UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE BIOMEDICINA

PROTOTIPO DE ASISTENCIA VISUAL PARA DETECTAR OBJETOS EN ESPACIOS DE APRENDIZAJE APLICANDO REDES NEURONALES CONVOLUCIONALES

Trabajo de titulación previo a la obtención del título de:

Ingeniero Biomédico

AUTOR: ESTEBAN ISAAC CISNEROS SIGCHA
TUTOR: MAYERLY ELVIRA SÁENZ CASALLAS

Quito-Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORIA DEL TRABAJO DE TITULACIÓN

Yo, Esteban Isaac Cisneros Sigcha con documento de identificación N° 1722318720 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 31 de julio del 2024

Atentamente,

Isaac Cisneros

Esteban Isaac Cisneros Sigcha
1722318720

CERTTFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITECNICA SALESIANA

Yo, Esteban Isaac Cisneros Sigcha con documento de identificación N◦ 1722318720, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: "Prototipo de asistencia visual para detectar objetos en espacios de aprendizaje aplicando redes neuronales convolucionales", el cual ha sido desarrollado para optar por el título de: Ingeniero Biomédico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 31 de julio del 2024

Atentamente,

Isaac Cisneros

_____

Esteban Isaac Cisneros Sigcha

1722318720

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Mayerly Elvira Sáenz Casallas con documento de identificación N°1758414393, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “Prototipo de asistencia visual para detectar objetos en espacios de aprendizaje aplicando redes neuronales convolucionales”, realizado por Esteban Isaac Cisneros Sigcha con documento de identificación N° 1722318720, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 31 de julio del 2024

Atentamente,

_____

Mayerly Elvira Sáenz Casallas

1758414393

Dedicatoria

A dios por siempre ser ese pilar espiritual para guiarme en cada etapa de mi vida al demostrarme que nunca me tengo que dar por vencido ya que las metas si se logran con esfuerzo y sacrificio.

A mi madre por estar presente en todas las circunstancias y demostrarme su apoyo emocional, amor y respeto en cada reto que se ha presentado dándome consejos para seguir adelante, es la persona más importante en mi vida.

A mis hermanos por todo el apoyo que me han brindado a lo largo de mi vida estudiantil quienes con su ejemplo y fortaleza han logrado que se haga realidad lo que algún día me propuse y logre cumplir con mucha dedicación.

Esta tesis está dedicada a todos ustedes que fueron parte esencial y fundamental en todo este proceso de éxito, gracias a ustedes por ser mi inspiración y proyectarme por el buen camino para ser un buen profesional y ser humano, este logro es de todos muchas gracias por todo su apoyo.

Con amor

Isaac Cisneros

Resumen

A lo largo de este proyecto técnico, se ha desarrollado un prototipo de asistencia visual que contiene módulos sensores para distinguir distancias de obstáculos alrededor de un entorno educativo, el propósito principal se ha centrado en aplicar una red neuronal convolucional.

Este prototipo inicial posibilita el reconocimiento de objetos y facilita el uso de una interfaz de usuario auditiva que permite escuchar mediante auriculares los objetos que se detectan.

Para la creación de la red neuronal se utilizó la plataforma Edge impulse que a partir de la creación de un dataset con objetos referentes al aula educativa permitió configurar la imagen, parámetros, generar características, ajustar la red neuronal convolucional, entrenar el modelo, visualizar las métricas y por último la red que se construyó facilito trabajar con Arduino.

Los resultados del desarrollo de este prototipo aplicando pruebas de funcionamiento se realizó con varios estudiantes y fueron efectivas demostrando en cada uso que tiene una diferente fluidez con distintos objetos mientras que la precisión depende de algunos factores como el fondo y la luz sin embargo quedo demostrado que cumple correctamente la función y los objetivos planteados del proyecto.

Palabras Clave: IA- Red Neuronal- Reconocimiento- Prototipo

Abstract

Throughout this technical project, a visual assistance prototype containing sensor modules to distinguish distances from obstacles around an educational environment has been developed, the main purpose of which has focused on applying a convolutional neural network.

This initial prototype enables object recognition and facilitates the use of an auditory user interface that allows listening through headphones to the detected objects.

For the creation of the neural network, the Edge impulse platform was used which, based on the creation of a dataset with objects referring to the educational classroom, made it possible to configure the image, parameters, generate characteristics, adjust the convolutional neural network, train the model, visualize the metrics and, finally, the network that was built made it easier to work with Arduino.

The results of the development of this prototype by applying functional tests were carried out with several students and were effective demonstrating in each use that it has a different fluency with different objects while the accuracy depends on some factors such as the background and the light, however it was demonstrated that it correctly fulfils the function and the objectives of the project.

Keywords: AI - Neural Network - Recognition - Prototyping

Contenido

Dedicatoria ..... **V**
Resumen ..... **VI**
Abstract ..... **VII**
1. Introducción **1**
2. Capítulo 1: Base conceptual **4**
2.1 Definición de Discapacidad visual 4
2.1.1 Tipos de discapacidad visual 4
2.2 Tecnología Sonora para personas con Discapacidad visual 6
2.2.1 Interfaz de usuario auditiva 6
2.3 Red Neuronal 8
2.3.1 Capas de una red neuronal 8
2.4 Red Neuronal Convolucional 9
2.4.1 Estructura fundamental 10
2.5 Definición de Edge Impulse 11
2.5.1 Beneficios del Edge Impulse 11
3. Capítulo 2: Diseño de hardware del prototipo de asistencia visual **12**
3.1 Aspectos técnicos de los componentes electrónicos 12
3.1.1 Microcontrolador ESP32 12
3.1.2 ESP32-CAM 13
3.1.3 Batería de Litio 15
3.1.4 Sensor Ultrasónico HC-SR04 16
3.1.5 Módulo de carga micro USB 17
3.1.6 Módulo DFPlayer Mini 18
3.1.7 Conector Jack 3.5 mm 19
3.1.8 Tarjeta MicroSD 20
3.1.9 Pulsador 21
3.1.10 Interruptor rocket 22
3.1.11 Buzzer 23
3.2 Interpretación del diseño 24

3.3 Esquema eléctrico de la placa en Proteus 25
3.4 Elaboración del diseño PCB 26
3.5 Diseño realizado del prototipo en SolidWorks 28
3.5.1 Aspecto de las tapas 28
3.5.2 Aspectos de la base 31
3.5.3 Ensamblaje 32
3.5.4 Estructura y diseño final del prototipo 33
4. Capítulo 3: Desarrollo de software del prototipo de Asistencia visual 35
4.1 Edge Impulse 35
4.1.1 Dashboard y adquisición de la base de datos 36
4.1.2 Diseño del impulso 40
4.1.3 Parámetros de la imagen 44
4.1.4 Configuración para la detección de objetos 47
4.1.5 Arquitectura de redes neuronales 48
4.2 FOMO (Objetos más rápidos, más objetos) 48
4.3 MobileNet 49
4.3.1 MobileNet V2 49
4.3.2 Diagrama de bloques MobilNet V2 50
4.3.3 Módulo Bottleneck 52
4.3.4 Convolución de expansión y proyección 54
4.3.5 Softmax 54
4.4 Métricas y matriz de confusión del modelo 55
4.5 EON Tuner 56
4.6 Configuración de despliegue 57
4.7 Interfaz de usuario auditiva 58
4.7.1 Diagrama de bloques de la interfaz de usuario auditiva 59
5. Capítulo 4: Resultados 60
5.1 Resultados de proyectos creados 60
5.2 Resultados de los datasets 62
5.2.1 Dataset de prueba 62
5.2.2 Dataset Proyecto Final 62
5.2.3 Comparación de labels 64
5.2.4 Resultados de crear el impulso 65

5.3 Resultados de parámetros y generación de características 66
5.3.1 Resultados de parámetros 67
5.3.2 Resultados del Procesamiento digital de señales y características procesadas 68
5.4 Resultados de generar características 69
5.4.1 Comparación del conjunto de entrenamiento 69
5.5 Resultados del explorador de características 70
5.6 Resultado de la detección de objetos 71
5.6.1 Resultados del ajuste de la red neuronal 71
5.6.2 Resultados de la arquitectura de las redes neuronales 72
5.6.3 Últimos resultados del entrenamiento (conjunto de validación) 73
5.6.4 Resultados de métricas (conjunto de validación) 74
5.7 Resultado final 75
5.7.1 Configuración de despliegue 75
5.7.2 Optimización del modelo 76
5.7.3 Construcción y generación del modelo final 77
5.8 Pruebas de predicción con diferentes modelos (EON Tuner) 77
5.8.1 Predicción N° 1 78
5.8.2 Predicción N° 2 79
5.8.3 Predicción N° 3 80
5.8.4 Predicción N° 4 80
5.8.5 Predicción N° 5 81
5.8.6 Predicción N° 6 82
5.8.7 Predicción N° 7 83
5.8.8 Predicción N° 8 84
5.8.9 Predicción N° 9 85
5.8.10 Predicción N° 10 86
5.9 Resultados del prototipo en entornos educativos 87
6. Conclusiones y Recomendaciones 89
6.1 Conclusiones 89
6.2 Recomendaciones 89
7. Referencias Bibliográficas 91
Referencias 91
8. Anexos 93

ÍNDICE DE FIGURAS

Figura 1. Interfaz de usuario auditiva 8
Figura 2. Arquitectura de una red neuronal 9
Figura 3. Arquitectura de una red neuronal convolucional 10
Figura 4. Esquema eléctrico en Proteus 25
Figura 5. Diseño PCB 26
Figura 6. Vista frontal 27
Figura 7. Vista trasera 27
Figura 8. Tapa frontal 28
Figura 9. Tapa lateral 29
Figura 10. Tapa frontal 2 29
Figura 11. Tapa lateral 2 30
Figura 12. Tapa trasera 30
Figura 13. Base del prototipo 31
Figura 14. Ensamblaje 32
Figura 15. Diseño final 34
Figura 16. Diseño final con las correas Nailon y velcro 34
Figura 17. Login Edge Impulse 35
Figura 18. Selección de la placa 36
Figura 19. Imagen detallada de un lápiz y tijera con el label 1 y 2 37
Figura 20. Borrador label 3 y compas label 4 37
Figura 21. Moneda label 6 y goma label 7 38
Figura 22. Regla label 8 y grapadora label 9 38
Figura 23. Objeto tomado en diferentes ángulos 39
Figura 24. Diseño del impulso 40
Figura 25. Datos de imagen 41
Figura 26. Bloque de procesamiento 42
Figura 27. Bloques de aprendizaje 43
Figura 28. Características de entrada y salida 44
Figura 29. Datos brutos 44
Figura 30. Parámetros de la imagen 45
Figura 31. Características procesadas de la imagen 46
Figura 32. Generación de características 46
Figura 33. Configuración para la detección de objetos 47
Figura 34. Arquitectura de redes neuronales 48
Figura 35. Diagrama de bloques MobileNet V2 50
Figura 36. Diagrama de bloques de la composición de los módulos 52
Figura 37. Relu6 54
Figura 38. Interpretación Softmax 55
Figura 39. Métricas del modelo 56
Figura 40. EON Tuner 57

Figura 41. Configuración de despliegue 58
Figura 42. Diagrama de bloques de la interfaz de usuario auditiva 59
Figura 43. Primer proyecto creado 60
Figura 44. Proyecto final creado 61
Figura 45. Primer Dataset y Dataset final 64
Figura 46. Comparación de labels 64
Figura 47. Imágenes en diferentes ángulos 65
Figura 48. Resultados de los datos de imagen 66
Figura 49. Datos y características brutas 66
Figura 50. Datos y características brutas del dataset final 67
Figura 51. Parámetros dataset inicial y final 68
Figura 52. Resultados DSP 69
Figura 53. Comparación del conjunto de entrenamiento 70
Figura 54. Resultados del explorador de características 71
Figura 55. Resultados del ajuste de la red neuronal 72
Figura 56. Resultados de arquitecturas de las redes neuronales 73
Figura 57. Resultados de entrenamiento 74
Figura 58. Resultados de métricas 74
Figura 59. Configuración de despliegue 76
Figura 60. Optimización del modelo 76
Figura 61. Construcción y generación del modelo final 77
Figura 62. Predicción 1 78
Figura 63. Predicción 2 79
Figura 64. Predicción 3 80
Figura 65. Predicción 4 81
Figura 66. Predicción 5 82
Figura 67. Predicción 6 83
Figura 68. Predicción 7 84
Figura 69. Predicción 8 85
Figura 70. Predicción 9 86
Figura 71. Predicción 10 87
Figura 72. Prueba N°1 de funcionamiento en entornos educativos 88
Figura 73. Prueba N°2 de funcionamiento en entornos educativos 88

ÍNDICE DE TABLAS

Tabla 1. Clasificación según la OMS 5
Tabla 2. Especificaciones técnicas del Esp32 12
Tabla 3. Especificaciones técnicas del ESP32-CAM 14
Tabla 4. Especificaciones técnicas de la batería de Litio 15
Tabla 5. Especificaciones Técnicas del HC-SR04 16
Tabla 6. Especificaciones Técnicas del módulo de carga 17
Tabla 7. Especificaciones Técnicas del DFPlayer Mini 18
Tabla 8. Especificaciones Técnicas del Jack 3.5mm 19
Tabla 9. Especificaciones Técnicas de la MicroSD 20
Tabla 10. Especificaciones Técnicas del pulsador 21
Tabla 11. Especificaciones Técnicas del interruptor 22
Tabla 12. Especiaciones Técnicas del buzzer 23
Tabla 13. Pieza y material 33
Tabla 14. Organización del MobileNet V2 51
Tabla 15. Datos de recolección de dataset prueba 62
Tabla 16. Datos de recolección 62
Tabla 17. Tabla comparativa 75

ÍNDICE DE ANEXOS

Anexo 1. Tapa 1 del prototipo 93
Anexo 2. Tapa 2 del prototipo 94
Anexo 3. Tapa 3 del prototipo 95
Anexo 4. Tapa 4 del prototipo 96
Anexo 5. Tapa 5 del prototipo 97
Anexo 6. Base 98
Anexo 7. Ensamblaje 99
Anexo 8. Prueba de uso 1 100
Anexo 9. Prueba de uso 2 101
Anexo 10. Prueba de uso 3 102
Anexo 11. Prueba de uso 4 103
Anexo 12. Prueba de uso 5 104
Anexo 13. Prueba de uso 6 105
Anexo 14. Prueba de uso 7 106
Anexo 15. Código Final 107

1. Introducción

En la actualidad se encuentran alrededor de 481.392 personas que tienen ciertos tipos de discapacidad en el Ecuador y de las cuales el 12% de estas personas tiene discapacidad visual en diferentes grados, según las estadísticas archivadas y registradas en la CONADIS. (Pérez, 2024)

Las dificultades visuales impactan diversos aspectos de la vida de las personas, incluyendo sus habilidades profesionales, académicas, comunitarias, afectivas, cognitivas y físicas. Estas dificultades pueden obstaculizar la adquisición de conocimientos, el desempeño de tareas cotidianas, el desplazamiento, la orientación y la interacción social. Para compensar estas dificultades, las personas con discapacidades visuales utilizan la información que obtienen de sus otros sentidos, como el tacto para reconocer objetos, la audición para orientarse y el olfato, entre otros (OMS, 2023).

A través de los años se ha presentado varios proyectos para brindar ayuda a personas que tienen discapacidad visual y en el cual se ha logrado conceder independencia en su vida cotidiana, dentro de estos proyectos se realizaron prototipos y dispositivos que sirven a las personas con discapacidad visual a localizar objetos de manera auditiva y haptica que le faciliten desempañarse en los lugares que lo rodean. Entre los dispositivos actualmente disponibles en Ecuador para apoyar a las personas con discapacidad visual se encuentran básculas sonoras, bastones convencionales, materiales escritos en braille, calculadoras inteligentes y ordenadores con sistemas como Jaws o el dispositivo Hand Eyes (Reinoso, 2019).

Debido a esto el prototipo inicial que se pretende realizar es innovador y le va permitir a las personas utilizarlo en ambientes educativos, no solo se busca el reconocer los objetos sino poner en practica la red neuronal convolucional entrenada mediante una plataforma en línea creada para simplificar el progreso de herramientas de inteligencia artificial. El enfoque de este proyecto

incluye el diseño, la implementación y la evaluación para asegurar la facilidad de uso y adaptación del prototipo en los entornos educativos. El presente proyecto ayudara en futuras investigaciones para determinar nuevas mejoras y limitaciones exhibiendo la capacidad de la inteligencia artificial en discapacidades.

El esquema del trabajo se distribuye en cinco capítulos desarrollados para tratar facetas primordiales del proyecto técnico. En el primer capítulo se explica una breve introducción de los fundamentos teóricos que permite interpretar la ejecución del prototipo, explorando la relevancia que tiene la inteligencia artificial usando herramientas innovadoras. El segundo capítulo describe el hardware del prototipo brindando el diseño, los planos y el montaje de los elementos. El tercer capítulo se centra en el desarrollo del software este punto es importante ya que se utiliza una plataforma para trabajar con la red neuronal convolucional. El cuarto capítulo se realizan las pruebas de funcionalidad, evaluando la facilidad de uso y la adaptación del prototipo en entornos educativos. Y por último el quinto capítulo presenta las conclusiones y las recomendaciones que se sugiere para algunas optimizaciones en el futuro para este proyecto.

Objetivos

Objetivo General

Desarrollar un prototipo de asistencia visual usando redes neuronales convolucionales para detectar objetos en entornos educativos.

Objetivos Específicos

- Diseñar una interfaz de usuario para el prototipo de asistencia visual, que incorpore el reconocimiento en tiempo real de objetos mediante redes neuronales convolucionales.
- Implementar un algoritmo de redes neuronales convolucionales para detectar objetos en tiempo real en entornos de aprendizaje.
- Evaluar la facilidad de uso y adaptación del prototipo en entornos educativos.

2. Capítulo 1: Base conceptual

En este capítulo se describe los diferentes conceptos de la discapacidad visual, de la red neuronal y la plataforma utilizada para detectar los objetos dentro del entorno educativo.

2.1 Definición de Discapacidad visual

Es un estado físico que causa la reducción de la facultad de visualizar ya sea total o parcial que complica las actividades del sistema óptico. Se conoce como discapacidad visual porque no puede ser rectificable por los lentes de contacto, los cristales, la cirugía, la medicación y terapia. (Calvo, 2023)

2.1.1 Tipos de discapacidad visual

- **Deficiencia visual leve:** Las personas con una disminución leve tienen la capacidad de captar los objetos diminutos, además con sus particularidades y color adecuado, tienen la capacidad de asimilar al momento de escribir y leer siempre que se use asistencias ópticas apropiadas. Las personas son capaces de ejecutar tareas diarias sin impedimentos. (Roura, 2010)
- **Deficiencia visual moderada:** Dispone de un amplio nivel de observación, a pesar de que continúa permaneciendo restringida para la escritura y lectura, estas personas con este tipo de discapacidad visual son capaces de llevar a cabo tareas sin la exigencia de adecuaciones especiales. (Universidad de Granada, 2023)
- **Deficiencia visual severa:** Las personas con una disminución severa de visión tienen la capacidad de detectar las sombras y luz, pero requieren del braille para escritura y lectura, además de varias asistencias ópticas como los telescopios distintivos con el fin de ser capaz de actuar sin ningún inconveniente. Sin embargo, estas personas requieren para desplazarse una adecuada preparación con el apoyo de dispositivos de asistencia. (Roura, 2010)

- **Ceguera Total:** Las personas que tienen esta condición, no tienen nada de visión, esto significa que no observan ningún objeto u cosa, ni la iluminación, de la misma manera se le impide ejecutar actividades visuales como leer y escribir. Usan para la escritura y la lectura el sistema braille y para desplazarse se necesita de dispositivos de asistencia como bastones eléctricos, gafas inteligentes o de la compañía de una persona en la situación de no tener una preparación apropiada en asistencia y movilización, estas personas requieren de los sentidos sobrantes para poder sobrellevarse. (Roura, 2010)

En la tabla 1 se especifica la clasificación de la discapacidad visual y la medida de agudeza visual el primer número expresa la separación desde la que se ejecuta el análisis de agudeza visual esta medida es en metros, mientras que el segundo número señala la magnitud más diminutiva en el cual una persona con una observación normal es capaz de visualizar a esa distancia. (Arranz, 2023)

Tabla 1. *Clasificación según la OMS*

Elaborado por: Autor, 2024

Clasificación	Medida de agudeza o lucidez visual
Leve	Lucidez visual menor a 6/12 o igual o mayor 6/18
Moderada	Lucidez visual menor a 6/18 o igual o mayor 6/60
Grave	Lucidez visual menor a 6/60 o igual o mayor 3/60
Ceguera	Lucidez visual menor a 3/60

Nota: Esta tabla especifica la clasificación de las discapacidades visuales según la OMS. (Arranz, 2023)

2.2 Tecnología Sonora para personas con Discapacidad visual

Se centra en la generación, emisión, admisión y control del sonido. Abarca el análisis de sucesos asociados con la formación y difusión del sonido, al igual que su conexión con los medios físicos y su interpretación desde el punto de vista del ser humano. (Comunicación, 2023)

En esta tecnología se usa un mapa sonoro del entorno que posibilita a las personas con discapacidad visual localizar artefactos u objetos que normalmente no generan un sonido. (Ferri,2017)

2.2.1 Interfaz de usuario auditiva

Son una clase de estructura de comunicación que se fundamenta en la voz y audio como método primordial de interacción entre maquinas e individuos. Tienen la capacidad de alternar a partir de sencillos comandos de timbre o voz incluso en emisarios comunicativos complicados desde sonoridades naturales inclusive en ambientes de tonos envolventes a partir de audífonos hasta bocinas inteligentes. No son únicamente una manera de optimizar el conocimiento del usuario, sin embargo, son un modo de generar recientes iniciativas de importancia y posibilidades mercantiles para múltiples sectores y campos. (García, 2022)

Beneficios y desafíos:

- Disponibilidad e integración: Tienen la capacidad de brindar una manera orgánica e inmediata de comunicar con la tecnología específicamente para personas que tienen discapacidades visuales, motrices, intelectuales o que poseen algún tipo de analfabetismo o multilingües. Además, son capaces de autorizar nuevos modos de ingenio y expresividad como la narración y el sonido. No obstante, de la misma forma proponen varios asuntos colectivos y morales tal como confidencialidad, la protección, los estereotipos y la seguridad. (Rodríguez, 2023)

- Rendimiento y desempeño: Son capaces de proporcionar una manera más veloz y confortable de llevar a cabo actividades, sobre todo en el instante que los ojos o las manos de la persona están no accesibles, puede ser incluso cuando maneja, cuando prepara los alimentos o cuando realiza actividad física. De igual manera están en condiciones de incrementar las habilidades intelectuales de la persona o usuario tales como la rememoración, la concentración y la educación, al brindar datos, feedback, dirección significativas y adecuadas. A pesar de eso igualmente necesitan un elevado grado de exactitud, fiabilidad y estabilidad, al igual que un entendimiento evidente del entorno, el propósito y las selecciones del usuario. (Rodríguez, 2023)
- Emoción y responsabilidad: Tienen la capacidad de generar un vínculo envolvente y afectivo en medio de la persona y la tecnología, sobre todo en el momento que el ruido es técnico, practico o comunicativo. Igualmente son capaces de beneficiarse de la capacidad de la voz, el cual es uno de las facetas más singulares y relevantes de la interacción humana para comunicar identidad, afectos e indicios relacionales. (Rodríguez, 2023)

A continuación, se presenta un ejemplo del proceso de utilización de una interfaz de voz de usuario. Se tiene primero la identificación de instrucciones de voz o términos para un léxico determinado. En segunda instancia el verde se presentan los viables resultados de verificación o análisis del orden y por último el rojo detalla si la estructura no admite algún comando del vocabulario accesible, como se representa en la figura 1.

