 – Estado de aprendizaje del modelo.

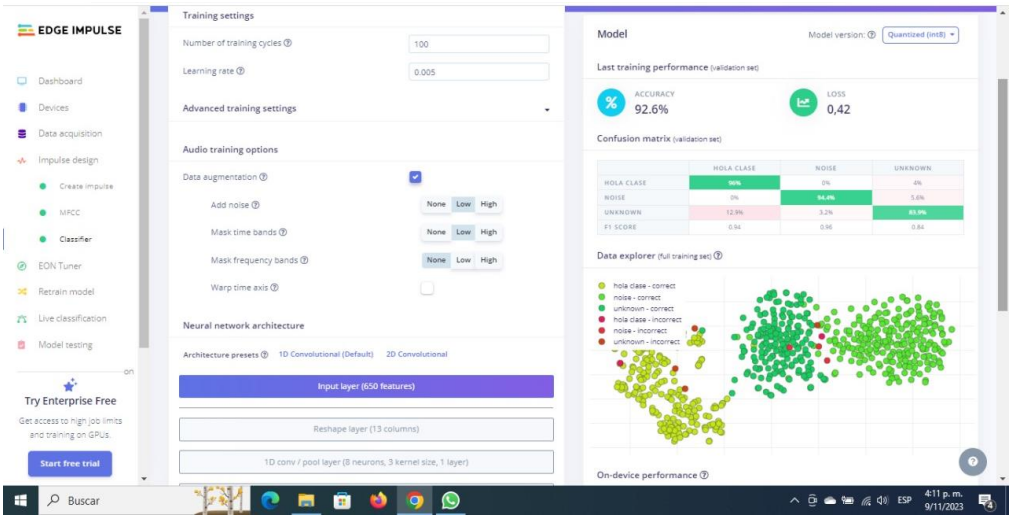

Figura 55 – Estado de aprendizaje del modelo

Una vez exportado el archivo *Model.h,* Se deberá cargar en la aplicación de Arduino y visualizar los datos mediante un monitor serial, si el ejercicio se realizó de forma exitosa

se deberá visualizar los datos observados en la Figura 56 – Datos obtenidos del modelo de aprendizaje.

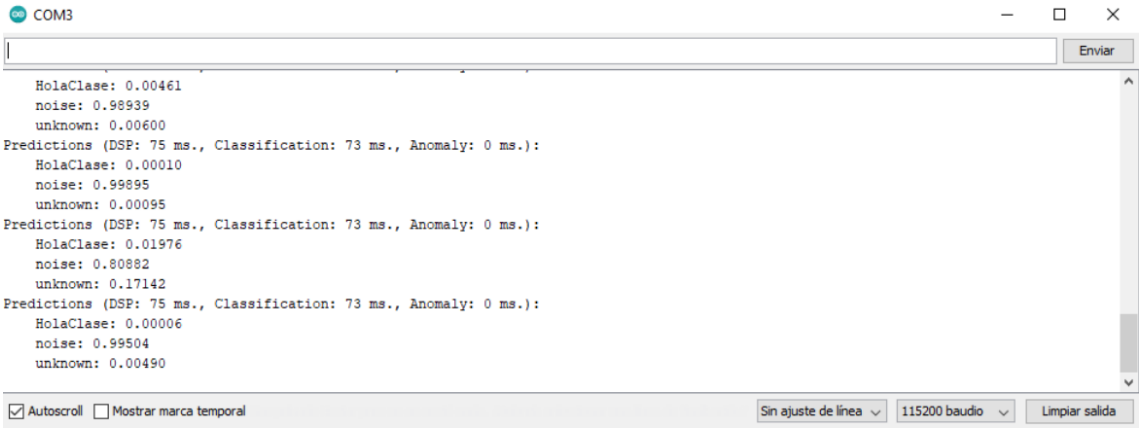

Figura 56 – Datos obtenidos del modelo de aprendizaje

Como Se puede observar en la Figura 56 – Datos obtenidos del modelo de aprendizaje, el modelo de aprendizaje de clasificación de audio puede captar sonidos como el ruido, palabras clave que se deseen captar y cuantificar o sonidos extraños e indistinguibles.