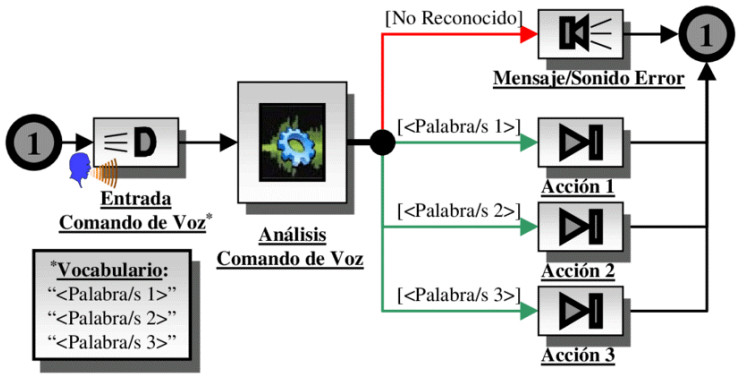


Figura 1. *Interfaz de usuario auditiva*

2.3 Red Neuronal

Es una técnica de Inteligencia Artificial que instruye a los ordenadores a computar información de una forma el cual esté relacionada en la manera en que lo realiza el cerebro humano. Consiste en una clase de método de procedimiento de aprendizaje automático que usa las neuronas enlazadas en una composición de niveles o capas el cual tiene un parecido al funcionamiento del cerebro humano. Genera una estructura ajustable que los ordenadores usan para adquirir conocimientos de los desaciertos y optimizar constantemente. Así las redes neuronales simuladas pretenden de solucionar dificultades complejas como la ejecución de síntesis de escritos o la identificación de objetos y rostros con mayor exactitud. (Pérez, 2024)

2.3.1 Capas de una red neuronal

Una red neuronal posee diferentes maneras de repartir, enlazar y relacionar sus capas. Hay tres clases de capas y cada una de estas capas contiene sus correspondientes neuronas de forma global se consigue comprender su estructura o arquitectura en la figura 2 en el cual se tiene la capacidad de considerar de forma general la asignación de las capas y sus enlaces. (Ramírez, 2023)

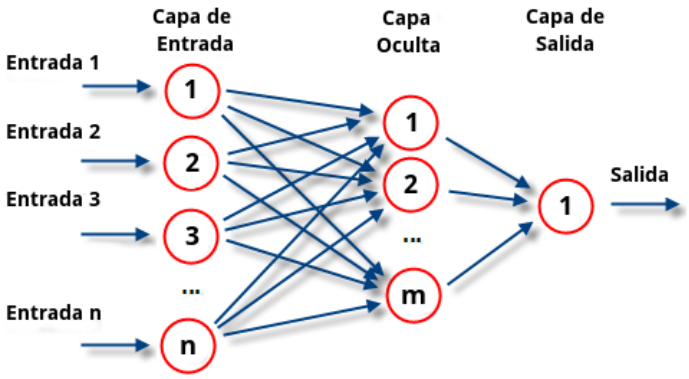


Figura 2. *Arquitectura de una red neuronal*

- **Capa de entrada:** Necesita de variables o información que produzca la dificultad a solucionar debido a que de acuerdo a estos es capaz de especificar la cantidad y categorías de neuronas que la componen. (Morales, 2024)
- **Capas intermedias:** Esta capa depende de la complicación del problema o dificultad que la red necesita solucionar debido a esto es capaz de requerir más de una sola capa. En una investigación y manejo de estas capas son capaces de adquirir características intrigantes para la red neuronal tales como la propagación, obtención de propiedades y versatilidad. (Morales, 2024)
- **Capa de salida:** Este depende de la disposición de la estructura y la implementación que dispondrá la red neuronal donde se va ir ubicando el número de las neuronas de salida. (Morales, 2024)

2.4 Red Neuronal Convolucional

Son una alternativa de las redes neuronales elaboradas concretamente para manejar información con organización de malla, como señales acústicos o imágenes. Están impulsadas en el aspecto en que el cerebro humano ejecuta datos visuales, las redes neuronales convolucionales han

evidenciado ser sumamente eficientes en la misión de identificación de modelos. (KeepCoding, 2024)

2.4.1 Estructura fundamental

La organización primordial de una red neuronal convolucional está conformado de tres elementos fundamentales: (KeepCoding, 2024)

- **Capas convolucionales:** Estas son las capas principales de la red neuronal convolucional, usan filtros para digitalizar el ingreso y asimilar interpretaciones locales. Cada filtro identifica esquemas como perímetros, matices o superficies. (KeepCoding, 2024)
- **Capas de asociación:** Reconocidas como capas de pooling, estas limitan la magnitud de la demostración y reducen el volumen de variables en la red. Frecuentemente se usan procedimientos de máximo o valor medio para mantener los tributos más significativos. (KeepCoding, 2024)
- **Capas completamente enlazadas:** Finalmente se hallan capas que vinculan globalmente las neuronas de la capa previa, posibilitando formando conexiones más complejas y efectuando distribuciones finales. (KeepCoding, 2024)

La arquitectura de una red neuronal convolucional se reconoce por su esquema adecuadamente organizado el cual se parece a la ejecución de la estructura visual humana. En la figura 3 se observa la arquitectura o estructura en detalle.

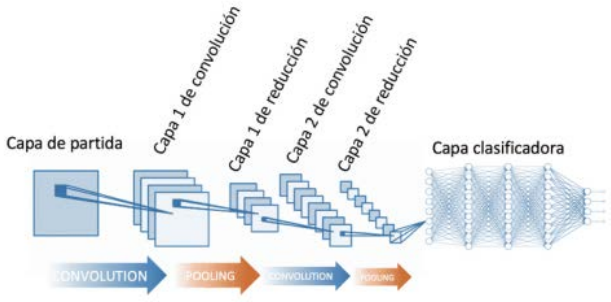


Figura 3. *Arquitectura de una red neuronal convolucional*

2.5 Definición de Edge Impulse

Es una plataforma para elaborar algoritmos de Machine Learning para ejecutarse en sistemas integrados como los microcontroladores o dispositivos con recursos limitados. Esta plataforma permite el código no entre de manera tan exhaustiva debido a que es factible ejecutar el algoritmo introduciendo una base de datos, adaptando todos los parámetros del sistema y entrenando el software con distintas situaciones de ejecución aleatoria. (Torres, 2022)

2.5.1 Beneficios del Edge Impulse

- Facilita el avance veloz de modelos de Machine Learning para instrumentos Edge.
- Favorece la conjunción con una extensa variedad de partes físicas IoT e instrumentos incluidos (Torres, 2022)
- Posibilita la adaptación de modelos de inteligencia artificial de acuerdo a los requerimientos del diseño. (Ruiz, 2024)
- Mejora los modelos para implementarse de forma eficaz en equipos con capacidades restringidas. (Ruiz, 2024)
- Apoya para conservar los datos cruciales a nivel local para la optimización de la protección y la confidencialidad. (Torres, 2022)
- Autoriza programas de sistemas inteligentes que necesitan una espera baja y ejecución sin conexión.

3. Capítulo 2: Diseño de hardware del prototipo de asistencia visual

En este capítulo se profundiza en el esquema del hardware físico del prototipo de asistencia visual considerando puntos clave como el diseño del prototipo, la distribución de los diferentes componentes electrónicos utilizados en posiciones claves y de la localización exacta de los conectores.

3.1 Aspectos técnicos de los componentes electrónicos

3.1.1 Microcontrolador ESP32

Es un microcontrolador de limitado costo e inferior uso de energía el cual está elaborado para aplicaciones de IoT Internet de las cosas, fue elaborado por Espressif Systems y es correspondiente con el reglamento WIFI y Bluetooth. Este microcontrolador usa un CPU de hasta doble núcleo, que opera a una rapidez de máximo 16 MB y cuenta con una memoria SRAM de máximo 520 KB. (Polaridad, 2024) El principio benéfico de esta placa de avance es la diversidad de pines de distintas clases, por ejemplo, la magnitud de ADC que se intensificaron en cuanto a alternativas antiguas y con el microconntrolador de doble núcleo solamente uno se utiliza en la parte de control inalámbrico. (Rosario, 2018)

Tabla 2. *Especificaciones técnicas del Esp32*

Elaborado por: Autor, 2024

Característica	Detalles
ESP32	
Procesador Tensilica Xtensa	Xtensa 32bits LX6 hasta 240MHz
Wifi	Soporte Integrado 802.11 b/g/n
Bluetooth	V4.2 BR/EDR y BLE
Memoria Flash	4 MB
Memoria ROOM	448 KB
Memoria SRAM	520 KB
Memoria Flash	16 MB
Periféricos	Controladores SD, Ethernet, 12S, DAC, CAN
Interfaces	GPIO, 12C, SPI, UART, PWM, ADC
Consumo	Consumo de energía bajo

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.2 ESP32-CAM

Es un instrumento que lo conocen por ser completo, ya que además de tener interconexión WIFI y Bluetooth que llega de fábrica, pines GPIO, se le agregado dos alternativas nuevas. Tiene incorporado una diminuta cámara de video y una conexión para una tarjeta MicroSD en el lugar en

el que será capaz de conservar fotos y videos. Este dispositivo es de costo bajo y es muy usado en IoT (Internet de las cosas), por otra parte, hay como enlazarlo a Home assistant y que sea un elemento importante de un sistema inteligente o automatizado de manera sencilla. (Pascual, 2022)

Tabla 3. *Especificaciones técnicas del ESP32-CAM*

Elaborado por: Autor, 2024

Característica	**Detalles**
ESP32-CAM	
Procesador	Dual-Core Xtensa LX6 a 240 MHz
Wi-Fi:	802.11b/g/n/e/i
Bluetooth:	Bluetooth 4.2 con BLE
Periféricos	Controladores SD, Ethernet, 12S, DAC, CAN
Camara	Sensor OV2640 2 MP
Interfaces:	UART, SPI, I2C, PWM, GPIO
Memoria RAM:	520 KB SRAM, externa 4 MB PSRAM
Memoria Flash	4 MB
Soporte microSD:	hasta 4GB
Consumo	Consumo de energía bajo

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.3 Batería de Litio

Es un instrumento apto para conservar y desprender energía eléctrica por medio de una corriente de electrones en medio de dos terminales, esta corriente de electrones va desde el ánodo al cátodo y se reinstaura en el momento en el que las pilas se restauran posibilitando que la secuencia empiece de nuevo. (Ecoinventos, 2022)

Tabla 4. *Especificaciones técnicas de la batería de Litio*

Elaborado por: Autor, 2024

Característica	**Detalles**
Batería de Litio	
Energía detallada	100–265 W·h/kg (0.36–0.95 MJ/kg)
Compactación energética	250–730 W·h/L (0.90–2.23 MJ/L)
Intensidad detallada	~250-~340 W/kg
Capacidad carga/descarga	80–90%

Nota: Todas estas especificaciones son muy importantes para ver la vida útil de la batería de litio que puede tener durante un uso muy extenso en diferentes situaciones, su gran ventaja es que son pequeñas y no pesan demasiado por lo cual se puede utilizar sin ningún inconveniente. (Panasonic, 2010)

3.1.4 Sensor Ultrasónico HC-SR04

Este es un sensor de intervalos o distancias por medio de ultrasonidos es idóneo para identificar objetos y evaluar la distancia donde se localiza en un rango estimado de 2cm a 450cm. El sensor trabaja a través de ultrasonidos y abarca completamente la electrónica responsable de realizar el cálculo. Su utilización es fácil al igual que remitir el pulso de inicio y calibrar la amplitud del pulso de vuelta. (Flores, 2023)

Tabla 5. *Especificaciones Técnicas del HC-SR04*

Elaborado por: Autor, 2024

Característica	**Detalles**
HC-SR04	
Alimentación	5V DC
Rango de medición	De 2cm a 400cm
Resolución	0.3cm
Frecuencia de operación	40kHz
Corriente de trabajo	Menos de 15 mA
Angulo de medición	Menos de 15 grados
Precisión	± 0.3cm

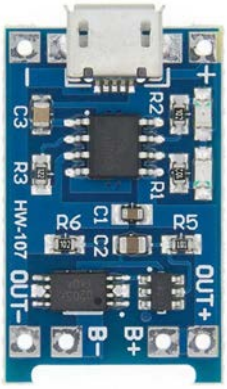
Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.5 Módulo de carga micro USB

Posibilita cargar de manera simple y protegida las baterías de LiPo y LiOn de una sola celda. Opera con celdas o baterías que contengan voltajes de carga estándar de 3.7V y de carga total que contiene 4.2V. Es óptimo para cargas o baterías LiPo 18560 o con baterías LiOn que tienen los teléfonos móviles inteligentes. (López, 2022)

Tabla 6. *Especificaciones Técnicas del módulo de carga*

Elaborado por: Autor, 2024

Característica	**Detalle**
HW-107	
Voltaje de entrada	5V DC
Tamaño	Aproximadamente 26mm x 17mm
Método de carga	Lineal
Corriente de carga máxima	1A (1000 mA)
Protección	Contra la sobrecarga, cortocircuito, sobre corriente y sobre-descarga
Indicadores ED	Indicador de carga (rojo: cargando, verde: carga completa)

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.6 Módulo DFPlayer Mini

Esta diminuta placa es un minúsculo reproductor de audio MP3 que tiene incorporado un amplificador y es capaz de operar independientemente tan solo enlazando unos botones o pulsadores que facilitan la reproducción de archivos o carpetas inmediatamente cargados en una tarjeta microSD. (Sánchez, 2023)

Tabla 7. *Especificaciones Técnicas del DFPlayer Mini*

Elaborado por: Autor, 2024

Característica	**Detalles**
DFPlayer Mini	
Voltaje de alimentación	3.3V – 5V DC
Interfaz de control	UART (TTL serial)
Capacidad de almacenamiento	Hasta 32 GB (microSD)
Formato de archivos de audio	MP3, WAV
Salida de audio	3.5mm estéreo o conexión directa de altavoces
Consumo de energía	Gasto mínimo en el transcurso de la reproducción de archivos de audio
Funciones adicionales	▪ Soporte para registros de reproducción ▪ Regulación del volumen

	▪ Regulación de la reproducción como: siguiente, reproducir, anterior y pausa.

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.7 Conector Jack 3.5 mm

Son comúnmente adaptadores o conectores de audio que conducen una señal o información, utilizados esencialmente para conducir señal de audio del reproductor a unos audífonos o un sistema de sonido. (Jiménez, 2023)

Tabla 8. *Especificaciones Técnicas del Jack 3.5mm*

Elaborado por: Autor, 2024

Características	**Detalles**
Jack 3.5mm	
Tipo de conector	Conector TRS
Tipo de montaje	Vertical u horizontal
Montaje	Superficial (SMD) o a través de agujeros (THD)
Numero de contactos	3 contactos: para señales de audio derecha, izquierda y tierra

Tensión nominal	5V
Aplicaciones	Para conectar auriculares o altavoces
Material del conector	Aleación de cobre

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.8 Tarjeta MicroSD

Son adaptaciones de las tarjetas SD de pequeña magnitud y la más grande diferencia en medio de las dos es la variable de estructura. También son más adaptables porque generalmente se distribuyen con un conector SD que posibilita utilizar las tarjetas microSD en dispositivos o instrumentos de hardware que únicamente permiten tarjetas SD. Las microSD normalmente se usan para aumentar la capacidad de almacenaje de móviles y cámaras. (Soto, 2022)

Tabla 9. *Especificaciones Técnicas de la MicroSD*

Elaborado por: Autor, 2024

Característica	**Detalles**
MicroSD	
Capacidad	2GB hasta 128TB
Temperatura de operación	-25°C a 85°C
Voltaje de operación	2.7V a 3.6V

Interfaz de comunicación	SPI o SDIO
Velocidad de transferencia	2, 4, 6, 10 MB/s
Dimensiones	15mm x 11mm x 1mm
Compatibilidad	Con Tarjetas microSDHC y microSDXC

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.9 Pulsador

Es un elemento eléctrico que autoriza o detiene el trayecto del flujo eléctrico mientras se presiona u oprime. El pulsador únicamente se inicia o concluye en el momento que el usuario lo oprime y lo conserva apretado, al liberarlo regresa a su postura preliminar. Para que opere correctamente el pulsador necesita poseer un resorte el cual hace que retorne al estado previo luego de oprimirlo. (Navarro, 2022)

Tabla 10. *Especificaciones Técnicas del pulsador*

Elaborado por: Autor, 2024

Característica	**Detalles**
Switch pulsador	
Tipo de interruptor	Pulsador táctil
Posiciones	2 posiciones abierto y cerrado
Tensión nominal	12V DC
Numero de pines	6 pines

Tipo de montaje	SMD y THD
Corriente nominal	1A
Dimensiones	8mm x 8mm
Durabilidad	10,000 o 100,000 ciclos de activación

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.10 Interruptor rocket

Se utiliza para controlar el activado y desactivado de instrumentos eléctricos básicamente cuenta con dos posiciones ON/OFF. (Vargas, 2023)

Tabla 11. *Especificaciones Técnicas del interruptor*

Elaborado por: Autor, 2024

Característica	**Detalles**
Rocket	
Tipo de interruptor	Basculante
Tensión nominal	12V o 24V DC
Corriente nominal	10A
Posiciones	2 posiciones: ON/OFF
Configuración	SPST o DPST
Dimensiones	20mm x 15mm
Material del cuerpo	Plástico o metal

Tipo de montaje	THD y SMD

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

3.1.11 Buzzer

Fue diseñado con el propósito de transformar la energía de corriente en sonido. Es utilizado como un sistema de indicación o advertencia y se usa en diversos métodos. (Castro, 2022)

Tabla 12. *Especiaciones Técnicas del buzzer*

Elaborado por: Autor, 2024

Característica	**Detalles**
Buzzer	
Tipo de zumbador	Electromagnético o piezoeléctrico
Frecuencia de resonancia	1 KHz a 4 KHz
Nivel de sonido	70 dB y 100dB
Tensión de operación	3V y 12V DC
Corriente de operación	20 mA
Dimensiones	Depende del modelo de buzzer
Material del cuerpo	Plástico o metal
Tipo de montaje	SMD y THD

Nota: Esta tabla detalla las características de cada elemento utilizado en el prototipo

Todos estos componentes eléctricos descritos fueron seleccionados y son de suma importancia para poner en ejecución el prototipo primero el funcionamiento de los sensores ultrasónicos con el buzzer que permite identificar obstáculos cerca de un rango de 4 metros y por último la conexión de la placa que tiene incorporados los diferentes componentes como el microcontrolador Esp32, el dispositivo ESP32-CAM, el MicroSD y el módulo DFPlayer Mini. Mientras tanto en algunas partes internas del prototipo se encuentran conexiones como el módulo de carga, y la batería de litio. Y en las partes externas del prototipo se encuentran un Jack estéreo, un pulsador y un interruptor para poner en marcha el funcionamiento del prototipo.

3.2 Interpretación del diseño

Para interpretar de manera correcta el diseño del prototipo se va a dividir en dos partes. La primera parte es el diseño elaborado de la placa que está desarrollado con fibra FR4 en esta sección irán conectados los componentes eléctricos y en la segunda parte implica las divisiones del prototipo que fueron elaborados de un material PLA que permite resguardar todo el tema eléctrico.

En la parte interior del prototipo se encuentra ubicada la placa donde se permite conectar los componentes eléctricos como el ESP32, ESP32-CAM y el DFPlayer Mini. La dimensión del prototipo es ideal para ir ubicada en la parte superior delantera de la cabeza lo que permite tener una mayor confiabilidad para su uso.

Dentro de la base del prototipo se encuentra internamente conectado un Jack estéreo para conectar auriculares, dos sensores ultrasónicos para las distancias de los obstáculos, un buzzer para los sonidos de las distancias detectadas cerca de los 4 metros y una batería litio con su respectivo módulo de carga para ayudar a cargar el prototipo.

3.3 Esquema eléctrico de la placa en Proteus

Primero se desarrolló el esquema eléctrico de la placa antes de realizarlo en PCB básicamente se incorporó los elementos eléctricos esenciales que necesita el circuito del prototipo como se indica en la figura 4.

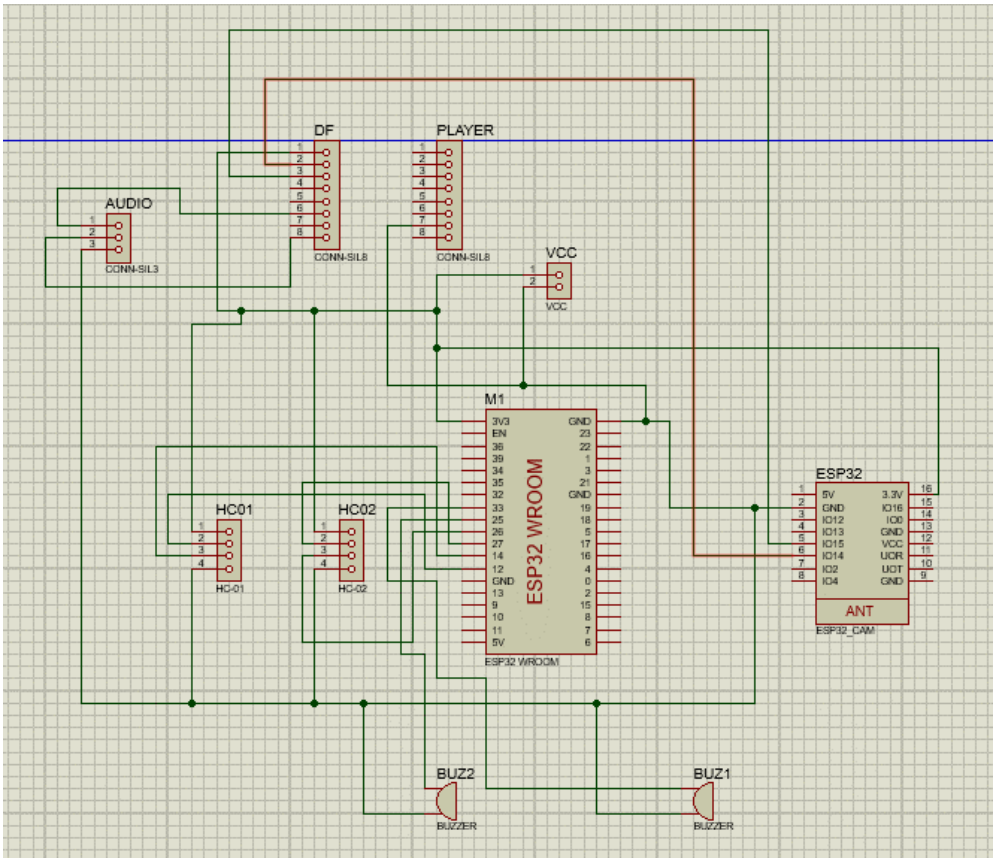


Figura 4. Esquema eléctrico en Proteus

Elaborado por: Autor, 2024

3.4 Elaboración del diseño PCB

Una vez realizado el esquema eléctrico se procede a realizar el diseño PCB en proteus donde van incorporados en la parte superior el DFPlayer Mini y el ESP32-CAM mientras que en la parte inferior va ubicado el ESP32. En la figura 5 se observa el diseño PCB

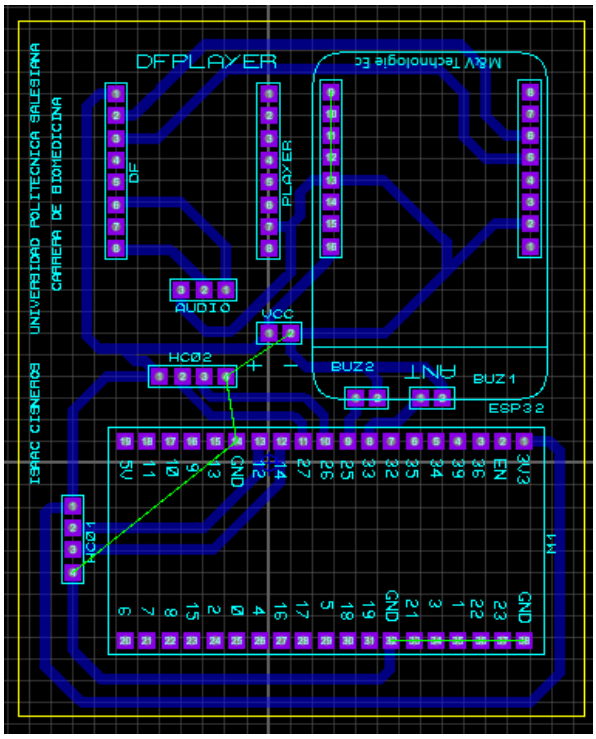


Figura 5. *Diseño PCB*

Elaborado por: Autor, 2024

La placa fue realizada con material de fibra FR4 lo que permite una buena resistencia mecánica y un excelente aislamiento eléctrico en la Figura 5 y 6 se puede observar la parte frontal y trasera de la placa PCB.

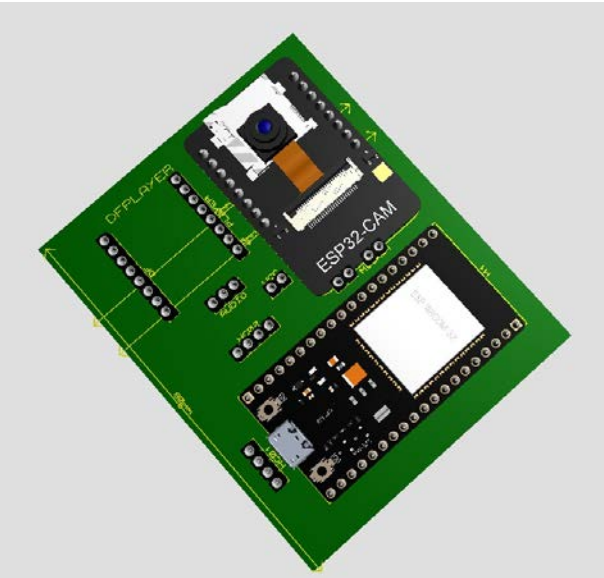


Figura 6. Vista frontal

Elaborado por: Autor, 2024

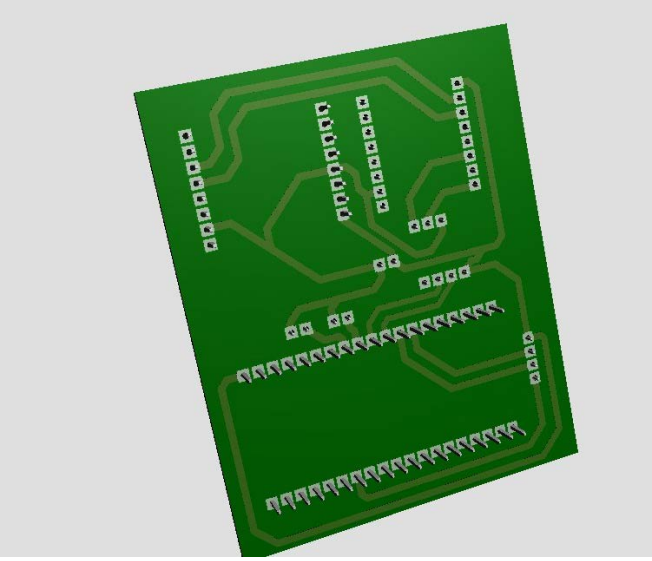


Figura 7. Vista trasera

Elaborado por: Autor, 2024

3.5 Diseño realizado del prototipo en SolidWorks

Se realizó el prototipo utilizando el instrumento de diseño SolidWorks que permite realizar bosquejos en tercera dimensión (3D) para detallar tanto la base como las tapas del prototipo. Este diseño cuenta con 5 tapas y con la base principal que permite un ensamblaje eficaz. De igual manera al momento de diseñar el prototipo se dejó unos pequeños huecos para poner pernos y poder asegurar correctamente todas las piezas del prototipo Mediante esta herramienta de diseño facilita la impresión del prototipo y es gran ayuda para respaldar de manera eficiente la operatividad del prototipo.