Por otro lado, la plataforma Edge Impulse es una plataforma intuitiva que ayuda al programador a seguir paso a pasos sencillos de entender, así mismo el emparejamiento y el sistema que la plataforma tiene para esta acción es sencillo e interesante, adicional la forma de visualización de los datos que la plataforma entrega es la mejor en la comparativa que se realizó en este documento, por lo cual Edge Impulse a nivel de implementación de modelos de redes neuronales cuenta con grandes ventajas no solo para programadores principiantes en el área del machine learning, sino que también es una herramienta que funciona para desarrolladores de alto nivel que busquen enfoques más precisos de los bloques de aprendizaje disponibles en la plataforma para su aplicación.

4 CONCLUSIONES

Este documento cuenta con las siguientes conclusiones sobre las comparativas y análisis realizados sobre las plataformas Software para TinyML:

- En cuanto al análisis realizado de las plataformas Software para TinyML, se concluye que TinyML, no es una técnica que haya avanzado a nivel documental, ya que si bien se puede encontrar información sobre las plataformas existentes, dentro de cada una de las páginas web y repositorios de las mismas plataformas software, no existe gran cantidad de documentación científica que hable sobre estas mismas, a su vez no se encontró documentación en repositorios de plataformas como IEEE, que hablen o comparen las plataformas visualizando así sus ventajas, desventajas o hablando de forma general de cada una de las plataformas analizadas en este documento, por lo cual esta comparativa deberá funcionar como base para el entendimiento y elección de plataformas software para TinyML, para usuarios que no cuenten con la suficiente experticia sobre el mundo de machine learning, a su vez este documento genera una visión más amplia de las redes neuronales, de los distintos dispositivos que se pueden emplear en proyectos de modelos de aprendizaje automático, así mismo como la elección de las plataformas que se están utilizando.
- Las plataformas que se analizaron en esta comparativa reflejan que las tecnologías actuales no se encuentran preparadas ni enfocadas en desarrollo de sistemas con base en TinyML, que si bien se ha indicado en múltiples foros y simposios que es una tecnología en auge y que tiene un futuro prometedor, en la actualidad Edge Impulse, TensorFlow Lite, Efinix para TinyML y NanoEdge AI Studio, son las plataformas que cuentan con la capacidad de generar proyectos de inteligencia artificial mediante TinyML, que aunque existan librerías que funcionen en diferentes dispositivos y puedan ser una solución óptima si no se desea utilizar las plataformas analizadas en esta comparativa, suelen contar con limitaciones como proyección de expansión, bloques de aprendizaje compatibles limitados, falta de soporte tanto en foros como en la comunidad en si, por lo cual el uso de estas librerías no es la opción favorable para el desarrollo de técnicas TinyML.
- AL realizar la comparativa y leer la documentación encontrada en foros, páginas web, artículos científicos y la información obtenida en cada uno de los repositorios de las plataformas, se evidencia que cada plataforma cuenta con un objetivo diferente y está orientado para que los programadores hagan uso de estas de acuerdo a intereses específicos, en el caso de Edge Impulse el enfoque principal se encuentra en la enseñanza y guía paso a paso de la creación de modelos de aprendizaje automático desde cero, a su vez está diseñado para que profesionales o principiantes puedan hacer uso de sus herramientas, para TensorFlow Lite está enfocado en programadores de nivel medio de conocimiento tanto en machine learning, como en programación en Python y análisis de bases de datos, por lo cual lo vuelve una herramienta que cuenta con el mayor número de bloques de aprendizaje a disposición de los usuarios para la implementación de modelos de redes neuronales, por lo cual lo vuelve una herramienta útil para cualquier caso, su enfoque principal se basa en el hecho de captar la mayor información posible y satisfaces las necesidades particulares