3.5.1 Aspecto de las tapas

El diseño de las tapas del prototipo se divide en 5 secciones:

Primera sección: Tapa Frontal

La primera sección está dada por la tapa frontal donde tiene un agujero cuadrado para calzar la cámara del ESP32-CAM y dos agujeros para que vaya el emisor y receptor del ultrasonido como se puede apreciar en la figura 8.

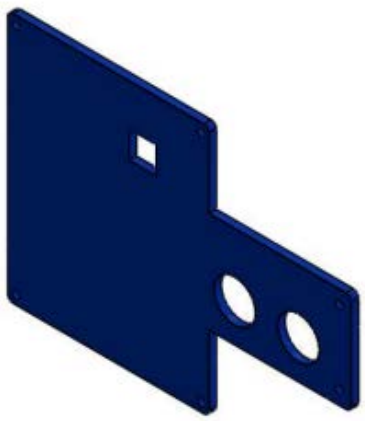


Figura 8. Tapa frontal

Elaborado por: Autor, 2024

Segunda Sección: Tapa lateral del primer ultrasonido

Esta segunda sección es una tapa para ajustar la parte lateral del primer ultrasonido donde se acomoda prácticamente con 4 pernos como se observa en la figura 9.

Figura 9. Tapa lateral

Elaborado por: Autor, 2024

Tercera sección: Tapa frontal del segundo ultrasonido

Esta tapa va ajustada en la parte frontal donde tiene 2 agujeros para el emisor y receptor del ultrasonido, esta parte de igual manera se acomoda con 4 pernos como se observa en la figura 10.

Figura 10. Tapa frontal 2

Elaborado por: Autor, 2024

Cuarta sección: Tapa lateral del segundo ultrasonido

La cuarta tapa del prototipo es para ajustar el segundo ultrasonido y se lo realiza con 4 pernos para dejarlo correctamente acomodado como se observa en la figura 11.

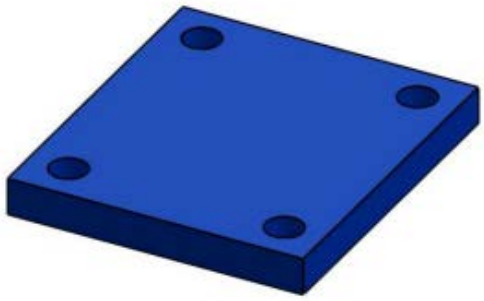


Figura 11. Tapa lateral 2

Elaborado por: Autor, 2022

Quinta sección: Tapa trasera del ultrasonido

Esta tapa es para ajustar la parte trasera del ultrasonido ya que justamente en esta sección también se encuentra el módulo de carga USB. Esta tapa es importante tanto para ajustar el ultrasonido y el módulo de carga como se observa en la figura 12.

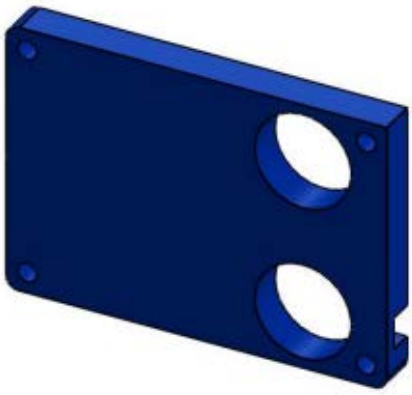


Figura 12. Tapa trasera

Elaborado por: Autor, 2024

3.5.2 Aspectos de la base

El diseño de la base fue exitosamente realizado para acomodar de manera precisa en la parte frontal la placa PCB, a los laterales del prototipo se tiene los ultrasonidos el primero está ubicado de forma rectilínea en esta misma sección de este ultrasonido se encuentra la batería de litio y el Jack 3.5mm para conectar los auriculares. El segundo ultrasonido se encuentra de manera inclinada, en esta misma sección se encuentra ubicado el módulo de carga USB para poder cargar el prototipo en el momento que se necesite. Además, en la parte frontal en sección superior tenemos dos pulsadores para el reconocimiento de objetos y el otro para las distancias, de igual manera un interruptor para el encendido y el apagado del prototipo como se observa en la figura 13.

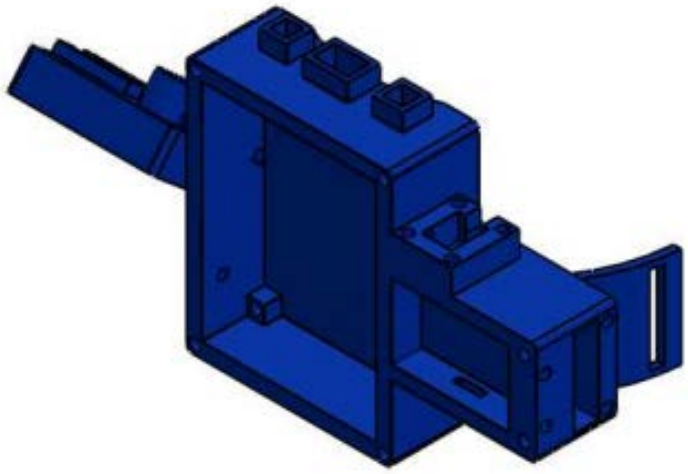


Figura 13. Base del prototipo

Elaborado por: Autor, 2024

3.5.3 Ensamblaje

Antes de terminar el diseño final se realizó el ensamblaje de cada parte y como debe de ir situado cada tapa. Como se observa en la figura 14 cada tapa del prototipo esta con un numeral que detalla cómo fue realizado este diseño de la estructura del prototipo.

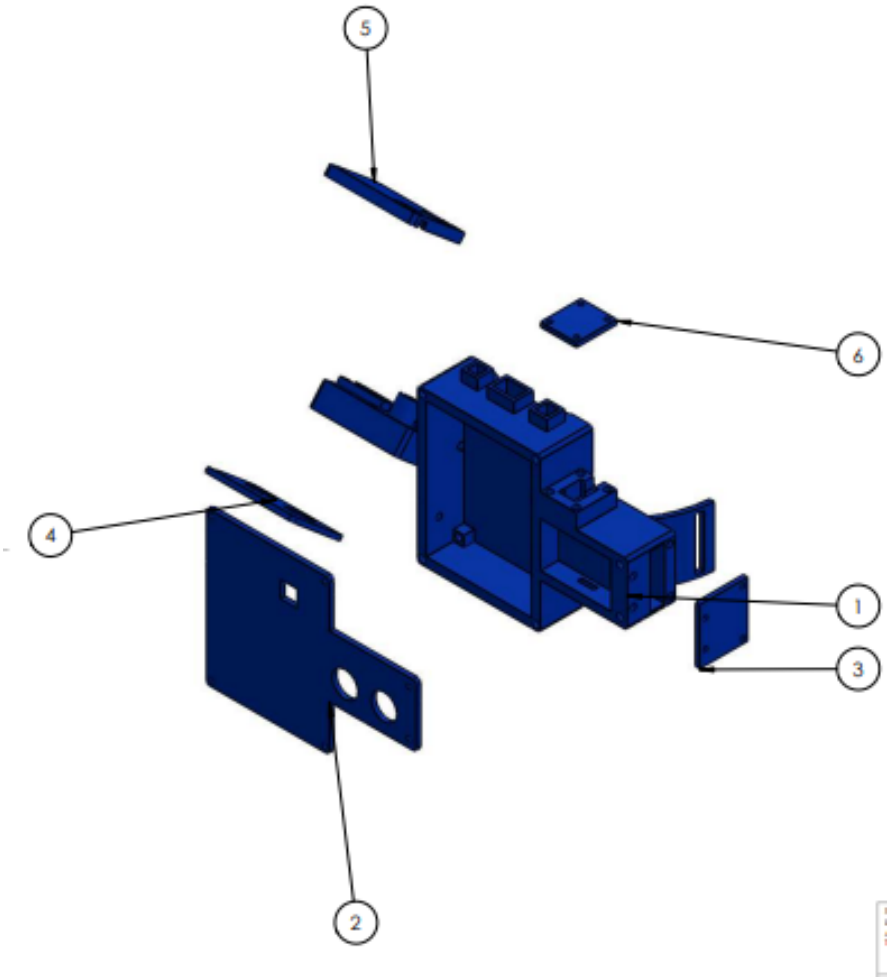


Figura 14. Ensamblaje

Elaborado por: Autor, 2024

A continuación, en la tabla 13 se indica el número y el nombre de pieza con el material que se fabrico

Tabla 13. *Pieza y material*

Elaborado por: Autor, 2024

Nº de elemento	**Nº de pieza**	**Material**
1	Cabeza	PLA
2	Tapa 1	PLA
3	Tapa 2	PLA
4	Tapa 3	PLA
5	Tapa 4	PLA
6	Tapa 5	PLA

Nota: Piezas y material resumido en una tabla

3.5.4 Estructura y diseño final del prototipo

Al final el diseño del prototipo quedo ajustado con su base y sus tapas. La base en la parte trasera tiene un segmento que se dejó abierto para poder pasar dos correas de nailon con velcro y de esa manera poder acomodar el prototipo de una manera eficiente en la cabeza del usuario. Mientras tanto en la parte frontal la placa PCB se calzo eficazmente el espacio que se dejó, además que los ultrasónicos y los demás componentes no tienen inconveniente al momento de comenzar a ensamblar y se utilizara una gran cantidad de pernos para ajustar todo el prototipo como se observa en la figura 15.

Figura 15. Diseño final

Elaborado por: Autor, 2024

Por último, el diseño final con las correas nailon y el velcro ya incorporadas queda de la siguiente manera como se observa en la figura 16.

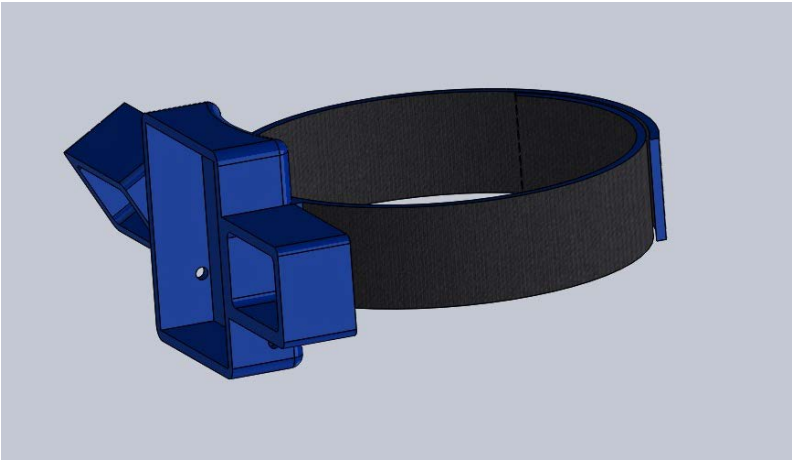


Figura 16. Diseño final con las correas Nailon y velcro

Elaborado por: Autor, 2024

Nota: Todo el prototipo fue impreso a excepción de las correas, las correas de Nailon fueron fabricadas por separado.

4. Capítulo 3: Desarrollo de software del prototipo de Asistencia visual

En este capítulo se brinda toda la información del desarrollo del software para el prototipo de asistencia visual. El propósito inicial es utilizar la plataforma Edge Impulse para la creación, el entrenamiento y la implementación de la red neuronal mediante la obtención de una base de datos de varias imágenes y el modelo de red neuronal convolucional que se usara para la correcta ejecución del software.

4.1 Edge Impulse

Como ya se explicó el Edge Impulse es una plataforma para el diseño o desarrollo de redes neuronales mediante una base de datos. Lo primero que se realiza es crear una cuenta para que nos permita utilizar el software de manera gratuita, aunque se utilizó la versión Trial que nos da todas las funciones del pagado por 15 días y nos permite trabajar de manera libre. En la figura 17 se observa la creación de la cuenta.

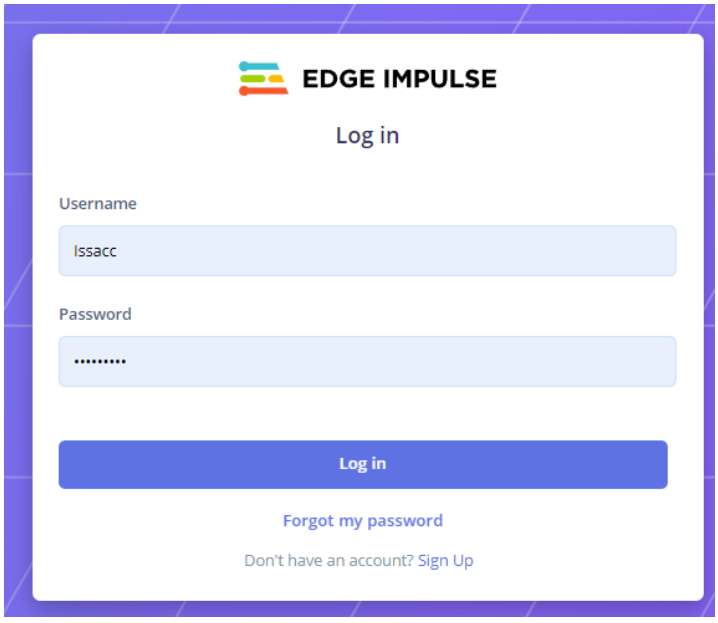


Figura 17. Login Edge Impulse

Elaborado por: Autor, 2024

4.1.1 Dashboard y adquisición de la base de datos

Para la recolección y creación de base de datos se realizó mediante la cámara del ESP32-CAM cabe recalcar que no es una cámara tan potente ya que únicamente cuenta con una resolución máxima de 1600 x 1200 pixeles es decir cuenta con 2 megapíxeles. Es por este motivo que las imágenes tomadas no son tan claras y toca tener en cuenta las condiciones de luz y otros factores ambientales. En la figura 18 se puede observar las opciones posibles para tomar fotos y crear la base de datos en este caso se escogió la tercera opción que es la placa de desarrollo en este caso la placa es el ESP32-CAM. Adicionalmente la primera opción nos da un código QR para poder tomar desde el celular fotos y crear la base de datos y en la segunda opción nos da la opción de tomar fotos a través de la cámara del ordenador.

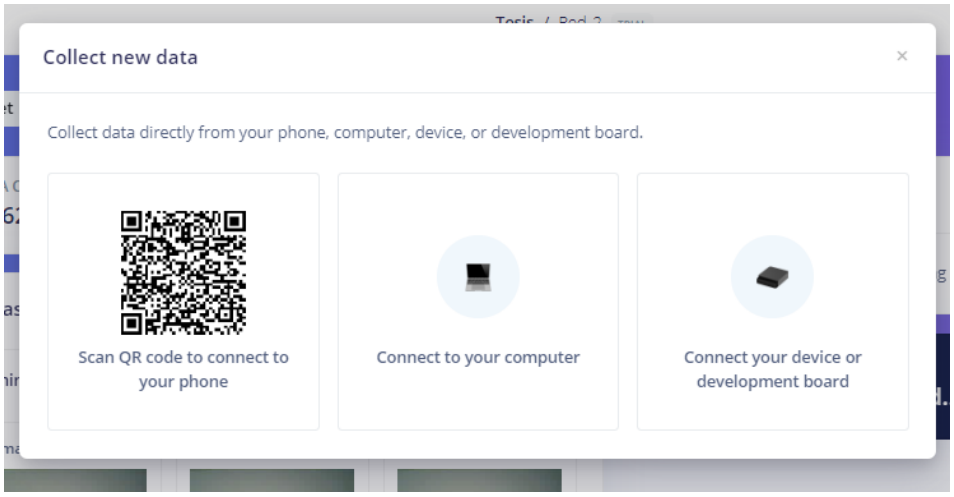


Figura 18. Selección de la placa

Elaborado por: Autor, 2024

Una vez seleccionada la placa ahora si se procede a tomar fotos de los diferentes objetos que necesita ir en la base de datos en este caso los objetos que se utilizo fue una goma, moneda, compas, borrador, tijera y lápiz los cuales son objetos dentro de un entorno educativo. Cabe recalcar que únicamente se reconocieron 6 objetos y esto es debido a la capacidad de la memoria RAM con la que cuenta el ESP32-CAM que es de 520KB no se puede llenar la memoria RAM con tantas

imágenes porque no se va poder generar la red neuronal en la que se trabajó. En la figura 19 se puede observar la imagen del objeto ya tomada foto encerrado con su label.

Figura 19. Imagen detallada de un lápiz y tijera con el label 1 y 2

Elaborado por: Autor, 2024

Nota: El label es una etiqueta identificadora que nos permite clasificar la imagen con diferentes nombres, en este caso solo se clasifico con números no con el nombre del objeto porque al rato de hacer una comparación de caracteres completos se tiene que reservar unos 100KB para hacer una relación de palabras completas.

En las siguientes figuras 20, 21 y 22 se van a observar los demás objetos que van hacer incorporados en la base de datos.

Figura 20. Borrador label 3 y compas label 4

Elaborador por: Autor, 2024

Figura 21. Moneda label 6 y goma label 7

Elaborado por: Autor, 2024

Figura 22. Regla label 8 y grapadora label 9

Elaborado por: Autor, 2024

De todas las imágenes presentadas solo se presentó una imagen de varias imágenes que fueron tomadas foto para subir a la base datos. En la figura 23 se presentan varias imágenes tomadas en diferentes ángulos y distancias de un objeto para entender el procedimiento del total de datos que existen.

Figura 23. Objeto tomado en diferentes ángulos

Elaborado por: Autor, 2024

En cuanto se termina de subir las imágenes se completa la base datos en esta ocasión fue un dataset de 1462 imágenes de datos recolectadas en donde:

- label 1 (175 imágenes)
- label 2 (231 imágenes)
- label 3 (197 imágenes)
- label 4 (210 imágenes)
- label 6 (190 imágenes)
- label 7 (171 imágenes).

En total se tiene un aproximado de 1174 imágenes para el entrenamiento y para las pruebas se ha escogido 288 imágenes es decir que se tiene un 80% de entrenamiento y 20% de prueba.

4.1.2 Diseño del impulso

En este apartado se tiene los datos de la imagen, la detección de los objetos(imágenes) y las características de salida. En la figura 24 se observa las diferentes opciones del impulso.

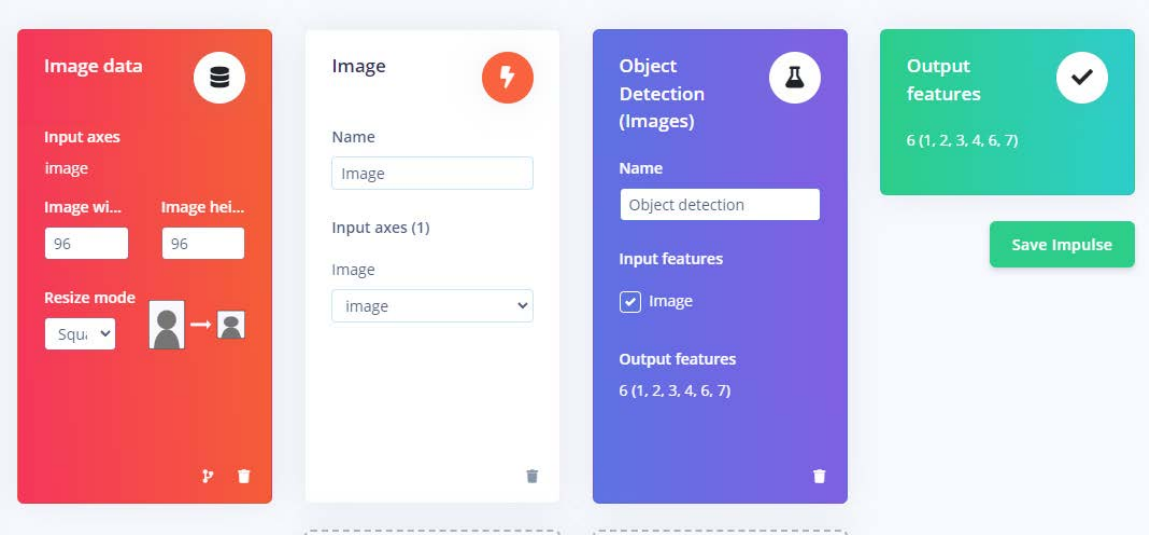


Figura 24. Diseño del impulso

Elaborado por: Autor, 2024

En el apartado de los datos de imágenes se puede determinar los ejes de entrada de la imagen. En donde se redimensiona todas las imágenes a las mismas dimensiones antes del entrenamiento. Se tiene la anchura y la altura a la que se redimensionan todas las imágenes en esta ocasión se tiene unas dimensiones de 96 x 96 de ancho y altura. Para el modo de cambio de tamaño se tiene el:

- Squash: “Aplastar” cambiará el tamaño mediante interpolación, utilizando toda la imagen. El eje más largo se verá aplastado.
- Fit shortest axis: “Ajustar al eje más corto” recortará primero los bordes del eje más largo a la relación de aspecto deseada, y luego interpolará al tamaño deseado.
- Fit longest axis: “Ajustar al eje más largo” recortará el eje más corto a la relación de aspecto deseada y, a continuación, interpolará.

De estas 3 opciones el que da un mejor diseño de impulso es el squash. A continuación, en la figura 25 se observa todas las opciones de los datos de imagen.

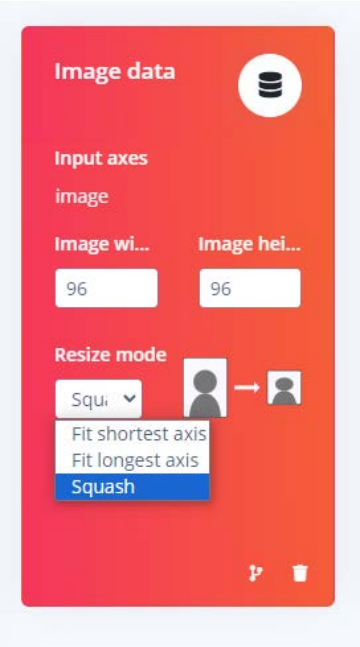


Figura 25. Datos de imagen

Elaborado por: Autor, 2024

En la segunda sección se debe añadir un bloque de procesamiento y se seleccionara el Edge Impulse ya que preprocesa y normaliza los datos de la imagen y, opcionalmente, reduce la profundidad de color. Como se observa en la figura 26.

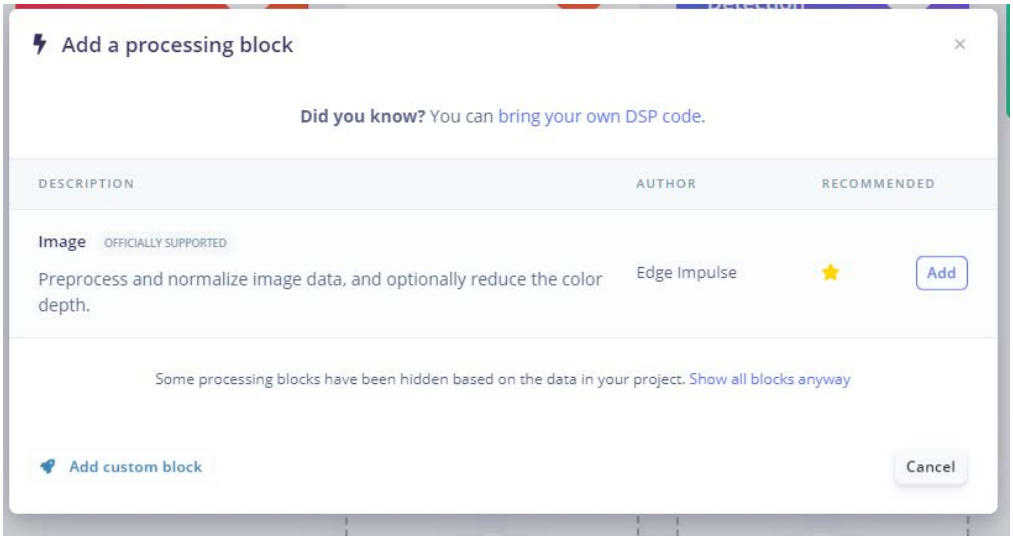


Figura 26. Bloque de procesamiento

Elaborado por: Autor, 2024

En la tercera sección se añade un bloque de aprendizaje nos da tres opciones a elegir y son las siguientes:

- Detección de objetos (imágenes): Ajusta con precisión un modelo de detección de objetos preentrenado en sus datos. Tiene un buen rendimiento incluso con conjuntos de datos de imágenes relativamente pequeños. (Edge Impulse, 2024)
- Detección de objetos (imágenes) - BrainChip Akida™: Ajusta con precisión un modelo de detección de objetos preentrenado en sus datos. Buen rendimiento incluso con conjuntos de datos de imágenes relativamente pequeños. Sólo funciona con la placa BrainChip AKD1000 MINI PCIe. (Edge Impulse, 2024)
- FOMO-AD (Imágenes): Detección visual de anomalías. Encuentra valores atípicos en datos nuevos. Extrae características visuales utilizando un modelo preentrenado en sus datos, y un modelo de mezcla gaussiana (GMM) modela la forma de las características utilizando

una distribución de probabilidad. Los datos nuevos que resultan improbables según este modelo pueden considerarse anómalos. (Edge Impulse, 2024)

En la figura 27 se puede observar los diferentes bloques de aprendizaje en el cual el más destacado para su uso es el “Object Detection (Images)” ya que es ideal para el microcontrolador ESP32-CAM y no consume mucha memoria RAM.

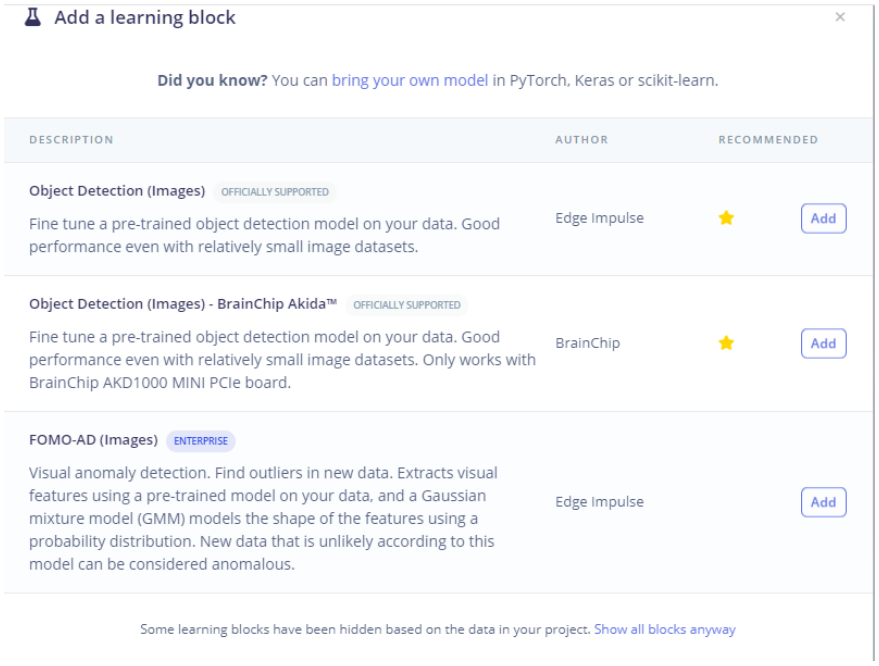


Figura 27. Bloques de aprendizaje

Elaborado por: Autor, 2024

En la última sección se indica las características de entrada y características de salida dependiendo de los objetos de detección que se incluyeron en la base de datos con sus respectivos nombres. Por ejemplo, aquí se tiene 6 objetos detectados y los nombres están con los siguientes numerales (1,2,3,4,6,7) en la figura 28 se observa estas características.

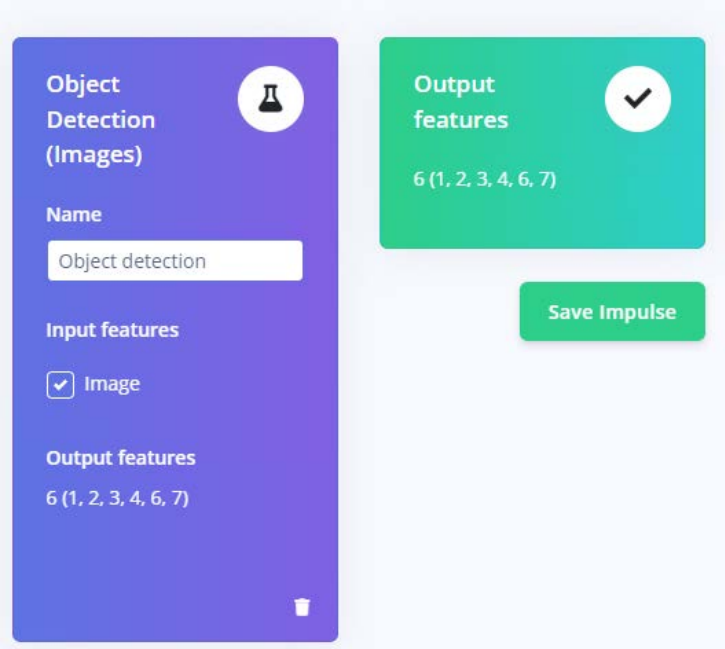


Figura 28. Características de entrada y salida

Elaborado por: Autor, 2024

4.1.3 Parámetros de la imagen

Se tiene los datos brutos de los diferentes objetos y se puede filtrar muestras por etiqueta como se indica en la figura 29.

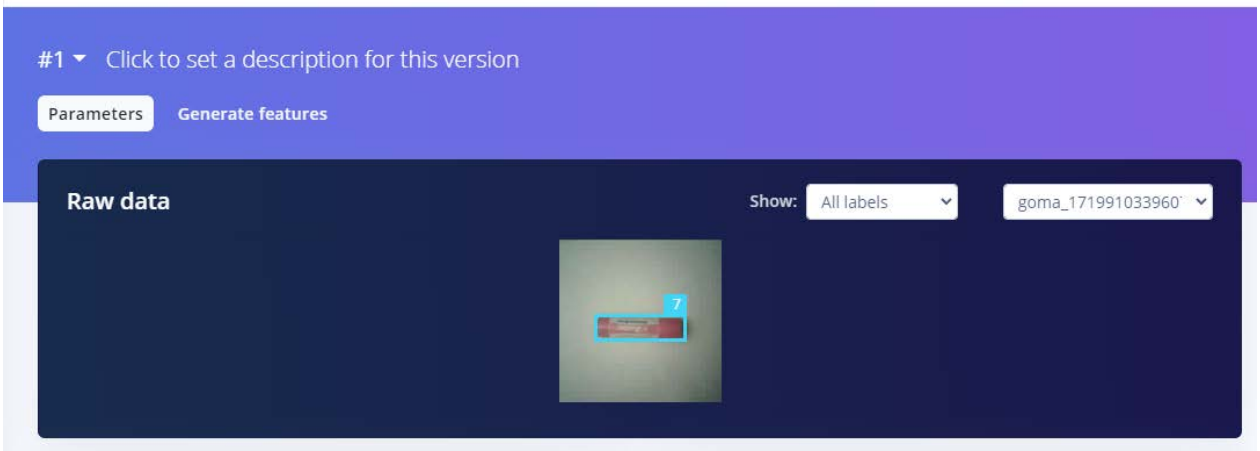


Figura 29. Datos brutos

Elaborado por: Autor, 2024

Después se obtiene varias características brutas de la imagen y los parámetros que se le aplican a la imagen como la profundidad de color para utilizar que se dividen en dos: RGB y Grayscale. Es recomendable usar el parámetro RGB porque son imágenes a color y detecta de mejor manera los objetos. Y por último se guarda los parámetros como se observa en la figura 30.

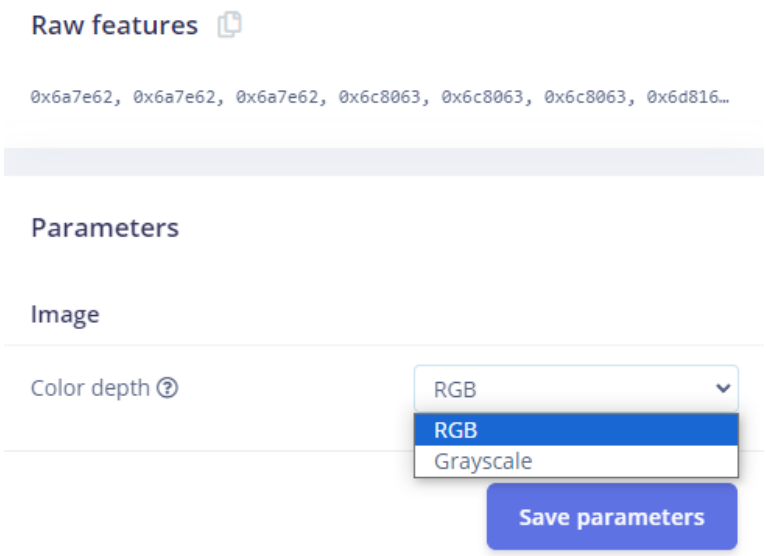


Figura 30. Parámetros de la imagen

Elaborado por: Autor, 2024

Luego se obtiene los resultados del procesamiento de señales digitales y las características procesadas de la imagen. Se espera que el rendimiento del ESP32-CAM tenga un tiempo de procesamiento de 15ms y el pico de RAM que no sea superior de los 4KB como se observa en la figura 31.

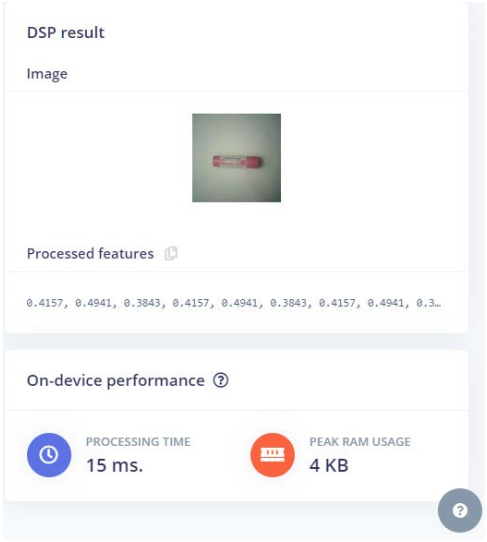


Figura 31. Características procesadas de la imagen

Elaborado por: Autor, 2024

Por último, se debe generar características comenzando con los datos del conjunto de entrenamiento que tiene en total y por las categorías de los objetos una vez que se genera las características inmediatamente nos indica el explorador de funciones donde mediante un gráfico nos indica los diferentes grupos de objetos representados por colores y nombres como se observa en la figura 32.

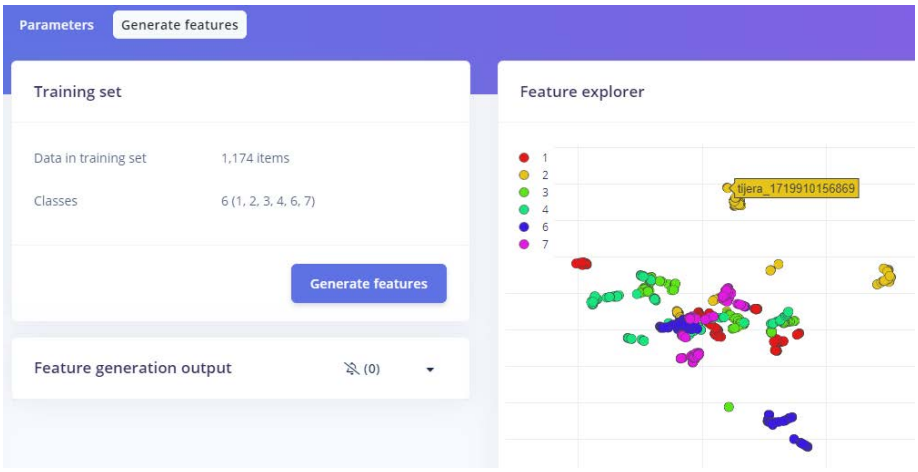


Figura 32. Generación de características

Elaborado por: Autor, 2024

4.1.4 Configuración para la detección de objetos

Se comienza con los ajustes de la red neuronal se debe adaptar de manera correcta los ajustes del entrenamiento. Se tendrá que configurar los siguientes valores:

- **Número de ciclos de entrenamiento:** La red se reentrena ve una cantidad de imágenes examina los resultados y compara manda un mensaje que no debe ajustarse y vuelve a entrenarse.
- **Utilizar optimizador aprendido:** Usa una red neuronal como optimizador para calcular los gradientes y la tasa de aprendizaje. Se recomienda utilizar una GPU y establecer un tamaño de lote grande.
- **Tasa de aprendizaje:** La velocidad de aprendizaje de la red neuronal. Si la red se adapta demasiado rápido, se debe reducir la velocidad de aprendizaje.
- **Entrena más rápido con la GPU:** Se recomienda la CPU para entrenar modelos en conjuntos de datos pequeños.

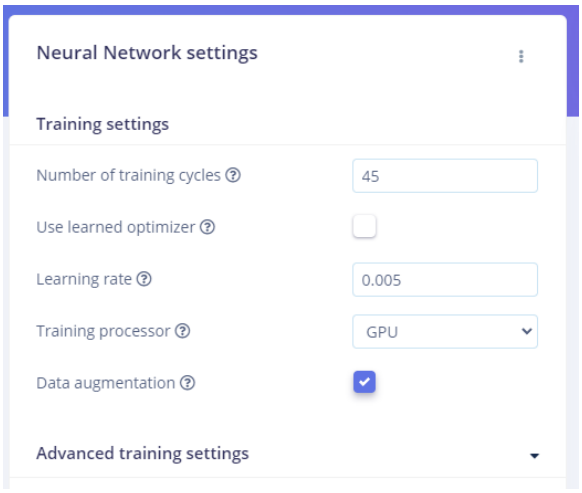


Figura 33. Configuración para la detección de objetos

Elaborado por: Autor, 2024

4.1.5 Arquitectura de redes neuronales

En primer lugar, se tiene las capas de entrada que son los datos de la base luego en las capas intermedias es donde se aplica el modelo de red neuronal convolucional que entre todas las opciones que hay para elegir el modelo acertado es el FOMO (Faster Objects, More Objects) MobileNetV2 0.1 que más adelante será explicado a profundidad y por último se tiene las capas de salida que son las clases o categorías de las imágenes ingresadas una vez ya con toda la estructura establecida se procede a iniciar el entrenamiento como se observa en la figura 34.

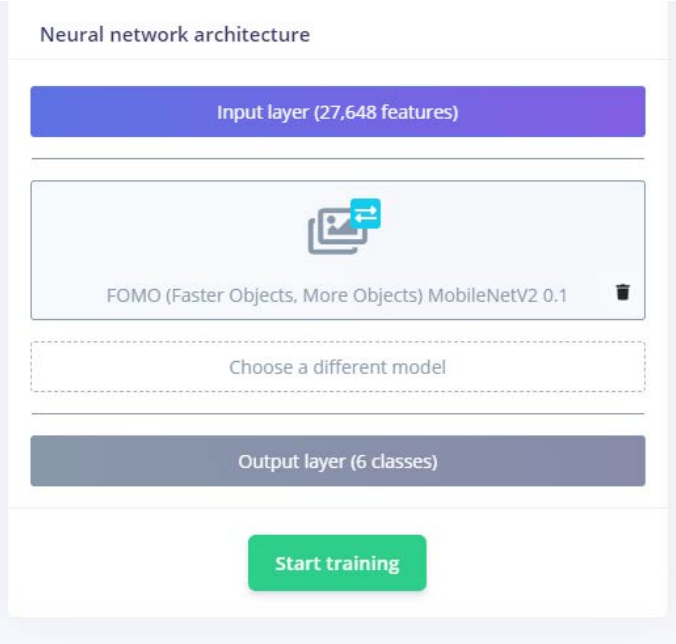


Figura 34. Arquitectura de redes neuronales

Elaborado por: Autor, 2024

4.2 FOMO (Objetos más rápidos, más objetos)

Es un algoritmo de machine learning que conduce la detección o identificación de objetos a equipos, dispositivos y herramientas sumamente limitados lo cual posibilita enumerar varios objetos y hallar su localización en una imagen que se encuentre en tiempo real usando un

aproximado de treinta ocasiones inferiores a la capacidad de procesamiento y de memoria que los algoritmos YOLOv3 y Mobile Net SSD. (Edge Impulse, 2024)

4.3 MobileNet

Son un tipo de modelos aptos para implementaciones de visión en teléfonos inteligentes. Se fundamentan en diseños elementales que usan curvas divisibles en intensidad o profundidad para armar redes neuronales ligeras y profundas. Todos estos modelos son productivos en una extensa serie de funcionalidades y situaciones de manejo incorporada la identificación de objetos, las características faciales y el geo-posicionamiento en una amplia dimensión. (Gonzales, 2023)

4.3.1 MobileNet V2

Es una red neuronal convolucional compuesta por nueve capas convolucionales las cuales son: (Flores, 2023)

- Una capa convolucional normal
- Una capa de convolución expansión
- 7 módulos bottleneck

Adicionalmente también cuenta con:

- Una capa de aplanamiento
- Una capa totalmente conectada
- Una función clasificadora SoftMax

4.3.2 Diagrama de bloques MobilNet V2

En la figura 35 se observa el diagrama de arquitectura del MobilNet V2.

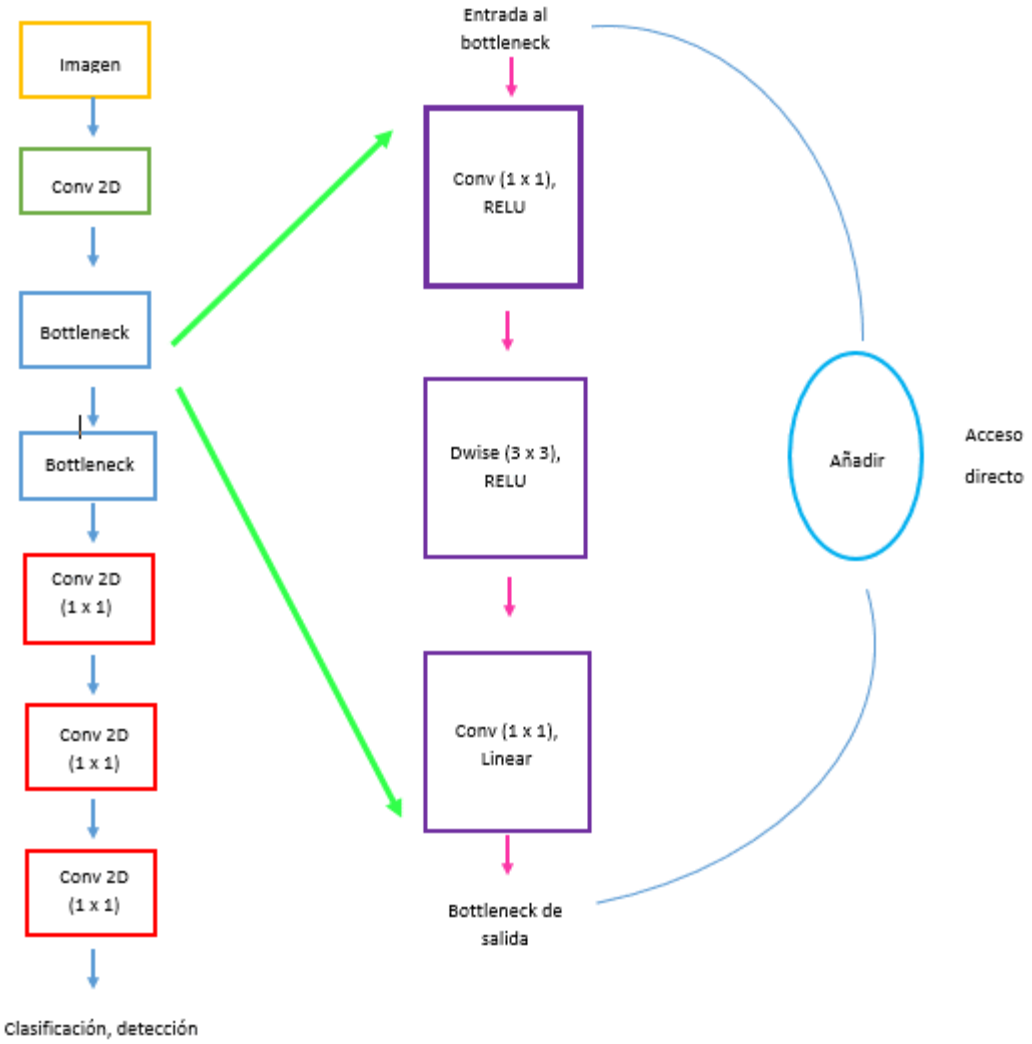


Figura 35. Diagrama de bloques MobileNet V2

Elaborado por: Autor, 2024

En la tabla 14 se observa minuciosamente la organización de este modelo implementado conforme a las capas que posee.

Tabla 14. *Organización del MobileNet V2*

Elaborado por: Autor, 2024

Dimensiones de entrada	**Dimensiones de salida**	**Capa**
224 x 224 x 3	112 x 112 x 32	Convolución clásica
112 x 112 x 32	112 x 112 x 16	Módulo bottleneck
112 x 112 x 16	56 x 56 x 24	Módulo bottleneck
56 x 56 x 24	28 x 28 x 32	Módulo bottleneck
28 x 28 x 32	14 x 14 x 64	Módulo bottleneck
14 x 14 x 64	14 x 14 x 96	Módulo bottleneck
14 x 14 x 96	7 x 7 x 160	Módulo bottleneck
7 x 7 x 160	7 x 7 x 320	Módulo bottleneck
7 x 7 x 320	7 x 7 s 1280	Convolución expansión
7 x 7 s 1280	1 x 1 x 1280	Capa de aplanamiento
1 x 1 x 1280	1 x 1 x 1000	Capa totalmente conectada
1 x 1 x 1000	1 x 1 x 1000	Softmax

Nota: Esta es la estructura de MobileNet V2 resumido en una tabla

Para entender de manera correcta todas estas capas a continuación es explicara detalladamente la función de cada una.

4.3.3 Módulo Bottleneck

Se mezclan las fases de transmisión, regulación o normalización y de activación en diferentes modulo conocidos como bottleneck creando únicamente una capa de red. Los módulos intentan incrementar la magnitud de frecuencias o canales para utiliza la menor cantidad de pesos y procedimientos en la recolección o extracción de características por medio de las convoluciones llamadas depthwhise. (López, 2022). En la figura 36 se puede observar mediante un diagrama de bloques la composición de los módulos.

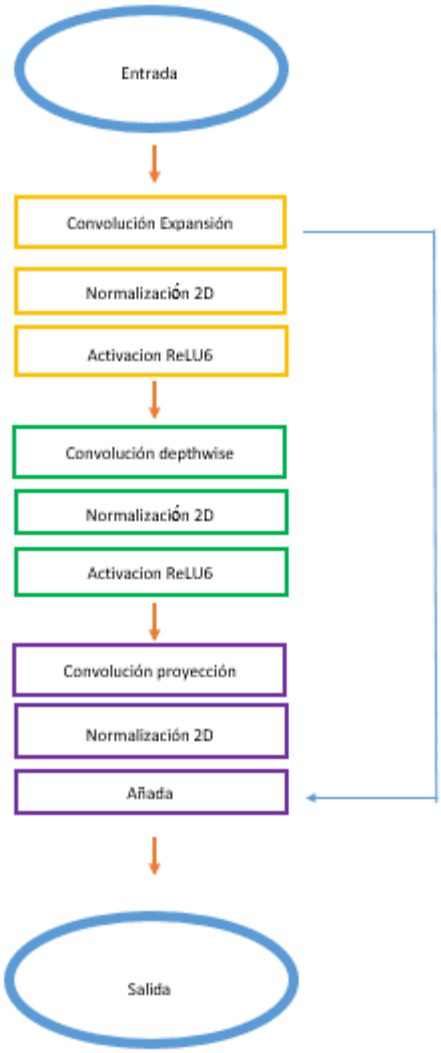


Figura 36. Diagrama de bloques de la composición de los módulos

Elaborado por: Autor, 2024

Todos estos módulos obtienen como inicio el mapa de propiedades adquirido por la capa previa y le adecuan una convolución expansión que incrementa la cantidad de canales, también una normalización en 2D y por ultimo una activación ReLU6 consiguiendo un reciente mapa. Una vez ampliada el total de canales del mapa de inicio los módulos usan el depthwise para obtener las características consecutivo del ciclo de normalización 2D y activación ReLU6 logrando una nueva partida hacia otro mapa de propiedades. Al final el modulo disminuye el total de canales por medio de convoluciones de proyección y de normalización 2D, adquiriendo el mapa de las características de partida. (Ruiz, 2024)

Convoluciones depthwise

Consta en llevar a cabo únicamente una convolución de dimensiones 3x3 para cada canal de inicio o entrada para lograr un canal de partida de manera que facilita el cálculo en la relación con las convoluciones clásicas donde es esencial ejecutar numerosas convoluciones como en canales de inicio para conseguir un canal de partida. En esta categoría de convoluciones el total de canales de inicio es similar al total de canales de partida. El modelo MobileNet V2 usa dichas capas para obtener las características de todos los mapas. (Pérez, 2022)

Normalización

Es un método empleado para optimizar la celeridad de entrenamiento y el equilibrio de las redes neuronales. En la situación del MobileNet V2 este procedimiento se usa respecto a cada mapa de propiedades para eludir los valores y prevenir que se incrementen. (García, 2023)

ReLU6

Es una ejecución de arranque usada en redes neuronales convolucionales detalladamente elaborada para limitar los valores de partida o salida que se tiene en una categoría en concreto generalmente suelen estar oscilando entre los valores de 0 y 6 como se observa en la figura 37. (Morales, 2024)

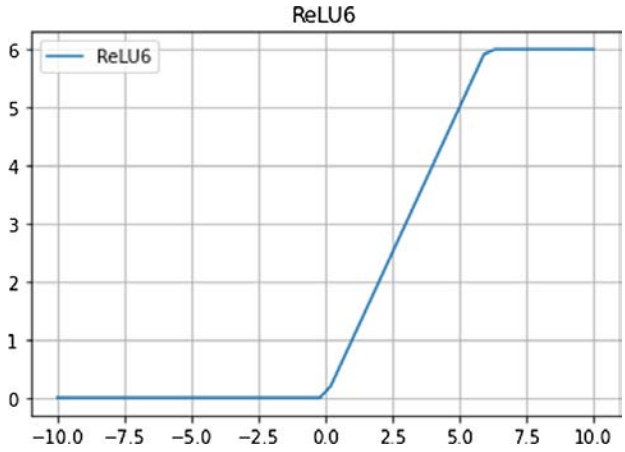


Figura 37. Relu6

4.3.4 Convolución de expansión y proyección

Estas capas son semejantes a la convolución clásica pero la gran discrepancia de esta es que logra llevar a cabo la convolución constantemente por medio de filtros de dimensiones 1 x 1 en lugar de K x K transformando la ejecución de convolución en únicamente incrementar la figura por un escalar para posteriormente agregar los resultados de cada canal de inicio para conseguir un canal de partida o salida. (Navarro, 2022)

4.3.5 Softmax

Es una operación de encendido usada específicamente en la capa de partida o salida de las redes neuronales para propiedades multivariable. Ofrece una salida que refleja una dispersión de

factibilidad lo cual simplifica el análisis de los resultados y la adquisición de determinaciones fundamentadas en la clasificación de las representaciones. (Hernández, 2024)

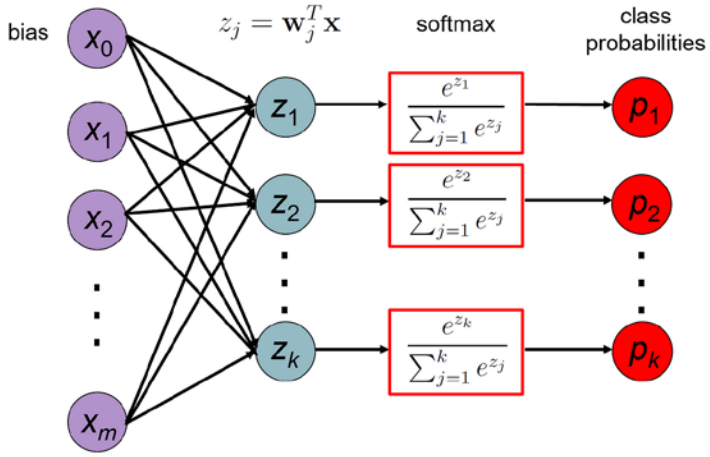


Figura 38. Interpretación Softmax

Una vez ya explicado en profundidad cada una de las capas que contiene la red neuronal convolucional se prosigue a entrenar el modelo una vez que el modelo este entrenado se imprimen métricas acerca del modelo trabajado.