de cada proyecto mediante una amplia gama de bloques de aprendizaje, por otro lado Efinix simula el mismo concepto de TensorFlow Lite al apoyarse en los bloques de aprendizaje de código abierto disponible en la página principal y en la comunidad de Github en donde se encuentran todos estos repositorios, sin embargo, Efinix cuenta con un enfoque principal el cual es realizar la implementación de estos bloque de aprendizaje compatibles con las tarjetas de desarrollo FPGA y por último NanoEdge AI Studio simula el mismo concepto de Efinix con sus bloques de aprendizaje los cuales son limitados, pero que al tener una amplia gama de tarjetas STM32 compatibles con estos genera una comunidad solida con un sistema de soporte e información en repositorios que amplían la confianza en el uso de la plataforma software, por lo cual su enfoque principal es la implementación de modelos de redes neuronales específicamente en las tarjetas STM32.

- Como conclusión personal de la comparativa al realizar el análisis de cada una de las plataformas Software para TinyML y evaluando todos los factores descriptos en el capítulo 3.5 COMPARATIVA , la plataforma que es idónea tanto para principiantes como profesionales para realizar implementaciones y despliegues de redes neuronales es Edge Impulse, ya que cuenta con una amplia variedad de bloques de aprendizaje que dan solución a muchos problemas tanto de las empresas como de la vida cotidiana, así mismo como la amplia variedad de referencias de tarjetas de desarrollo de bajos recursos compatibles con el sistema, como la posibilidad de asociar dispositivos móviles a la toma de datos y modificación de los mismos, cuenta con una interfaz intuitiva y visualización de variables de interés, entre otros aspectos que posicionan a la plataforma Software para TinyML, como la mejor opción para la elaboración de proyectos de aprendizaje automático.

4.1 RELACIÓN DEL TRABAJO DESARROLLADO CON LOS ESTUDIOS CURSADOS

La relación que encuentro del trabajo de fin de máster realizado con los estudios cursados se encuentra principalmente en las asignaturas en donde su enfoque principal ha sido el IoT, Materias como Sistemas IoT inteligentes, la cual en mi caso despertó el deseo de realizar el documento de comparativas de plataformas software para TinyML, en la cual nos dieron a conocer plataformas como TensorFlow o Edge Impulse, que apoyaron a la creación de este documento, además fue el punto de inicio para obtener información sobre la existencia de otras plataformas que funcionan específicamente para el desarrollo de TinyML , por otro lado el uso de plataformas como STM32cube, la cual fue presentada en materias como IoT para la industria y Redes inalámbricas de sensores y actuadores, apoyaron al entendimiento de cómo funcionaba la plataforma NanoEdge AI Sudio y el reto que impone la realización de proyectos de inteligencia artificial mediante las placas de desarrollo STM32, así mismo la información y el conocimiento obtenido sobre las diferentes aplicaciones que tiene STM32 para proyectos de TinyML se basó en la búsqueda de información proveniente de repositorios oficiales de la plataforma Software por lo que el conocimiento previo de la plataforma fue esencial para obtener datos relevantes sobre NanoEdge AI Studio, por último el entendimiento del funcionamiento de las FPGA, es un conocimiento vital que se obtuvo en la asignatura sistemas avanzados de cómputo para la industria, por lo cual el indagar

y entender el funcionamiento de los aceleradores de hardware que utiliza Efinix en sus proyectos para TinyML mediante las placas de desarrollo FPGA fue más sencillo, por lo cual estas asignaturas impartidas en el Mater de ingeniería de computadores y redes han sido de utilidad no solo para la creación de este trabajo final, sino también para la aplicación y la comprensión de muchos temas relevantes que de no haberlos adquirido no hubiese sido posible concluir este documento de comparativas de plataformas software para TinyML. Por último, destacó la importancia de realizar trabajos practicas sobre estas plataformas, ya que gran parte del máster fue enfocado en IoT y esto generaría mayor comprensión del funcionamiento de tecnologías como lo es TinyML.