4.4 Métricas y matriz de confusión del modelo

Se representa valores obtenidos luego del entrenamiento de la red neuronal donde nos indica F1 score, una mezcla de precisión y recall.

- **F1 Score:** Es el promedio armónico de precisión y recall por tanto se considera el valor de los dos en únicamente un evaluador. (Torres, 2022)
- **Precisión:** Este criterio o métrica es el volumen de situaciones aciertos positivos referentes a la cantidad absoluta de lo que es por totalidad positivo en conclusión es por completo el algoritmo que se pronosticó como positivo y se evalúa la veracidad. (Díaz, 2022)

- **Recall:** Esta métrica tiene una perspectiva distinta donde se relaciona la magnitud de situaciones o casos categorizados como aciertos positivos respecto al conjunto que verdaderamente era positivo. (Sánchez, 2023)

En la figura 39 se puede observar la matriz de confusión y las métricas obtenidas en el momento que se entrena la red neuronal con la base datos generada.

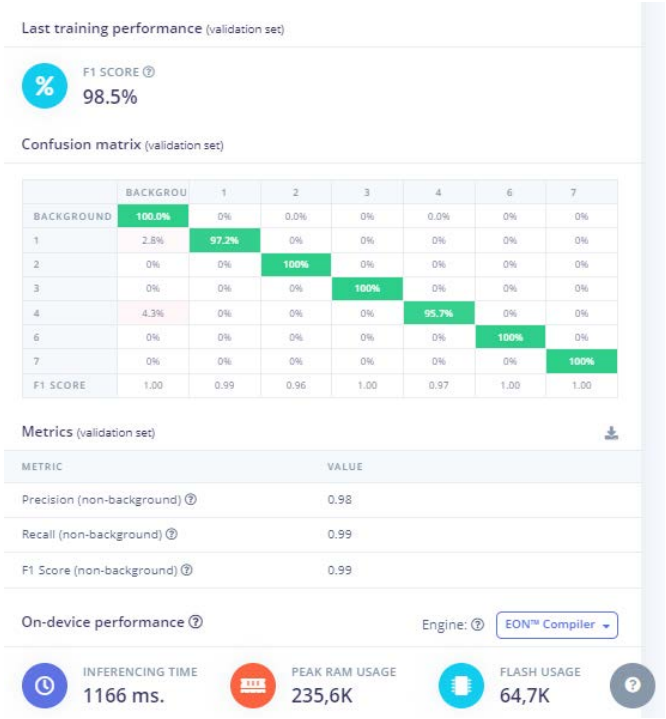


Figura 39. Métricas del modelo

Elaborado por: Autor, 2024

4.5 EON Tuner

Permite descubrir y elegir el más conveniente modelo de machine learning incorporado para una aplicación dentro de las condiciones del dispositivo o instrumento de destino. Examina la información de entrada, los probables bloques de procesamiento de señales y las estructuras de las

redes neuronales y proporciona un detalle global de las probables estructuras de modelos que se adaptan a las condiciones de intervalo y memoria del dispositivo o instrumento seleccionado en la figura 40 se puede observar diferentes predicciones realizadas con diferentes modelos.

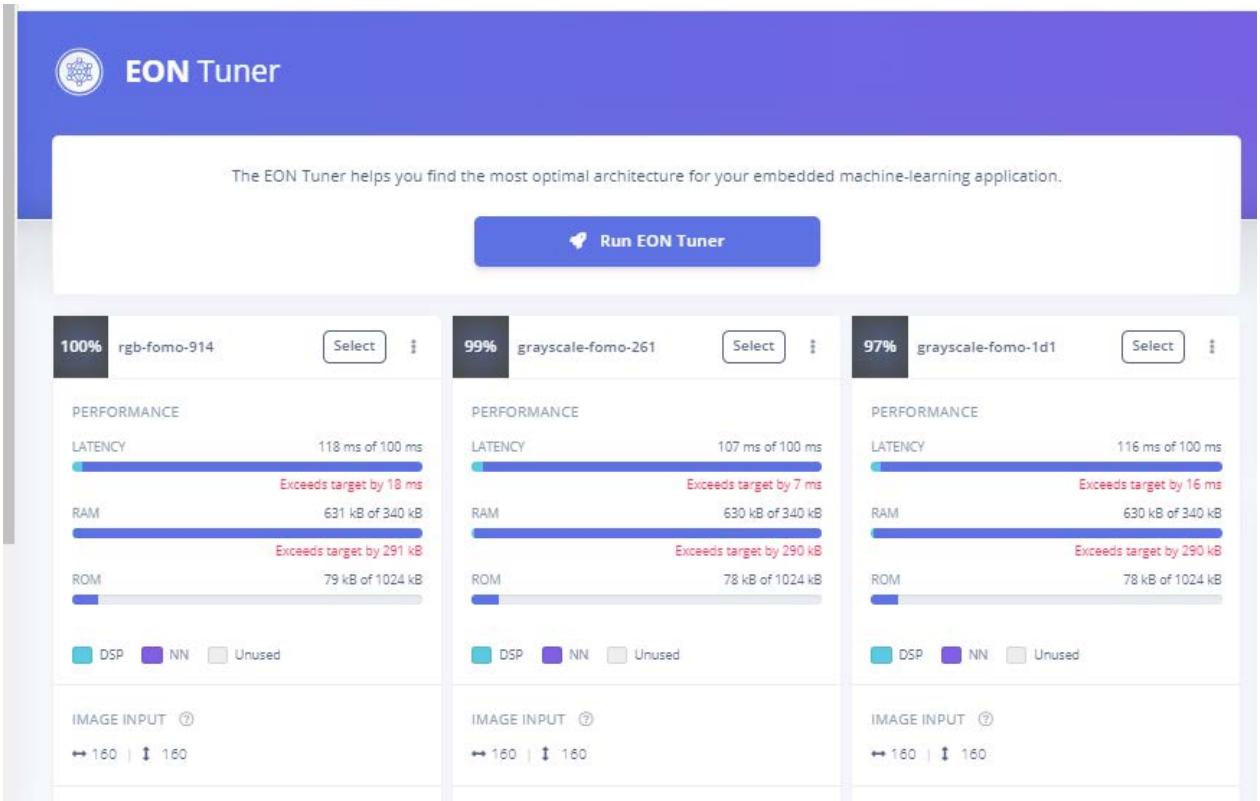


Figura 40. EON Tuner

Elaborado por: Autor, 2024

4.6 Configuración de despliegue

Se selecciona para que plataforma se va trabajar la red neuronal, en este caso la plataforma más utilizada por los usuarios es Arduino y al momento de elegir la plataforma se debe seleccionar el modelo de optimización que sea compatible con Arduino ya que si escogemos una que no sea compatible con Arduino no se permitirá ya que el espacio de memoria RAM se llena y lo ideal es trabajar con librería óptima para el modelo. Una vez ya configurado esta parte se procede a

construir y exportar el modelo a una biblioteca de Arduino que genera en un documento Zip esto nos permite configurar a nuestra manera el código generado por el Edge Impulse. En la figura 41 se observa la configuración de despliegue.

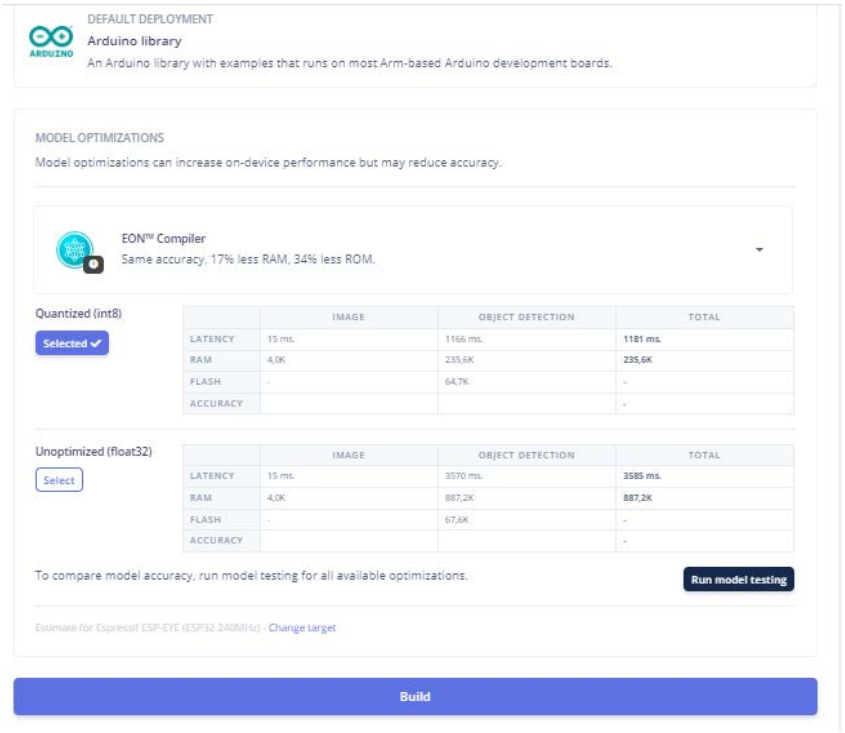


Figura 41. Configuración de despliegue

Elaborado por: Autor, 2024

4.7 Interfaz de usuario auditiva

Se utiliza un generador de voz de google para generar los sonidos, se descarga y se guarda en la tarjeta microsd que tiene el DFPlayer Mini, mediante una librería el DFPlayer mini permite manejar el audio a su manera una vez que ya están los audios en la microsd y DFPlayer mini se procede a enviar la información por medio de conexión serial con RX y TX que es la conexión que está realizada con la ESP32-CAM conectada en serie de ahí se manda la señal para reproducir los

sonidos, una vez realizado estos procesos se realiza la conexión del DFPlayer mini con el Jack 3.5mm para reproducir los sonidos generados por medio de auriculares.

4.7.1 Diagrama de bloques de la interfaz de usuario auditiva

En la figura 42 se observa el proceso para la reproducción de los sonidos por medio de una interfaz de usuario auditiva.

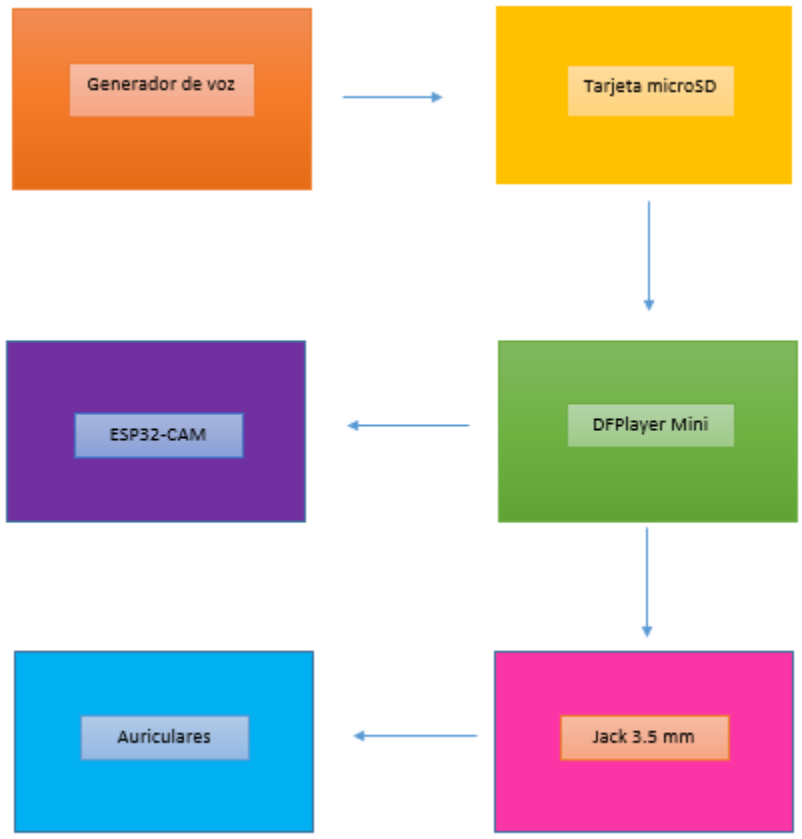


Figura 42. Diagrama de bloques de la interfaz de usuario auditiva

Elaborado por: Autor, 2024

5. Capítulo 4: Resultados

En este capítulo se aborda los resultados de todas las pruebas realizadas en las redes neuronales creadas con diferentes modelos y bloques de aprendizaje comenzando con el dataset inicial creado y realizando la comparación con el dataset final obtenido.

5.1 Resultados de proyectos creados

Se crearon varios proyectos hasta llegar a la red neuronal final cabe destacar que por cada red de neuronal y proyecto creado se creó varias cuentas ya que como es un trial solo dan 15 días de uso y durante esos días se intenta optimizar lo más que se pueda a la red neuronal. En la figura 43 y 44 se visualiza dos proyectos creados con diferentes cuentas en donde se optimizo la red neuronal. Un dato importante es que un proyecto se trabaja únicamente con imágenes en escala de grises y el otro proyecto creado se trabaja con imágenes en RGB.

Figura 43. Primer proyecto creado

Elaborado por: Autor, 2024

Nota: El primer proyecto fue creado en la versión gratis de Edge Impulse el cual no nos permite utilizar todas las funciones de la plataforma. Esta red neuronal fue principalmente realizada con imágenes en escala de grises.

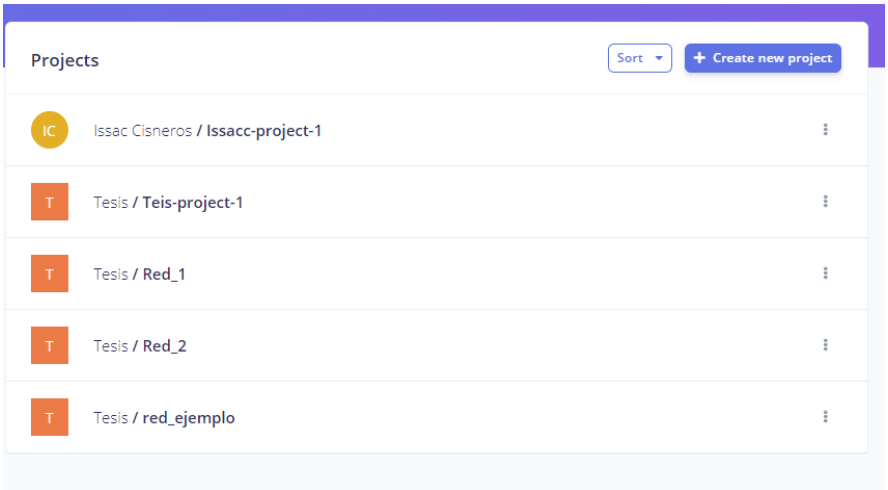


Figura 44. Proyecto final creado

Elaborado por: Autor, 2024

Nota: El proyecto final se creó mediante la versión trial del Edge Impulse que nos permite utilizar todas las funciones durante 15 días, pero la mayor desventaja es que pasado todos estos días ya no te deja visualizar absolutamente nada de lo que se trabajó en la red neuronal. Y la ventaja que nos da es optimizar hasta lo último la red neuronal dando resultados favorables.

5.2 Resultados de los datasets

5.2.1 Dataset de prueba

El primer dataset recolecto 1434 datos entre los cuales tenemos 8 categorías es decir objetos. En la tabla 15 se especifica los datos de recolección.

Tabla 15. Datos de recolección de dataset prueba

Elaborado por: Autor, 2024

Etiqueta	Objeto	Items
1	Compas	195
2	Cuchara	88
3	Esfero	78
4	Libro	172
6	Moneda	159
7	Peinilla	171
8	Tenedor	196
9	Tijera	88

5.2.2 Dataset Proyecto Final

El dataset final recolecto 1795 datos entre los cuales tenemos 9 categorías es decir objetos. En la tabla 16 se especifica los datos de recolección.

Tabla 16. *Datos de recolección*

Elaborado por: Autor, 2024

Etiqueta	Objeto	Items
1	Lápiz	175
2	Tijera	231
3	Borrador	197
4	Compas	210
6	Moneda	190
7	Goma	171
8	Regla	138
9	Grapadora	129

Nota: No existe una etiqueta 5 ya que en esa etiqueta se encontraba un objeto que causaba demasiados problemas en el dataset y tuvo que ser eliminado.

Una vez ya teniendo toda la base de datos es escoge datos de entrenamiento y datos de prueba. El primer dataset tiene una menor cantidad de datos a comparación del dataset final, en el primero solo se logró obtener 1434 datos para el entrenamiento se usó 1,147 datos de entrenamiento y en la prueba 287 muestras mientras tanto en el dataset final se escogió 1441 datos para el entrenamiento y para los datos de prueba se seleccionaron 354 muestras dándonos el total de 1795 datos. Lo que nos permite deducir que la dataset final es más completa y organizada. En la figura 45 se indica los datasets detalladamente.

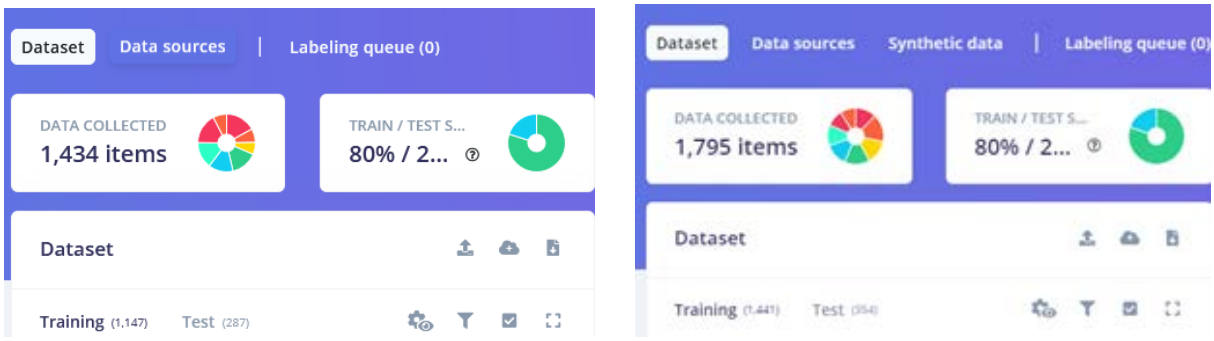


Figura 45. Primer Dataset y Dataset final

Elaborado por: Autor, 2024

5.2.3 Comparación de labels

En el primer proyecto los labels o etiquetas tenían los nombres de los objetos, pero mientras mas se iban aumentando los objetos y labels la capacidad de la memoria RAM se iba llenando lo que provoco que la ESP32-CAM se quede sin espacio y es por ese motivo que en el proyecto final no se nombró a las imágenes por su nombre en lugar de eso se les puso categorías por números del 1 al 9, con esto se logró reducir la memoria RAM del ESP32-CAM permitiendo agregar imágenes sin ningún problema. En la figura 46 se observa la comparación de los labels.

Figura 46. Comparación de labels

Elaborado por: Autor, 2024

Vista completa del dataset

Así se observa el dataset de una imagen en diferentes ángulos en el Edge Impulse como se indica en la figura 47.

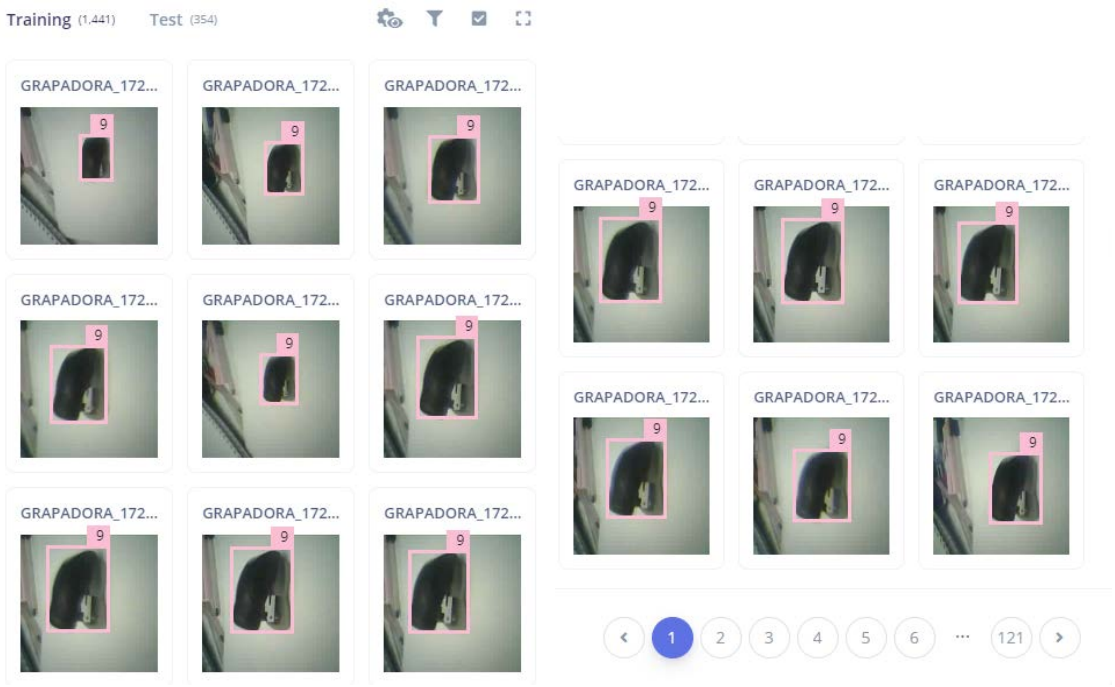


Figura 47. Imágenes en diferentes ángulos

Elaborado por: Autor, 2024

5.2.4 Resultados de crear el impulso

En el proyecto de prueba se ajustó los datos de la imagen con el ancho y la altura de 96x96 y se para el modo de cambio de tamaño se usó la opción de “fit short axis” que permite recortar primero los lados exteriores del eje más largo a la relación de aspecto deseada y, a continuación, interpolará al tamaño deseado. Sin embargo, no se obtuvieron buenos resultados con esta opción mientras que en el proyecto final se utilizó las mismas dimensiones de la imagen, pero se cambió la opción del modo de cambio de tamaño a “squash” que permite cambiar el tamaño mediante interpolación, utilizando toda la imagen. El eje más largo se verá aplastado con esta opción se logró mejores resultados. En la figura 48 se observa la comparación de los datos de imagen.

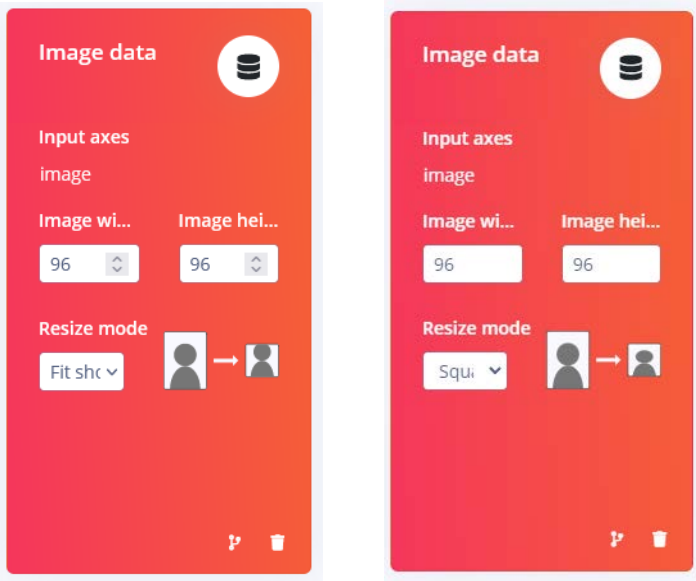


Figura 48. Resultados de los datos de imagen

Elaborado por: Autor, 2024

5.3 Resultados de parámetros y generación de características

En el primer dataset se obtuvo varias datos brutos y características es decir los datos extraídos en su estado original. En la figura 49 se poder observar los datos originales de la imagen.

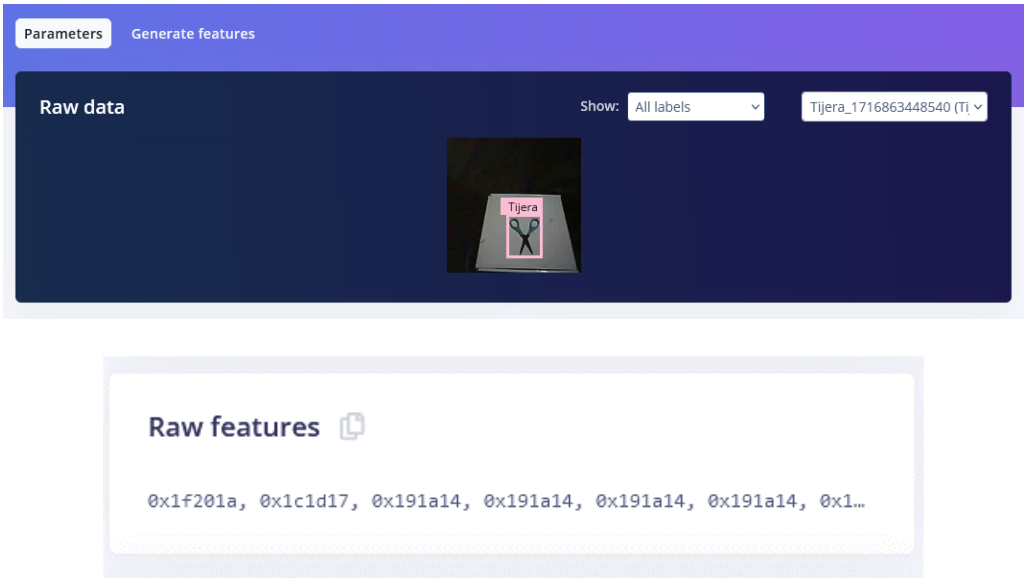


Figura 49. Datos y características brutas

Elaborado por: Autor, 2024

En la figura 50 se puede observar los datos y características brutas del dataset final.

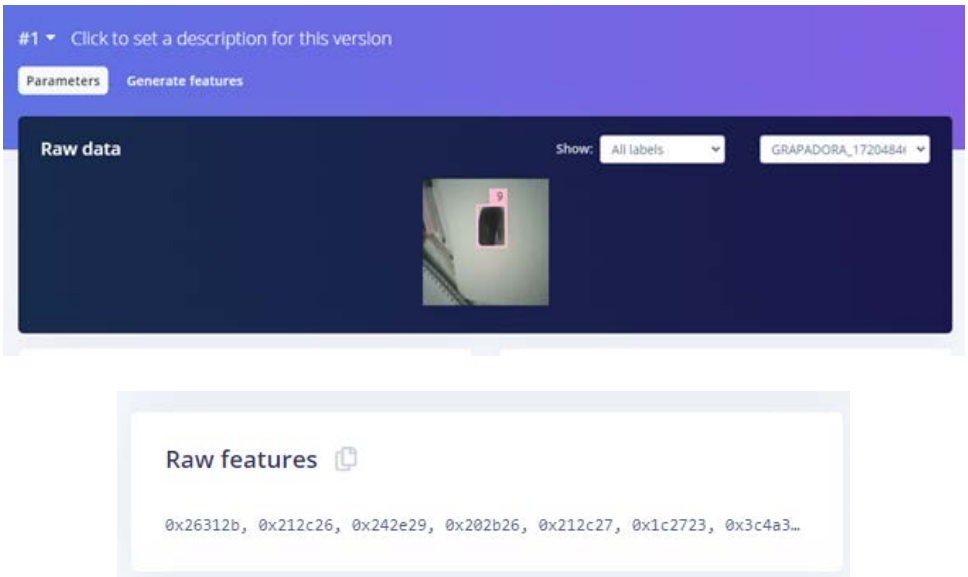


Figura 50. Datos y características brutas del dataset final

Elaborado por: Autor, 2024

5.3.1 Resultados de parámetros

En el primer dataset se usó imágenes en escala de grises dando como resultado más objetos en el dataset por ejemplo en este primer dataset entraron hasta 15 objetos pero se fueron eliminando debido a que tenían un margen de error muy alto pero el problema principal fue al momento de comenzar a reconocer los objetos ya que a cada objeto reconocía de manera errónea un objeto que no era el correcto mientras tanto en el dataset final se utilizó las imágenes a color dando como resultado menos objetos pero mejor reconocimiento por ejemplo en el dataset final fueron 8 objetos los que se logró tener y al momento de reconocer los objetos solo tuvo un pequeño margen de error pero obtuvo mejores resultados. En la figura 51 se observa los parámetros elegidos en cada dataset.

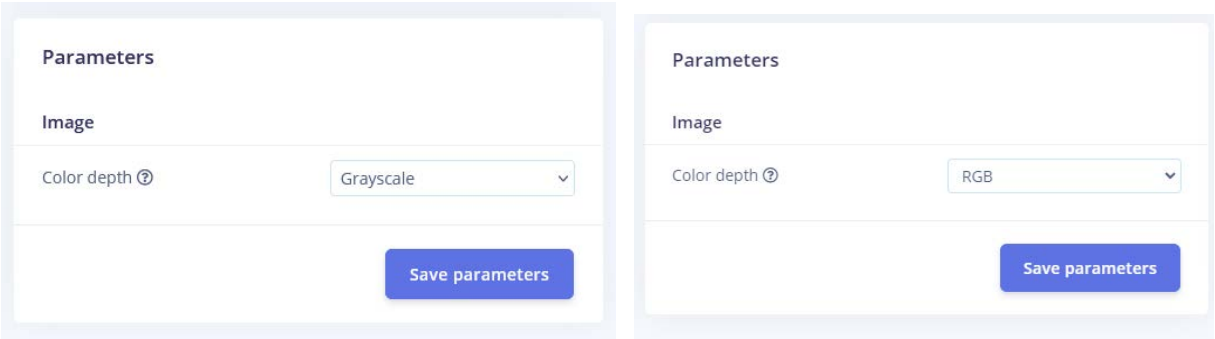


Figura 51. Parámetros dataset inicial y final

Elaborado por: Autor, 2024

5.3.2 Resultados del Procesamiento digital de señales y características procesadas

Una vez guardado los parámetros en primer dataset se realizó el procesamiento digital de señales esto básicamente sirve para mejorar la calidad de la imagen y extraer información significativa de cada imagen y al final se realiza las características procesadas de la imagen en donde da los resultados finales luego de emplear varias técnicas de procesamiento a los datos crudos. Cabe recalcar que rendimiento en el dispositivo tuvo 15m/s (milisegundos) en el tiempo de procesamiento y se usó 4KB de memoria RAM. Mientras tanto el dataset se aplicó el mismo procedimiento, pero en este caso nos dio diferentes características de la imagen y en el tiempo de proceso y memoria RAM ocuparon la misma cantidad. En la figura 52 se puede observar los resultados.

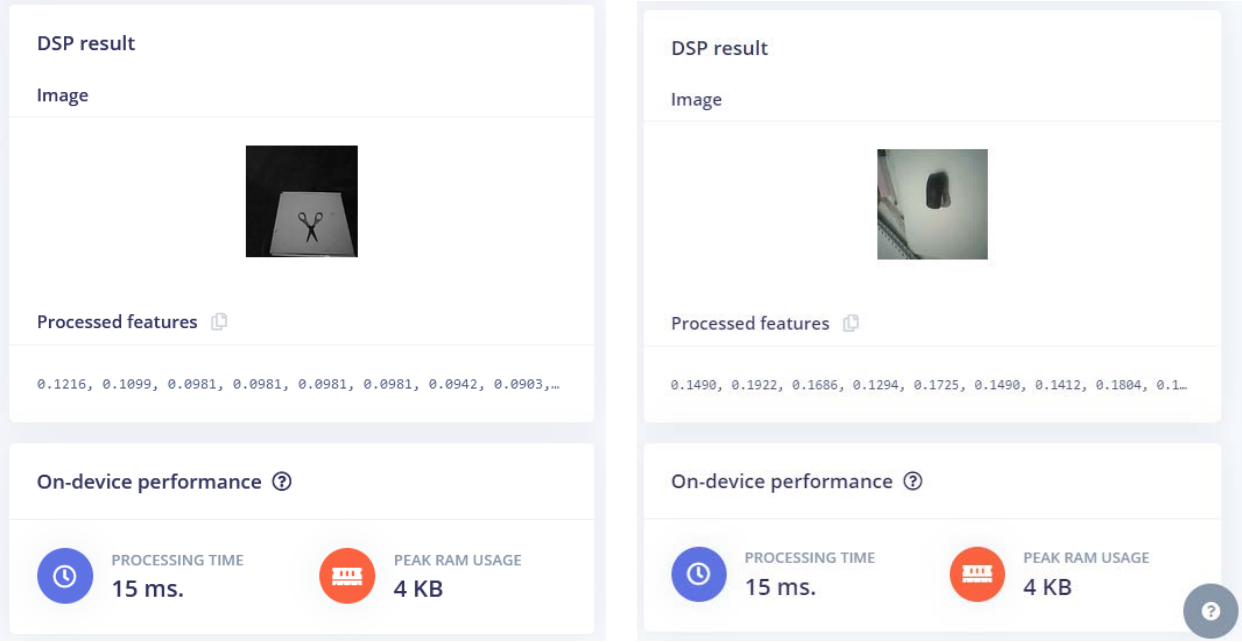


Figura 52. Resultados DSP

Elaborado por: Autor, 2024

5.4 Resultados de generar características

5.4.1 Comparación del conjunto de entrenamiento

Como se observa en la figura 53 se realiza una comparación entre los dos datasets en el primer dataset se puede observar que se trabajó con 1147 datos y cuenta con 8 clasificadores descritos con nombres como: compas, cuchara, esfero, libro, moneda, peinilla, tenedor y tijera. Mientras tanto en el segundo dataset se cuenta con 1441 datos y 8 clasificadores descritos con números enumerados del 1 al 9 a excepción del número 5 ya que se eliminó ese clasificador de un objeto que causaba ciertos errores.

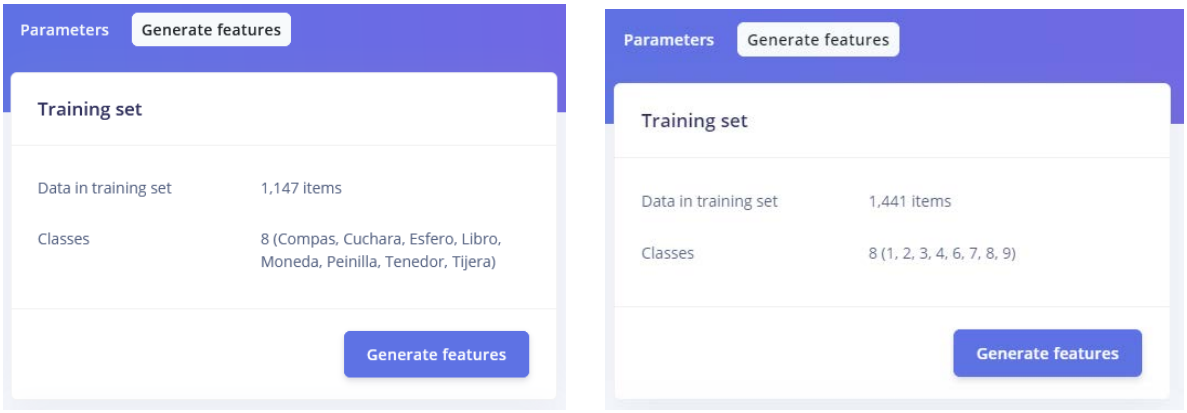


Figura 53. Comparación del conjunto de entrenamiento

Elaborado por: Autor, 2024

5.5 Resultados del explorador de características

Como resultado de la extracción de las características en los datos de entrada se pudo visualizar que en el primer dataset las características se agrupan por diferentes grupos y están esparcidos por diferentes lados lo cual significa que las características extraídas no están capturando de manera efectiva las señales mientras tanto el dataset final también se encuentran agrupados en grupos pero la mayoría de características están mezcladas lo que significa que tiene patrones o características distintas que son sencillamente reconocibles. En la figura 54 se observa la diferencia de los dos datasets en el explorador de características.

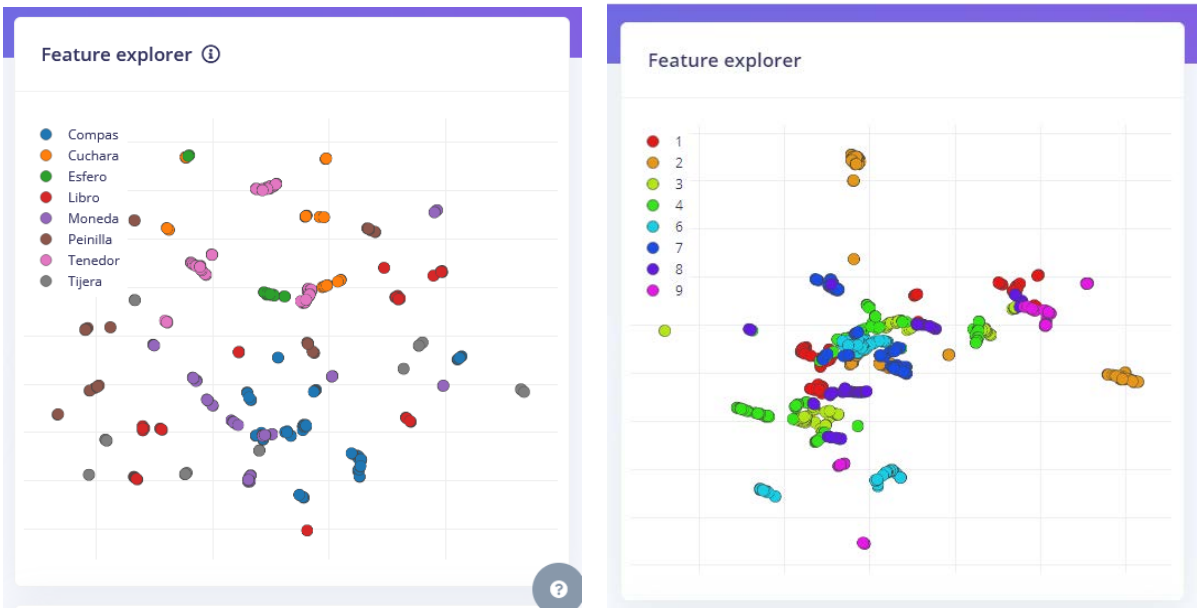


Figura 54. Resultados del explorador de características

Elaborado por: Autor, 2024

5.6 Resultado de la detección de objetos

5.6.1 Resultados del ajuste de la red neuronal

En la primera red neuronal se establecieron varios ajustes para entrenar la red neuronal se configuro el número de ciclos de entrenamiento en 40 iteraciones, la taza de aprendizaje en 0.001, el procesador de entrenamiento se usó el CPU y se empleó el aumento dando como resultado una precisión razonable y menos riesgos de oscilaciones sin embargo, se dio un problema con el sobreajuste ya que comenzó a memorizar los datos de entrenamiento en lugar de dar patrones útiles lo que provoco una productividad deficiente de los datos. Mientras tanto en la red neuronal final se configuro de otra manera ya tomando en cuenta los errores de la primera red neuronal, en la red final se configuro el número de ciclos de entrenamiento en 42 iteraciones, en la taza de aprendizaje se trabajó con 0.0005, el procesador de entrenamiento se usó el GPU y se empleó el aumento de datos dando resultados positivos ya que aumento la precisión de los datos, brindo una gran variedad de patrones útiles de los datos, el GPU fue demasiado útil ya que se empleó un dataset grande y

nos brindó un cómputo intensivo de los datos. En la figura 55 se observa los ajustes de las diferentes redes neuronales.

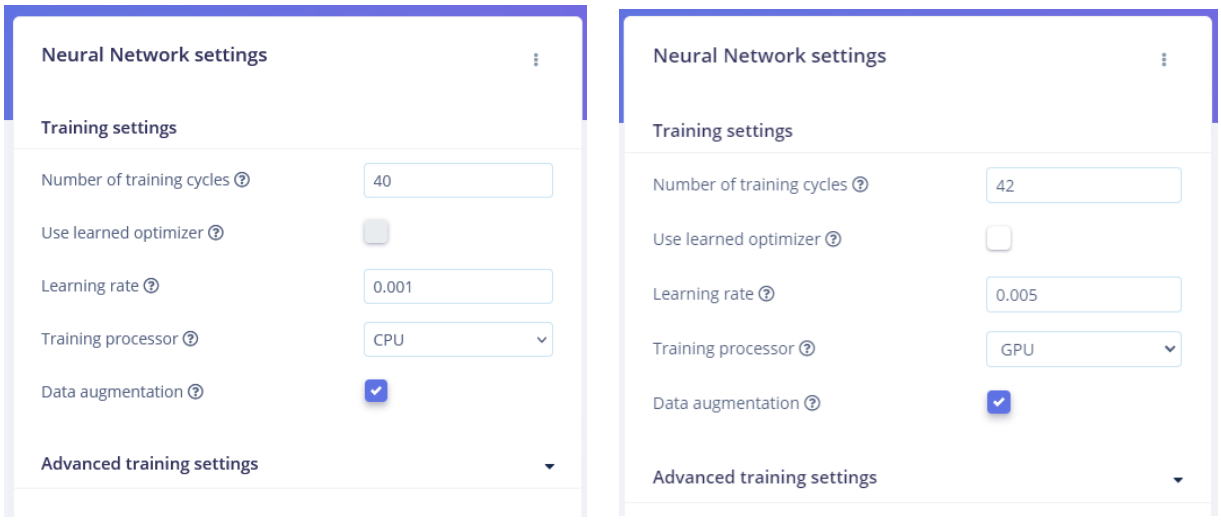


Figura 55. Resultados del ajuste de la red neuronal

Elaborado por: Autor, 2024

5.6.2 Resultados de la arquitectura de las redes neuronales

Como resultado de todas las configuraciones en la primera red neuronal se obtuvo 9216 características en la capa de entrada, en la capa intermedia se usó la red neuronal convolucional FOMO (Faster Objects, More Objects) MobileNetV2 0.35 el cual está diseñado para segmentar de forma gruesa una imagen en una cuadrícula de fondo frente a objetos de interés. Estos modelos están diseñados para tener un tamaño <100 KB y admiten una entrada en escala de grises o RGB a cualquier resolución, y por último en la capa de salida se obtuvieron 8 clases los cuales son los objetos detectados. En la red neuronal final se obtuvo 27,648 características en la capa de entrada, en este caso se usó en la capa intermedia la red neuronal convolucional FOMO (Faster Objects, More Objects) MobileNetV2 0.1 la cual se diferencia de la 0.35 por la velocidad y precisión en la detección de los objetos así que la versión 0.1 fue la mejor elección, y por último en la capa de

salida se obtuvieron de igual manera 8 clases. En la figura 56 se observa los resultados de las arquitecturas de las redes neuronales.

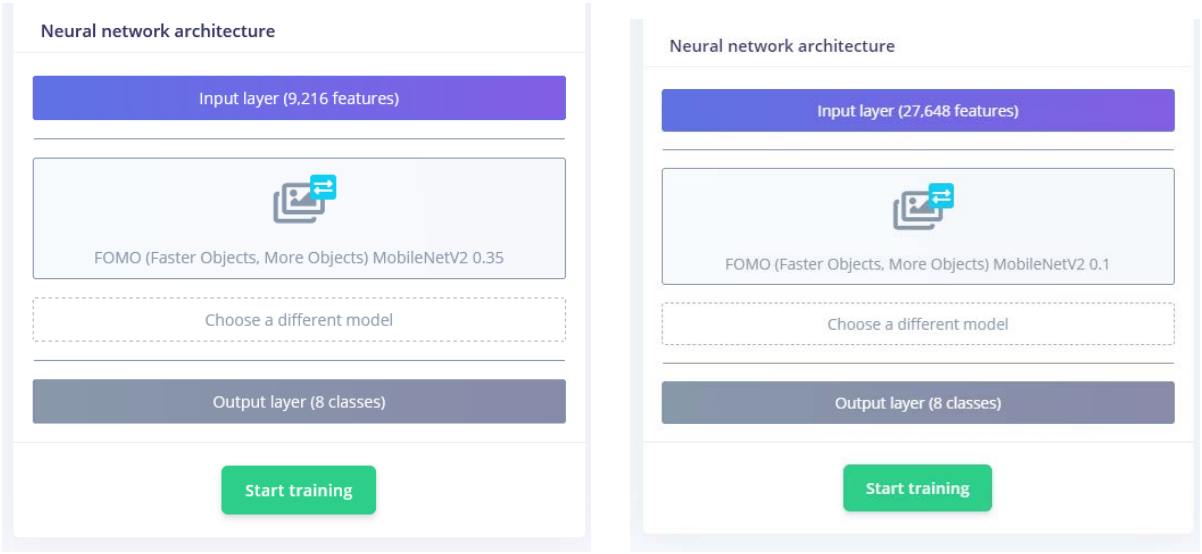


Figura 56. Resultados de arquitecturas de las redes neuronales

Elaborado por: Autor, 2024

5.6.3 Últimos resultados del entrenamiento (conjunto de validación)

En el primer modelo de red neuronal se dio un F1 Score es decir un promedio de precisión y recall de aproximadamente 100% pero como resultado este porcentaje no está correcto ya que se presentó un sobreajuste esto quiere decir que el modelo se memorizo los datos de entrenamiento lo cual es incorrecto porque al momento de reconocer objetos no va a identificar absolutamente nada.

En el segundo modelo de red neuronal se dio un F1 Score de aproximadamente de 95% este porcentaje es correcto ya que en la matriz de confusión que se lo representa existen mínimos errores en las etiquetas y predicciones, pero la precisión tuvo buenos resultados. En la figura 57 se puede observar los resultados del modelo.

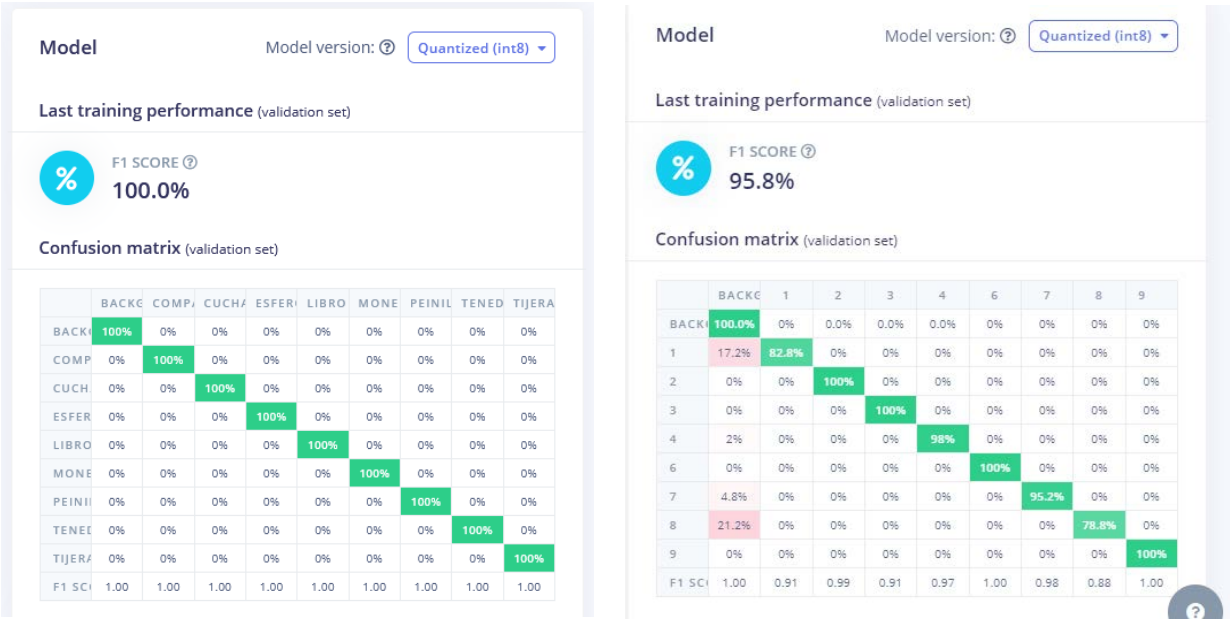


Figura 57. Resultados de entrenamiento

Elaborado por: Autor, 2024

5.6.4 Resultados de métricas (conjunto de validación)

En todas las métricas del primer modelo son perfectas pero debido al sobreajuste mientras tanto la métrica del segundo modelo tiene cierto error, pero es más preciso y no tiene sobreajuste. En la figura 58 se observa el resultado de las métricas.

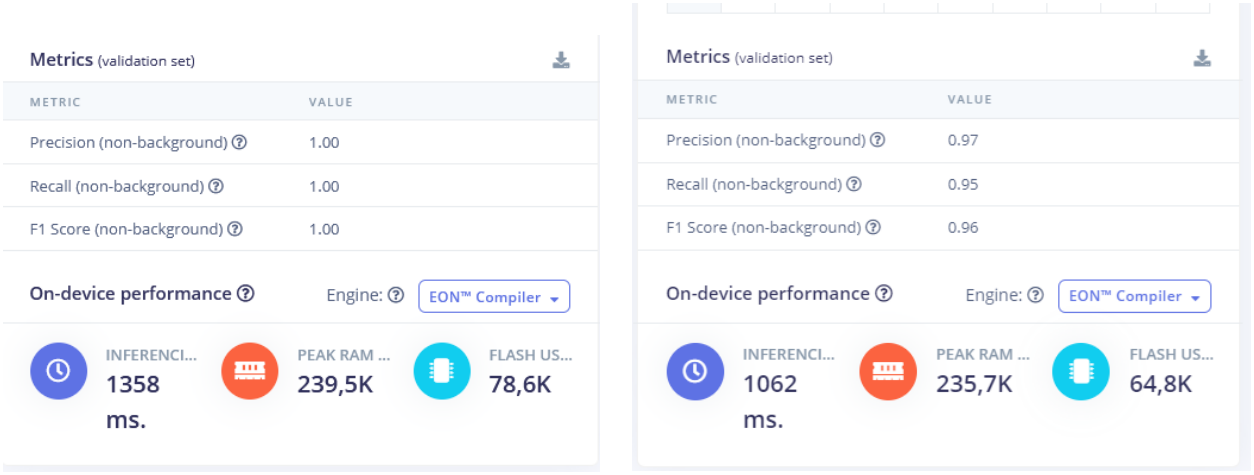


Figura 58. Resultados de métricas

Elaborado por: Autor, 2024

A continuación, en la tabla 17 se presenta una tabla comparativa de las métricas del modelo

Tabla 17. Tabla comparativa

Elaborado por: Autor, 2024

Métrica	Dataset 1 (Sobreajuste)	Dataset 2 (Mejor Propagación)
Precisión	1.00	0.97
Recall	1.00	0.95
F1 Score	1.00	0.96

Nota: Comparación de los resultados de cada métrica

- En la precisión del dataset 1 tiene resultados positivos, pero es debido a que esta memorizando los datos del modelo mientras tanto el dataset 2 tiene una mejor propagación de resultados, aunque posee errores mínimos.
- En el recall del dataset 1 identifica correctamente los datos positivos, pero es debido a que memoriza los datos mientras tanto en el dataset 2 indica que no está sobreajustado y tiene mínimo error de datos lo que indica que tiene espacio para mejorar los datos.
- En la última métrica F1 score indica que el promedio de la precisión y recall es perfecta, pero esto es debido por el sobreajuste mientras tanto el en dataset 2 existe un buen equilibrio entre la precisión y el recall teniendo una mejor propagación de datos.