4.2 TRABAJOS FUTUROS

Algunos de los trabajos futuros que se pueden realizar a partir de este análisis y comparativa sobre las aplicaciones software existentes son los siguientes:

- Continuación de la comparativa de plataformas Software para TinyML, añadiendo librerías de inteligencia artificial como Pytorch.
- Implementación de proyectos basados en problemáticas reales utilizando las plataformas software para TinyML presentadas en este documento.
- Desarrollo de comparativas a nivel de Hardware sobre las plataformas Software presentadas en este documento.

5 BIBLIOGRAFIA

[1] C. S. Diego et al. “Tiny ML: La nueva revolución en la IoT”. Inici. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://roderic.uv.es/handle/10550/81527

[2] “¿Qué es el Aprendizaje mediante refuerzo? – Explicación del Aprendizaje mediante refuerzo – AWS”. Amazon Web Services, Inc. Accedido el 3 de diciembre de 2023. [En línea]Disponible: https://aws.amazon.com/es/what-is/reinforcement-learning/#:~:text=El%20aprendizaje%20por%20refuerzo%20(RL,utilizan%20para%20lograr%20sus%20objetivos.

[3] “¿Qué es el aprendizaje supervisado? | IBM”. IBM in Deutschland, Österreich und der Schweiz | IBM. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://www.ibm.com/es-es/topics/supervised-learning

[4] “¿Qué es el aprendizaje no supervisado? | IBM”. IBM in Deutschland, Österreich und der Schweiz | IBM. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://www.ibm.com/es-es/topics/unsupervised-learning

[5] “Qué es TinyML y beneficios que aporta el aprendizaje automático integrado - Blog de arsys.es”. Blog de arsys.es. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://www.arsys.es/blog/tinyml#Que_es_TinyML_y_cuales_son_sus_beneficios

[6] “¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS”. Amazon Web Services, Inc. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://aws.amazon.com/es/what-is/neural-network/

[7] “Edge Impulse”. Edge Impulse. Accedido el 3 de diciembre de 2023. [En línea]. Disponible: http://www.edgeimpulse.com

[9] “¿Qué es TensorFlow y para qué sirve?” ¿Qué es TensorFlow y para qué sirve? Accedido el 3 de diciembre de 2023. [En línea]. Disponible: https://www.incentro.com/es-ES/blog/que-es-tensorflow

[10] “Libraries - Arduino Reference”. Arduino - Home. Accedido el 4 de diciembre de 2023. [En línea]. Disponible: https://www.arduino.cc/reference/en/libraries/

[11] “TensorFlow Lite para microcontroladores”. TensorFlow. Accedido el 9 de diciembre de 2023. [En línea]. Disponible: https://www.tensorflow.org/lite/microcontrollers?hl=es-419#:~:text=Es%20compatible%20con%20las%20siguientes,Kit%20de%20desarrollo%20STM32F746

[12] Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., & López, O. L. A. (2023). TinyML: Tools, applications, challenges, and future research directions. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-16740-9

[13] Reinforcement learning technique for application placement in edge and fog computing environments,” IEEE Transactions on Mobile Computing, p. 1–1, 2021.

[14] G. Muhammad and M. S. Hossain, “Emotion recognition for cognitive edge computing using deep learning,” IEEE Internet of Things Journal, vol. 8, no. 23, p. 16894–16901, Dec 2021.

[15] W. Li, W. Deng, R. She, N. Zhang, Y. Wang, and W. Ma, “Edge computing offloading strategy based on particle swarm algorithm for power internet of things,” in IEEE 2nd International Conference on Big Data, Artificial Intelligence, and Internet of Things Engineering (ICBAIE). IEEE, Mar 2021, p. 145–150.

[16] J. Liu, C. Liu, B. Wang, G. Gao, and S. Wang, “Optimized task allocation for iot application in mobile-edge computing,” IEEE Internet of Things Journal, vol. 9, no. 13, p. 10370–10381, Jul 2022.

[17] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, “A survey on cloudlets, mobile edge, and fog computing,” in 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and ScalableCloud (EdgeCom). IEEE, Jun 2021, p. 139–142.