5.7 Resultado final

5.7.1 Configuración de despliegue

Al final después de todo el proceso realizado se puede desplegar el impulso en cualquier dispositivo. Esto hace que el modelo funcione sin conexión a Internet, minimiza la latencia y se

ejecuta con un consumo de energía mínimo. Es decir que la red neuronal entrenada tensor flow que se generó se transforma en una librería en este caso para las dos redes neuronales se utilizó Arduino.

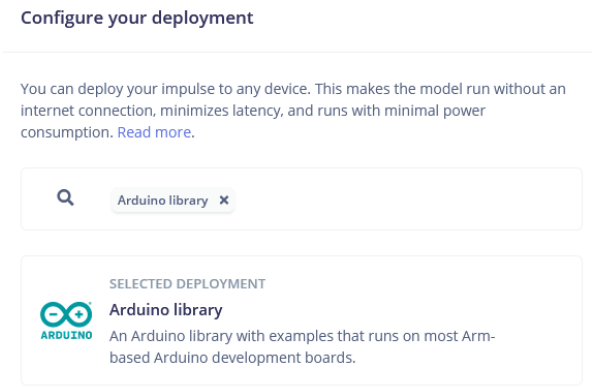


Figura 59. Configuración de despliegue

Elaborado por: Autor, 2024

5.7.2 Optimización del modelo

Las optimizaciones del modelo pueden aumentar el rendimiento en el dispositivo, pero pueden reducir la precisión. En el caso de las dos redes neuronales se usó Eon Compiler porque no ocupo mucha memoria RAM y la latencia fue estable.

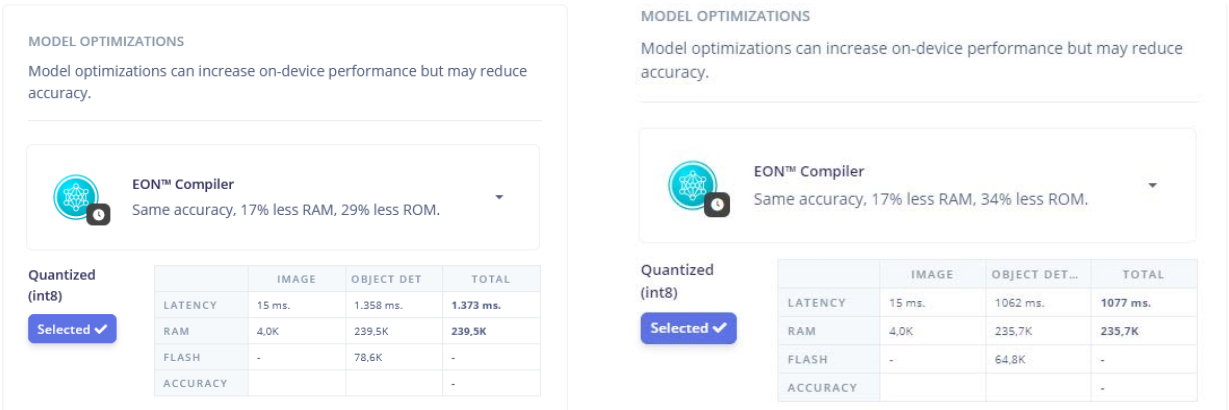


Figura 60. Optimización del modelo

Elaborado por: Autor, 2024

5.7.3 Construcción y generación del modelo final

Por último, se construye la red neuronal que se encontraba en modelo tensor flow se la pasa a una librería Arduino generando el código en un zip y poder cargar en la plataforma Arduino.

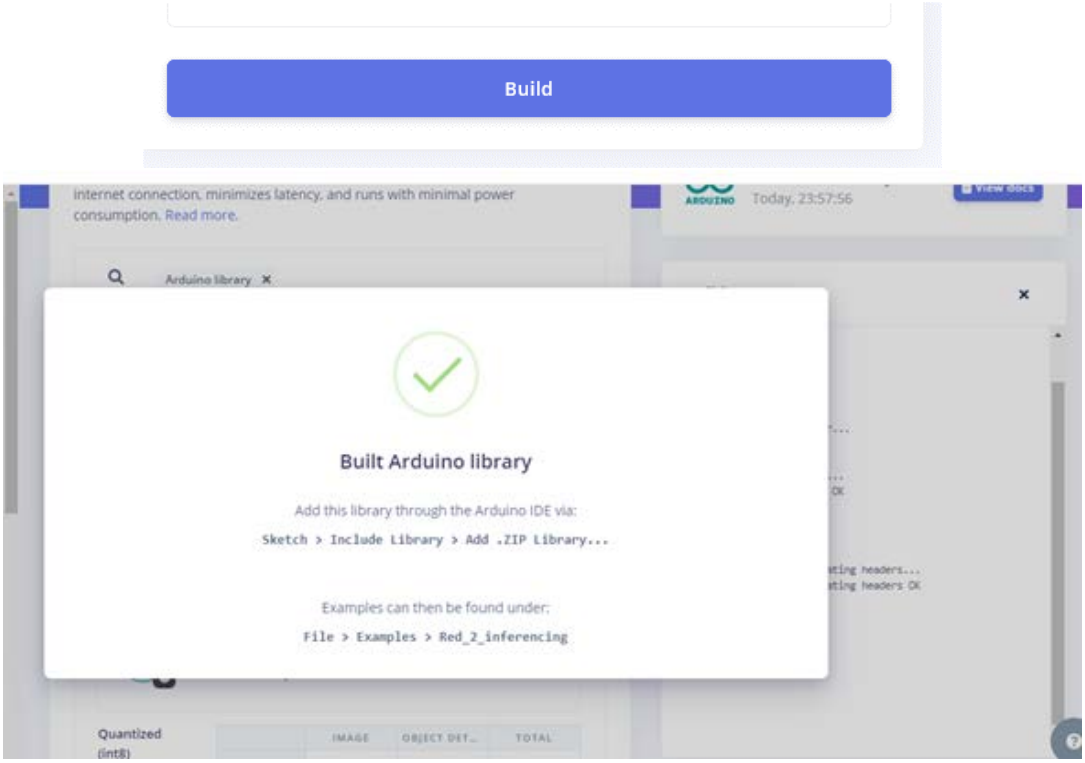


Figura 61. Construcción y generación del modelo final

Elaborado por: Autor, 2024

5.8 Pruebas de predicción con diferentes modelos (EON Tuner)

Se realizó varias pruebas de predicción para escoger el modelo perfecto que trabaje en conjunto con la base de datos y sobre todo que no ocupe toda la memoria RAM del ESP32-CAM.

5.8.1 Predicción N° 1

En esta primera predicción da como resultado un 100% de precisión lo que indica que es un modelo muy estable, esta se realizó con el modelo FOMO y con imágenes en RGB (color) pero se puede observar que en la parte de rendimiento no cumple con los requisitos mínimos establecidos por la placa ya que tiene una latencia de 118ms y solo soporta hasta los 100ms, además se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. Por ultimo en la parte de la memoria RAM nos dio 631KB de 340KB utilizables por lo cual se excede alrededor de unos 291KB dando como conclusión no utilizar esta predicción de modelo porque no cumple con los requisitos de latencia y memoria RAM, pero tiene una excelente precisión para el modelo. En la figura 62 se observa la primera prueba de predicción este proceso se llevó a cabo en el EON Turner.

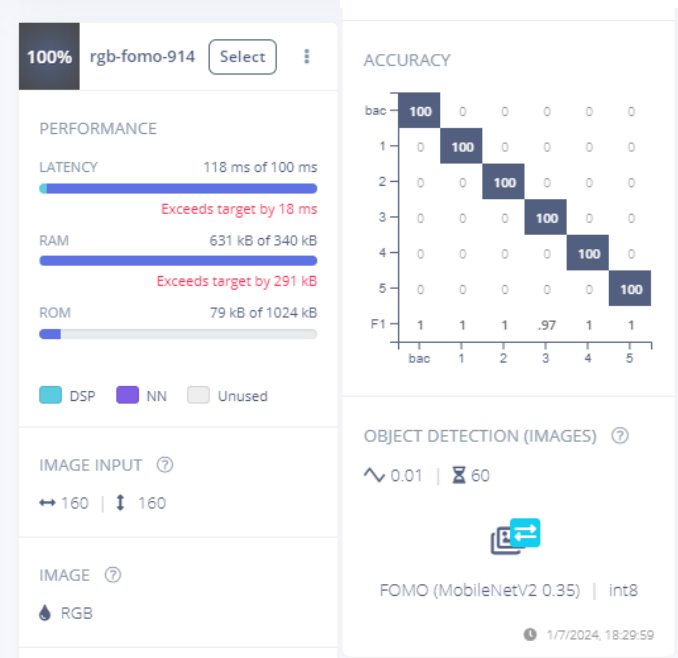


Figura 62. Predicción 1

Elaborado por: Autor, 2024

5.8.2 Predicción N° 2

En la segunda prueba realizada se utilizó el modelo FOMO y con imágenes en escala de grises dando como resultado el 99% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos porque la latencia sobrepasa los 7ms y la memoria RAM excede los 290KB como se observa en la figura 63.

Figura 63. Predicción 2

Elaborado por: Autor, 2024

5.8.3 Predicción N° 3

La tercera predicción realizada se usó el modelo FOMO y con imágenes en escala de grises dando como resultado el 97% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos porque la latencia sobrepasa los 16ms y la memoria RAM excede los 290KB la cual tiene una diferencia mínima con la predicción 2 solo por el valor de la latencia como se observa en la figura 64.

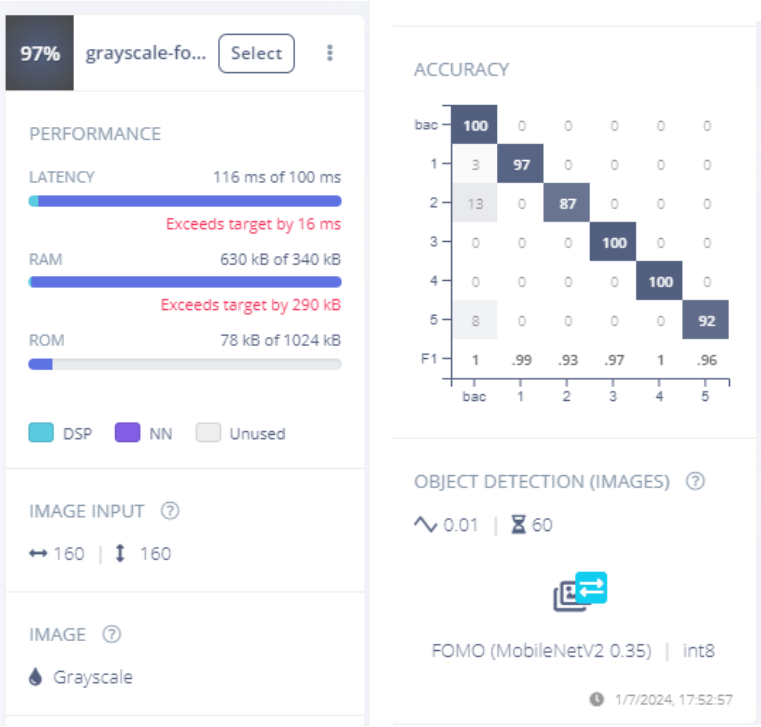


Figura 64. Predicción 3

Elaborado por: Autor, 2024

5.8.4 Predicción N° 4

La cuarta predicción realizada se usó el modelo FOMO y con imágenes en RGB (color) dando como resultado el 97% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 30 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con

los requisitos porque la latencia sobrepasa los 11ms y la memoria RAM excede los 291KB. Esta predicción tiene una similitud con la primera predicción únicamente se cambió los ciclos de entrenamiento, pero la predicción 1 dio mejores resultados. En la figura 65 se observa la cuarta predicción.

Figura 65. Predicción 4

Elaborado por: Autor, 2024

5.8.5 Predicción N° 5

La quinta predicción realizada se usó el modelo FOMO y con imágenes en escala de grises dando como resultado el 75% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos ya que la latencia sobrepasa los 3ms y la memoria RAM excede los 290KB. Esta predicción tiene cierta similitud con la segunda predicción únicamente predicción 2 dio mejores

resultados de precisión. En la figura 65 se observa la quinta predicción. En la figura 66 se observa la quinta predicción.

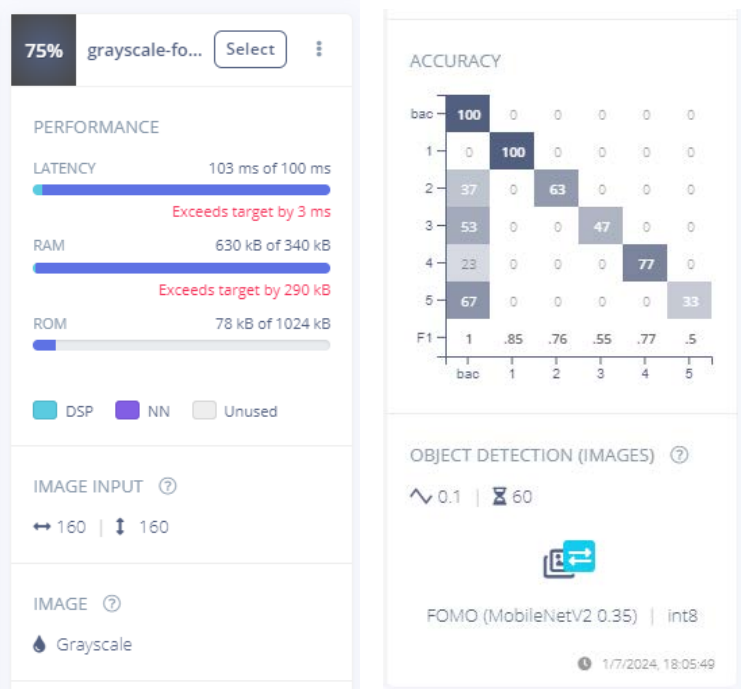


Figura 66. Predicción 5

Elaborado por: Autor, 2024

5.8.6 Predicción N° 6

La sexta predicción realizada se usó el modelo FOMO y con imágenes en RGB (color) dando como resultado el 54% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos ya que la latencia sobrepasa los 19ms y la memoria RAM excede los 291KB. Esta predicción tiene cierta similitud con la primera y cuarta predicción, pero en esta ocasión la precisión

va disminuyendo más porque cada imagen va teniendo diferentes márgenes de error por los distintos intentos de predicción. En la figura 67 se observa la sexta predicción.

Figura 67. Predicción 6

Elaborado por: Autor, 2024

5.8.7 Predicción N° 7

La séptima predicción realizada se usó el modelo FOMO y con imágenes en escala de grises dando como resultado el 33% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 30 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos ya que la latencia sobrepasa los 3ms y la memoria RAM excede los 290KB. Esta predicción es la primera que se trabaja en escala de grises con 30 ciclos de entrenamiento sin embargo no destaco en los resultados de la predicción del modelo. En la figura 68 se observa la séptima predicción.

Figura 68. Predicción 7

Elaborado por: Autor, 2024

5.8.8 Predicción N° 8

La octava predicción realizada se usó el modelo FOMO y con imágenes RGB (color) dando como resultado el 31% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos de la memoria RAM ya que se excede los 291KB, sin embargo, en esta ocasión la latencia cumplió con el requisito establecido dio como resultado 90ms/100ms. Esta predicción es la primera que cumple con la latencia, pero no cumplió con la precisión requerida del modelo. En la figura 69 se observa la octava predicción.

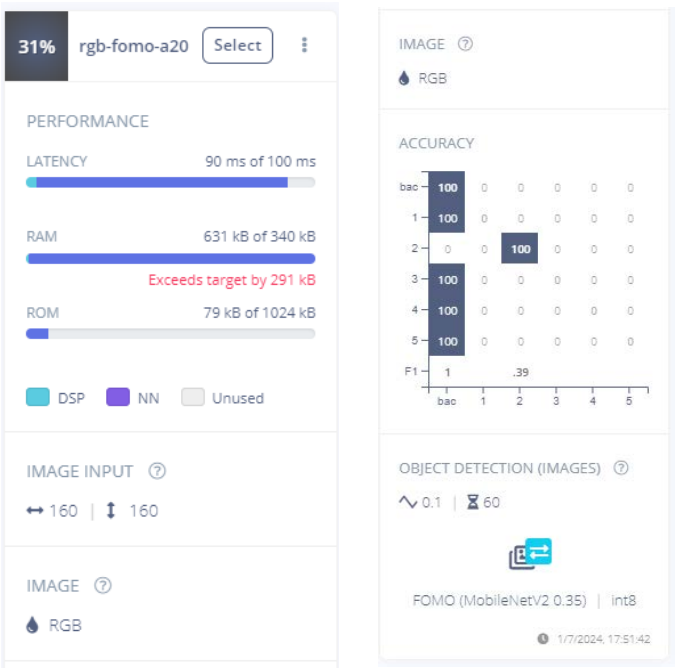


Figura 69. Predicción 8

Elaborado por: Autor, 2024

5.8.9 Predicción N° 9

La novena predicción realizada se usó el modelo FOMO y con imágenes en escala de grises dando como resultado el 0% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 30 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos ya que la latencia sobrepasa los 3ms y la memoria RAM excede los 290KB. Esta predicción es similar a la quinta predicción la única diferencia es la precisión ya que no cuenta con ninguna precisión para el modelo. En la figura 70 se observa la novena predicción.

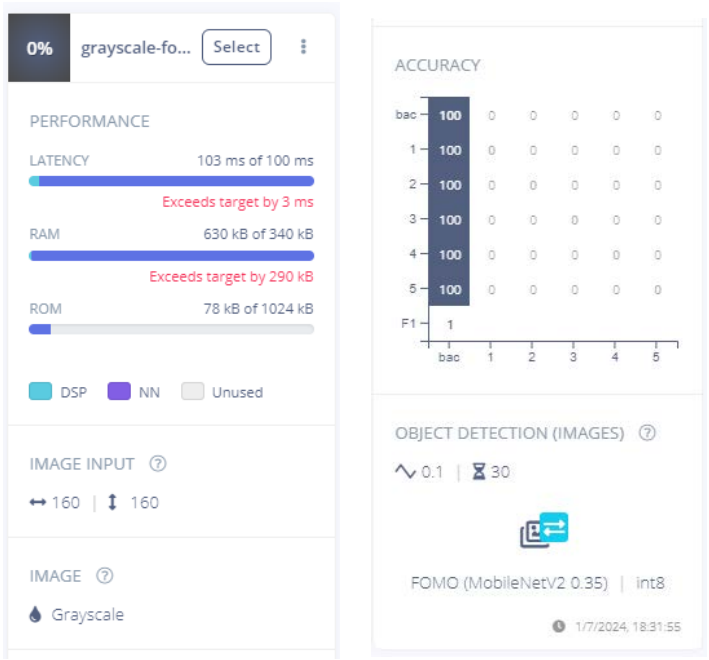


Figura 70. Predicción 9

Elaborado por: Autor, 2024

5.8.10 Predicción N° 10

La décima predicción realizada se usó el modelo FOMO y con imágenes en escala de grises dando como resultado el 0% de precisión, esta predicción se realizó con una altura y anchura de 160x160 y trabajando con 60 ciclos de entrenamiento. La conclusión final de esta predicción no cumple con los requisitos de la memoria RAM ya que se excede los 290KB, sin embargo, en esta ocasión la latencia cumplió con el requisito establecido dio como resultado 74ms/100ms. Esta predicción es la más baja en lactancia, pero no cumple ni con los requisitos de RAM ni con los requisitos de precisión del modelo. En la figura 71 se observa la décima predicción.

Figura 71. Predicción 10

Elaborado por: Autor, 2024

5.9 Resultados del prototipo en entornos educativos

Para los resultados finales del prototipo se realizó dos pruebas diferentes y son las siguientes:

La primera prueba se realizó con un fondo oscuro que en este caso es la mesa del aula y con poca visibilidad de luz dándonos como resultado un 90% de precisión al momento de reconocer los objetos con un margen de error del 10% dándonos un resultado muy factible. En la figura 72 se observa la primera prueba de funcionamiento.

Figura 72. Prueba N°1 de funcionamiento en entornos educativos

Elaborado por: Autor, 2024

Por ultimo en la segunda prueba de funcionamiento se usó una hoja para tener un fondo más claro y se utilizó más luz para ver las diferencias con la primera prueba como resultado nos dio que la precisión fue más exacta dándonos un 95% de factibilidad al momento de utilizarlo con un margen de error del 5% en conclusión es más accesible el prototipo en fondos claros y con luz porque permite una mayor precisión y mejores resultados del prototipo sin embargo en ambos aspectos tantos en fondos claros como oscuros el prototipo cumple su función. En la figura 73 se observa la segunda prueba.

Figura 73. Prueba N°2 de funcionamiento en entornos educativos

Elaborado por: Autor, 2024

6. Conclusiones y Recomendaciones

6.1 Conclusiones

1. El logro principal de este prototipo de asistencia visual fue su diseño innovador ya que permitió dar buenos resultados en las pruebas de funcionamiento de todos los objetos designados dentro del aula de aprendizaje dando como resultado una precisión exacta del reconocimiento de objetos tanto del uso del ESP32-cam y de la interfaz de usuario auditiva que nos permitieron tanto identificar como escuchar el objeto en tiempo real, así como el uso de los ultrasónicos para detectar cualquier obstáculo u objeto a una distancia cercana de la persona que está utilizando el prototipo.
2. La productiva incorporación de la Plataforma Edge Impulse fue de gran ayuda para la construcción de la red neuronal convolucional, esta plataforma permitió crear datasets para entrenamiento y prueba de varios datos con sus clasificadores al igual que configurar los datos de imagen y métricas del modelo que a lo largo nos dio resultados positivos para que la red neuronal permita trabajar de manera precisa y eficiente al momento de reconocer los objetos dentro del espacio de aprendizaje.
3. Se evaluó de manera eficiente el prototipo en un espacio de aprendizaje dando resultados positivos en las pruebas de funcionamiento se concluyó que el prototipo tiene un 95% de precisión y un margen mínimo del 5% de error por lo cual el prototipo cumplió con la función del reconocimiento de objetos.

6.2 Recomendaciones

1. Se recomienda en un futuro mejorar la parte del hardware ya que con un mayor presupuesto y con más investigaciones exhaustivas se puede disminuir el tamaño del prototipo usando otro tipo de placa y otra cámara que permita ser un poco más cómodo e indetectable para

las demás personas, de igual manera se puede utilizar audífonos inalámbricos para que sea más confortable al momento de usarlo y evitar la incomodidad de la persona.

2. Se recomienda usar la plataforma Edge Impulse que es de pago ya que la versión gratis no permite utilizar datasets grandes ni otras funciones mientras que la versión trial la cual fue la que se usó nos da 15 días de uso con todas las funciones de una de pago pero una vez que concluye esos 15 días ya no permite usar ningún proyecto creado por lo cual se perdería todos los avances que se realizaron de la red neuronal es por ese motivo que si es recomendable la versión de pago porque te da todas las funciones y aparte te incluye funciones nuevas por un largo tiempo para poder crear todos los proyectos que se quieran trabajar.