[18] J. Ying, J. Hsieh, D. Hou, J. Hou, T. Liu, X. Zhang, Y. Wang, and Y.-T. Pan, "Edge-enabled cloud computing management platform for smart manufacturing," in IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT). IEEE, Jun 2021, p. 682–686.

[19] D. Wu, X. Huang, X. Xie, X. Nie, L. Bao, and Z. Qin, "Ledge: Leveraging edge computing for resilient access management of mobile iot," IEEE Transactions on Mobile Computing, vol. 20, no. 3, p. 1110–1125, Mar 2021.

[20] W. Bao, C. Wu, S. Guleng, J. Zhang, K.-L. A. Yau, and Y. Ji, "Edge computing-based joint client selection and networking scheme for federated learning in vehicular iot," China Communications, vol. 18, no. 6, p. 39–52, Jun 2021.

[21] J. Singh, Y. Bello, A. R. Hussein, A. Erbad, and A. Mohamed, "Hierarchical security paradigm for IoT multiaccess edge computing," IEEE Internet of Things Journal, vol. 8, no. 7, p. 5794–5805, Apr 2021.

[22] C. Ding, A. Zhou, X. Ma, N. Zhang, C.-H. Hsu, and S. Wang, "Towards diversified iot services in mobile edge computing," IEEE Transactions on Cloud Computing, p. 1–1, 2021

[23] C. Guleria, K. Das, and A. Sahu, "A survey on mobile edge computing: Efficient energy management system," in 2021 Innovations in Energy Management and Renewable Resources (52042). IEEE, Feb 2021, p.1–4.

[24] "Home | tinyml foundation." [Online]. Available: https://www.tinyml.org/

[25] P. Warden and D. Situnayake, TinyML: machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers.O'Reilly, 2020. [Online]. Available: https://books.google.com/books/about/TinyML.html?id=sB3mxQEACAAJ

[26] N. N. Alajlan and D. M. Ibrahim, "TinyML: enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications," Micromachines, vol. 13, no. 6, p. 851, Jun 2022. [Online]. Available: https://www.mdpi.com/2072-666X/13/6/851

[27] D. L. Dutta and S. Bharali, "TinyML Meets IoT: a comprehensive survey," Internet of Things, vol. 16, p. 100461, Dec 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660521001025

[28] M. Shafique, T. Theocharides, V. J. Reddy, and B. Murmann, "TinyML: current progress, research challenges, and future roadmap," in 58th ACM/IEEE Design Automation Conference (DAC), Dec 2021, p. 1303–1306.

[29] TinyML platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

[30] Herramienta automatizada de aprendizaje automático (ML) para desarrolladores STM32. (s.f.). STM. https://www.st.com/en/development-tools/nanoedgeaistudio.html

[31] Immonen, R., & Hämäläinen, T. (2022). Tiny Machine Learning for Resource-Constrained Microcontrollers. Journal of Sensors, 2022, 1–11. https://doi.org/10.1155/2022/7437023

[32] Home. (s.f.). PyTorch. https://pytorch.org/mobile/home/

[33] GitHub - uTensor/uTensor: TinyML AI inference library. (s.f.). GitHub. https://github.com/uTensor/uTensor

[34] Development Kits | Efinix, Inc. (s.f.). Efinix, Inc. https://www.efinixinc.com/products-devkits.html

[35] TinyML Platform. (s.f.). Efinix, Inc. https://www.efinixinc.com/solutions-tinyml.html

[36] Efinix (2023) How EFINIX is conquering the hurdle of hardware acceleration for devices at the edge, EE Times. Available at: https://www.eetimes.com/how-efinix-is-conquering-the-hurdle-of-hardware-acceleration-for-devices-at-the-edge/

[37] Automated Machine Learning (ML) tool for STM32 developers. (s.f.). St. https://www.st.com/en/development-tools/nanoedgeaistudio.html

[38] Automated Machine Learning (ML) tool for STM32 developers. (s.f.-a). ST. https://www.st.com/en/development-tools/nanoedgeaistudio.html