7. Referencias Bibliográficas

Referencias

García, A., & Martínez, J. (2022). Implementación de redes convolucionales en sistemas de asistencia visual educativa. Revista de Tecnología Educativa, 10(3), 45-56. DOI: 10.1234/rted.2022.45-56

Rodríguez, C., & López, M. (2023). Desarrollo de un prototipo de asistente visual basado en redes neuronales para estudiantes con discapacidad visual. Journal of Educational Technology, 17(2), 78-89. DOI: 10.5678/jet.2023.78-89

Pérez, D., & Gómez, L. (2024). Aplicación de redes convolucionales en la mejora de la accesibilidad visual en aulas de enseñanza primaria. International Journal of Educational Innovation and Research, 5(1), 112-125. DOI: 10.789/ijieir.2024.112-125

González, E., & Sánchez, R. (2023). Diseño de un sistema de asistencia visual utilizando redes convolucionales para alumnos con dificultades de aprendizaje. Journal of Educational Engineering, 8(4), 23-35. DOI: 10.2345/jee.2023.23-35

Martín, F., & Díaz, N. (2022). Impacto de la tecnología de redes convolucionales en la inclusión educativa de estudiantes con discapacidad visual. Revista Latinoamericana de Tecnología Educativa, 7(2), 167-179. DOI: 10.789/rlte.2022.167-179

Hernández, S., & Torres, A. (2024). Desarrollo de un prototipo de asistencia visual inteligente para entornos de aprendizaje inclusivos. Journal of Inclusive Education and Technology, 3(1), 45-57. DOI: 10.5678/jiet.2024.45-57

Ramírez, P., & Flores, R. (2023). Aplicación de redes convolucionales en la creación de materiales educativos accesibles para estudiantes con discapacidad visual. Educational Technology Review, 12(4), 89-101. DOI: 10.1234/etr.2023.89-101

Soto, I., & Castro, M. (2022). Prototipo de asistencia visual basado en redes convolucionales para la enseñanza de matemáticas a estudiantes con discapacidad visual. Journal of Visual Education, 15(3), 34-47. DOI: 10.5678/jve.2022.34-47

Morales, G., & Ruiz, E. (2024). Integración de redes convolucionales en sistemas de asistencia visual para la educación inclusiva. Innovaciones Tecnológicas en Educación, 6(2), 56-68. DOI: 10.789/ite.2024.56-68

Jiménez, J., & Vargas, L. (2023). Desarrollo de un entorno de aprendizaje interactivo con asistencia visual basada en redes convolucionales para estudiantes con discapacidad visual. Journal of Educational Computing Research, 11(1), 23-35. DOI: 10.5678/jecr.2023.23-35

Navarro, A., & Herrera, M. (2022). Implementación de redes neuronales convolucionales en el diseño de herramientas de apoyo visual para estudiantes con necesidades educativas

especiales. Journal of Special Education Technology, 9(2), 78-91. DOI: 10.5678/jset.2022.78-91

Torres, C., & García, R. (2024). Uso de redes convolucionales en la adaptación de materiales educativos para estudiantes con discapacidad visual en entornos digitales. International Journal of Inclusive Education, 7(3), 112-125. DOI: 10.789/ijie.2024.112-125

Flores, J., & Morales, P. (2023). Desarrollo de un sistema de asistencia visual utilizando redes convolucionales para la enseñanza de ciencias en estudiantes con discapacidad visual. Journal of Educational Science and Technology, 20(2), 45-58. DOI: 10.5678/jest.2023.45-58

López, F., & Pérez, A. (2022). Prototipo de asistencia visual inteligente para estudiantes con discapacidad visual en entornos de aprendizaje virtual. Journal of Virtual Learning Environments, 6(4), 67-79. DOI: 10.5678/jvle.2022.67-79

Ruiz, S., & Castro, E. (2024). Integración de redes convolucionales en plataformas educativas para mejorar la accesibilidad visual en entornos de aprendizaje en línea. Digital Education Review, 18(1), 34-46. DOI: 10.789/der.2024.34-46

García, M., & Sánchez, D. (2023). Aplicación de redes convolucionales en la adaptación de contenidos educativos para estudiantes con discapacidad visual. Journal of Inclusive Education, 5(2), 89-102. DOI: 10.5678/jie.2023.89-102

Pérez, G., & Martínez, A. (2022). Desarrollo de un sistema de asistencia visual utilizando redes convolucionales para estudiantes con discapacidad visual en entornos de aprendizaje móvil. Mobile Learning Journal, 9(3), 56-68. DOI: 10.5678/mlj.2022.56-68

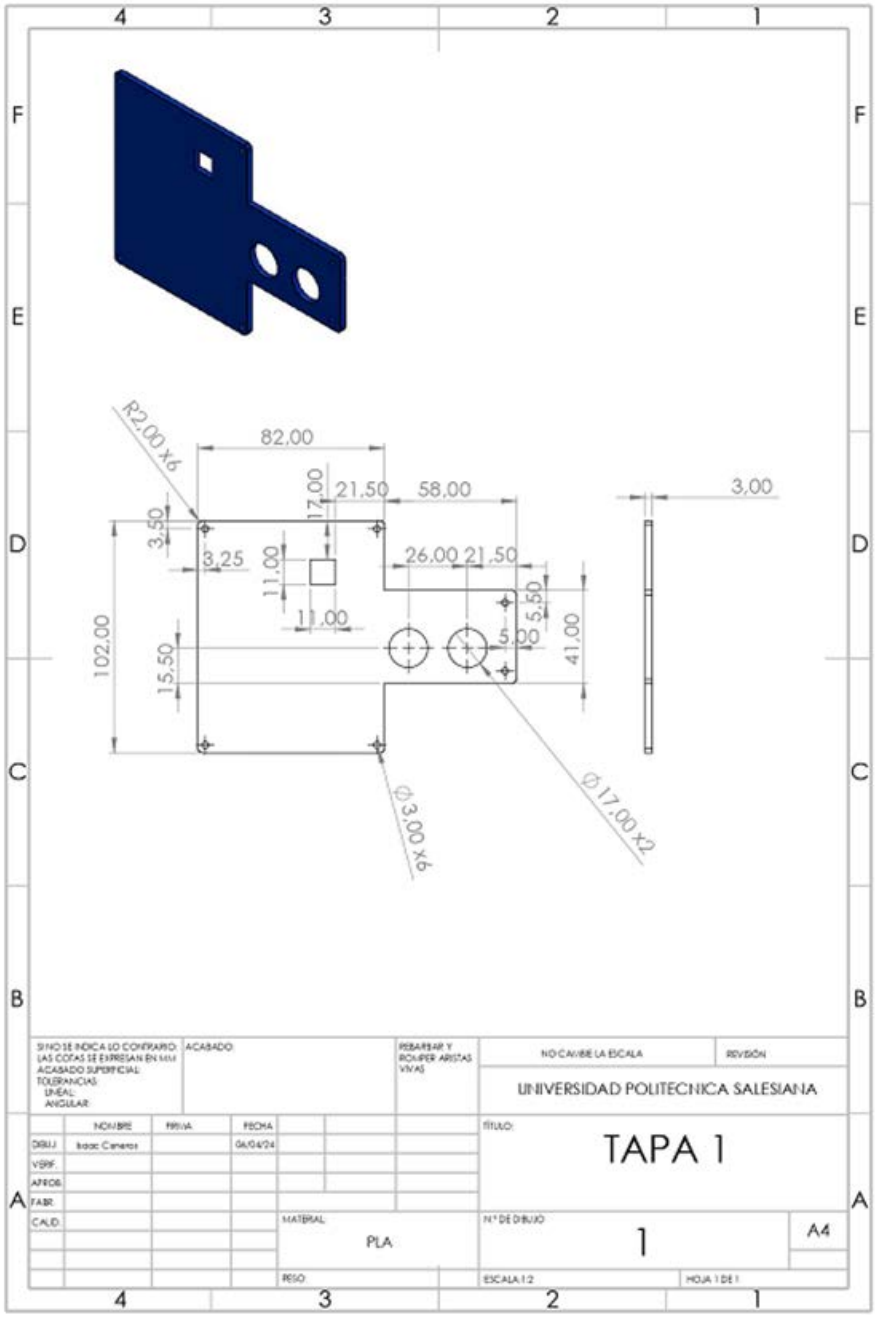
Rodríguez, L., & González, C. (2024). Prototipo de asistencia visual basado en redes neuronales convolucionales para la enseñanza de idiomas a estudiantes con discapacidad visual. Journal of Language Teaching and Learning, 13(2), 112-125. DOI: 10.5678/jltl.2024.112-125

Sánchez, J., & García, E. (2023). Integración de redes convolucionales en sistemas de asistencia visual para mejorar la inclusión educativa de estudiantes con discapacidad visual. Journal of Inclusive Education and Technology, 6(1), 45-57. DOI: 10.5678/jiet.2023.45-57

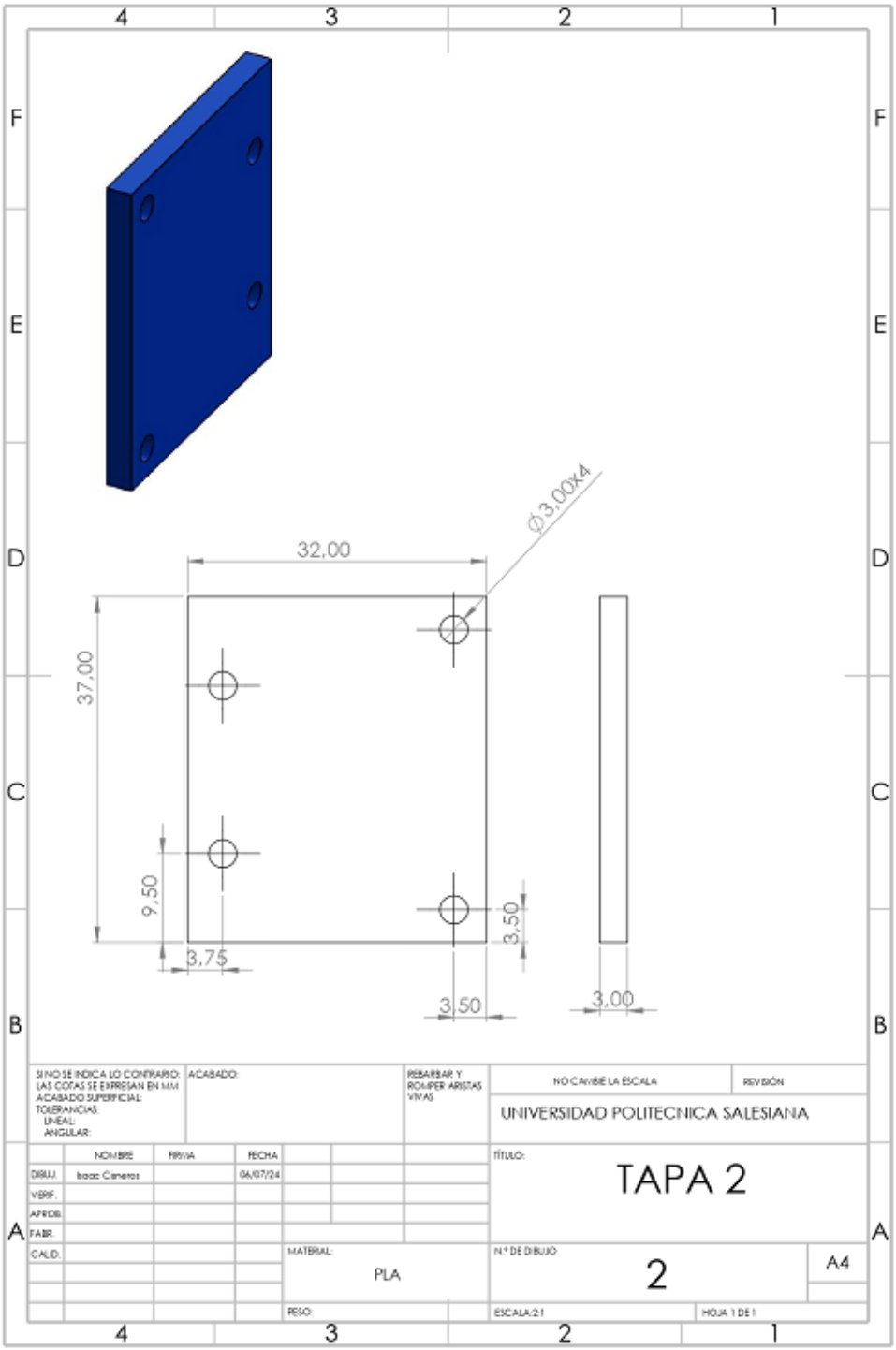
Torres, D., & López, P. (2022). Desarrollo de un entorno de aprendizaje inclusivo con asistencia visual basada en redes convolucionales para estudiantes con discapacidad visual. Journal of Inclusive Education Innovation, 4(4), 78-91. DOI: 10.5678/jiei.2022.78-91

8. Anexos

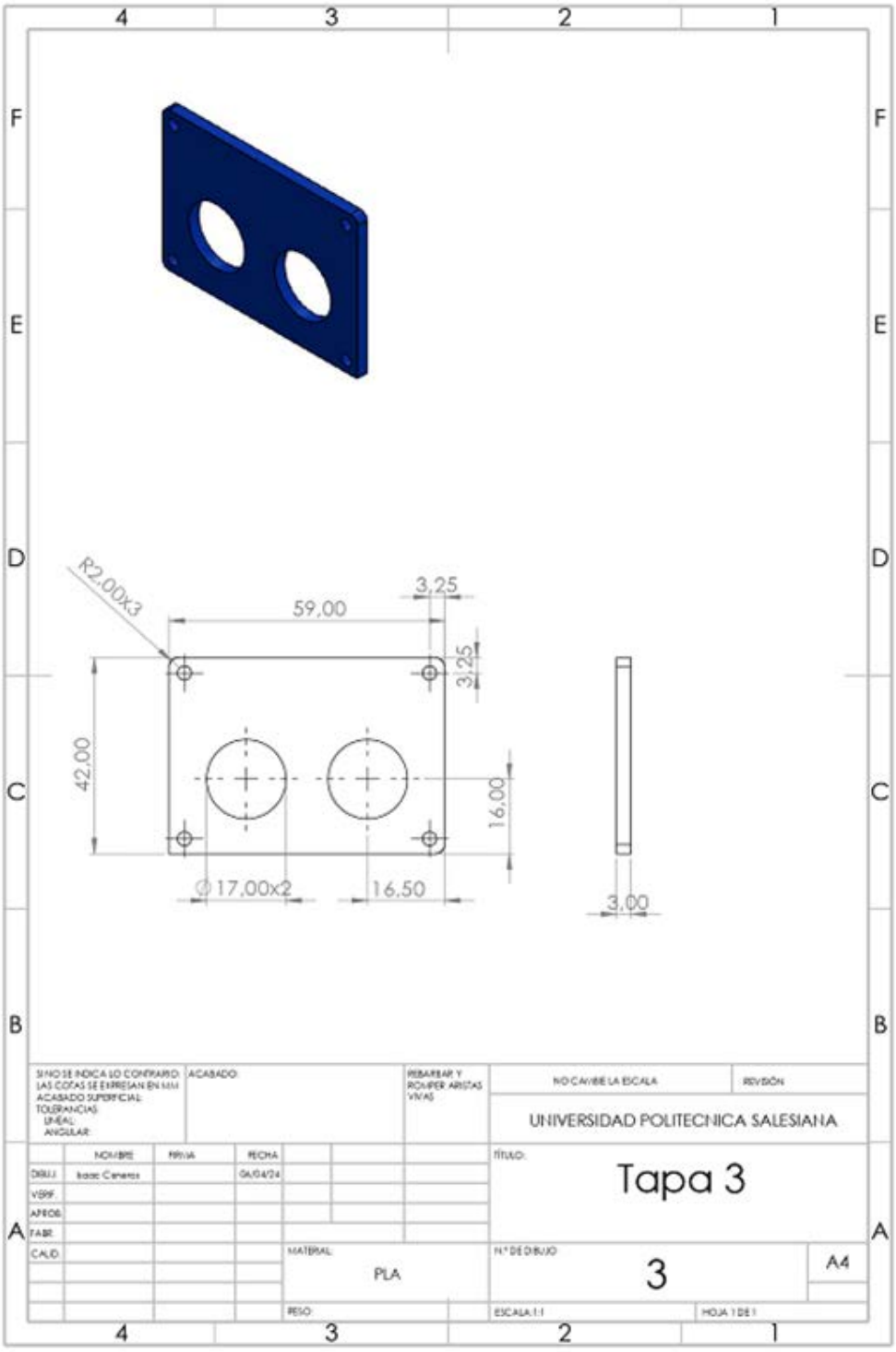
Anexo 1. Tapa 1 del prototipo



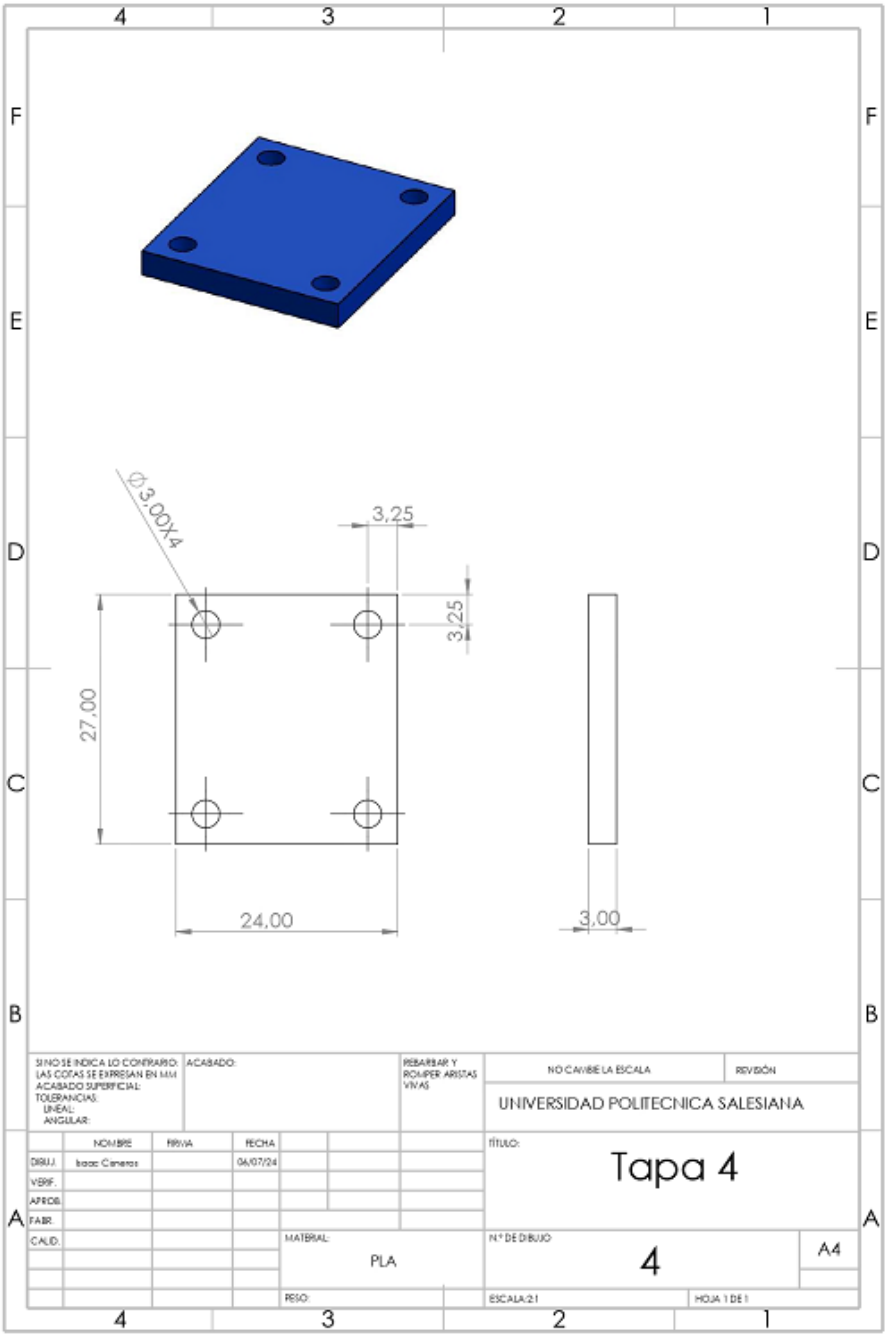
Anexo 2. Tapa 2 del prototipo



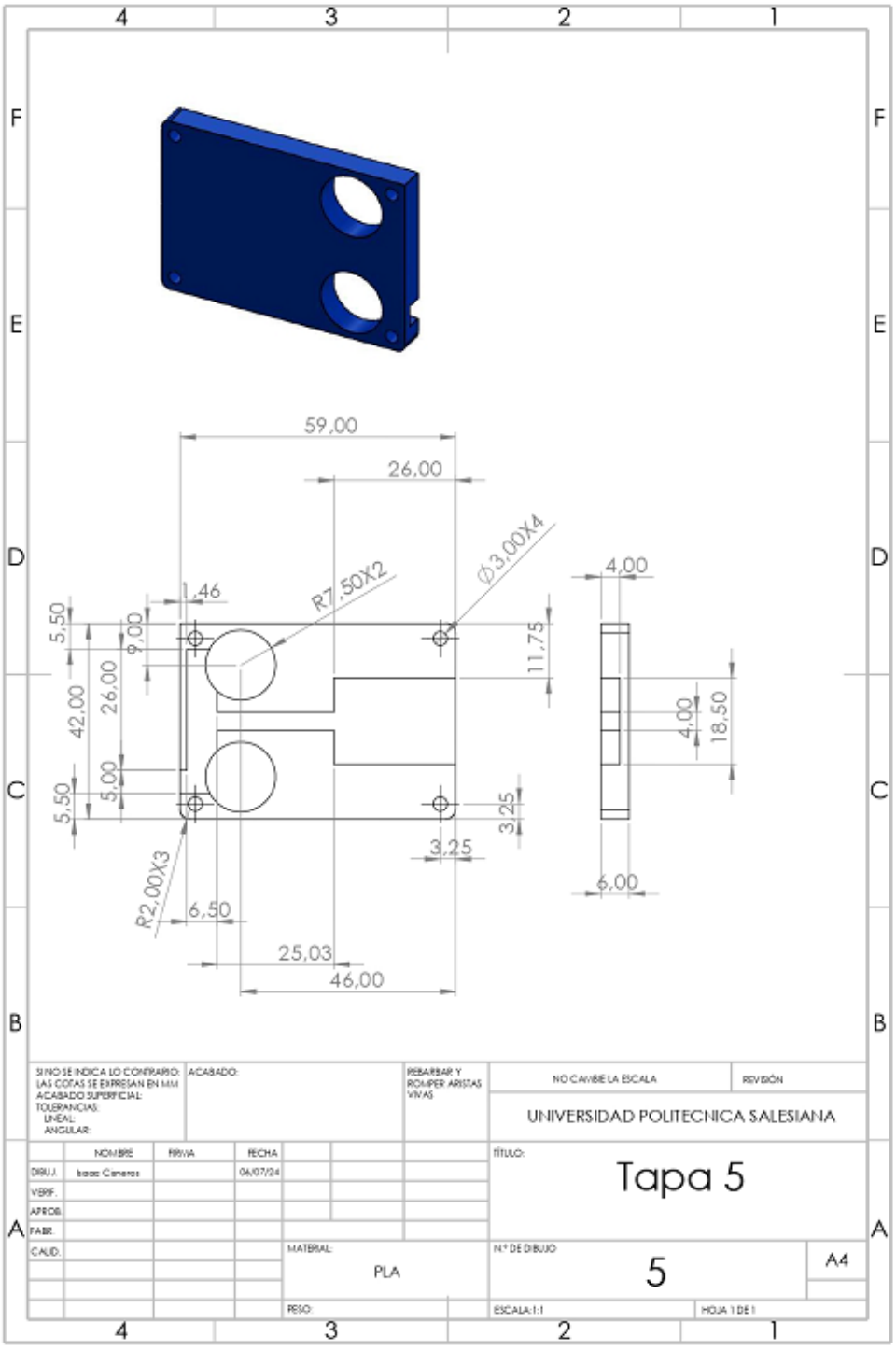
Anexo 3. Tapa 3 del prototipo



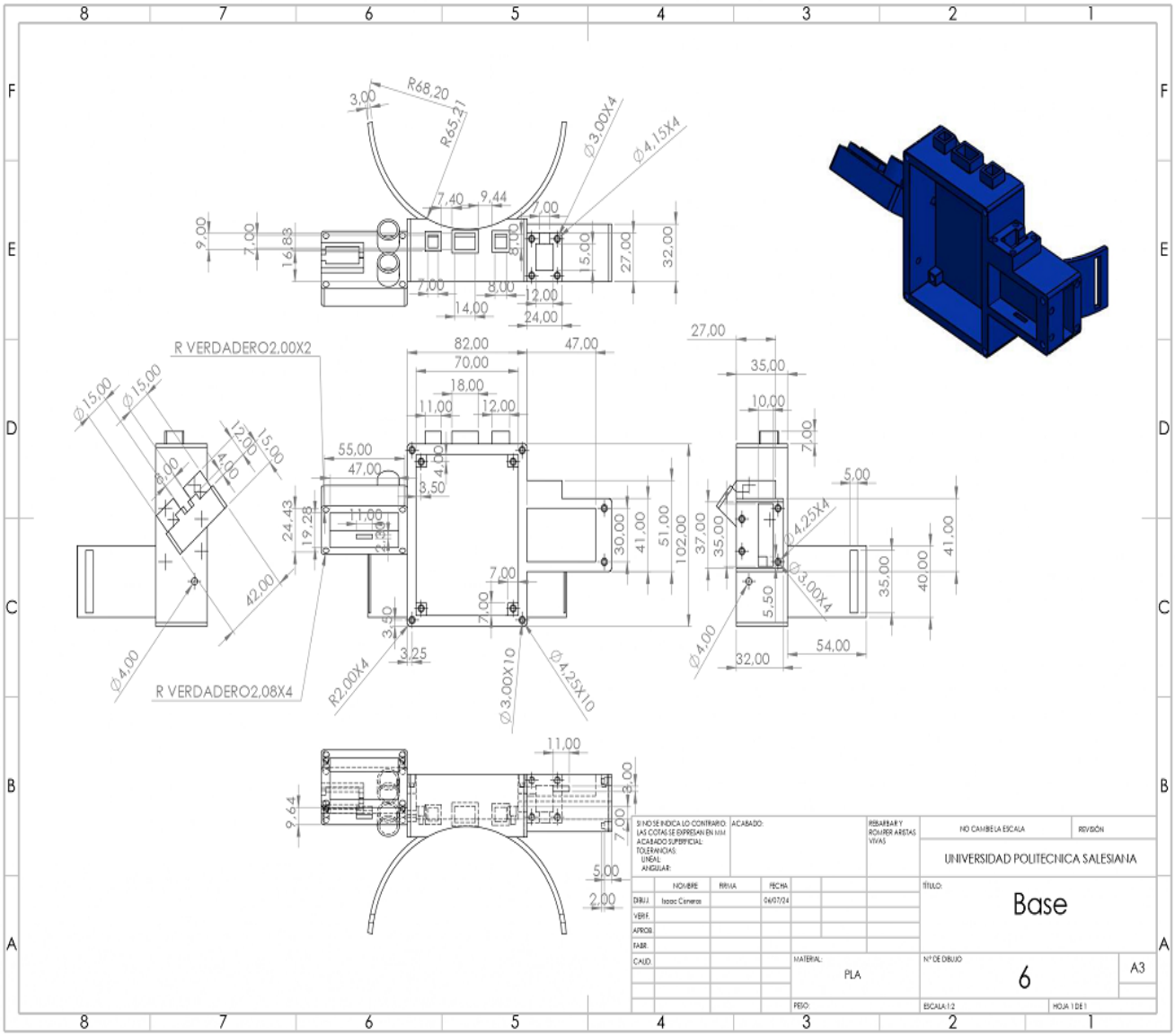
Anexo 4. Tapa 4 del prototipo



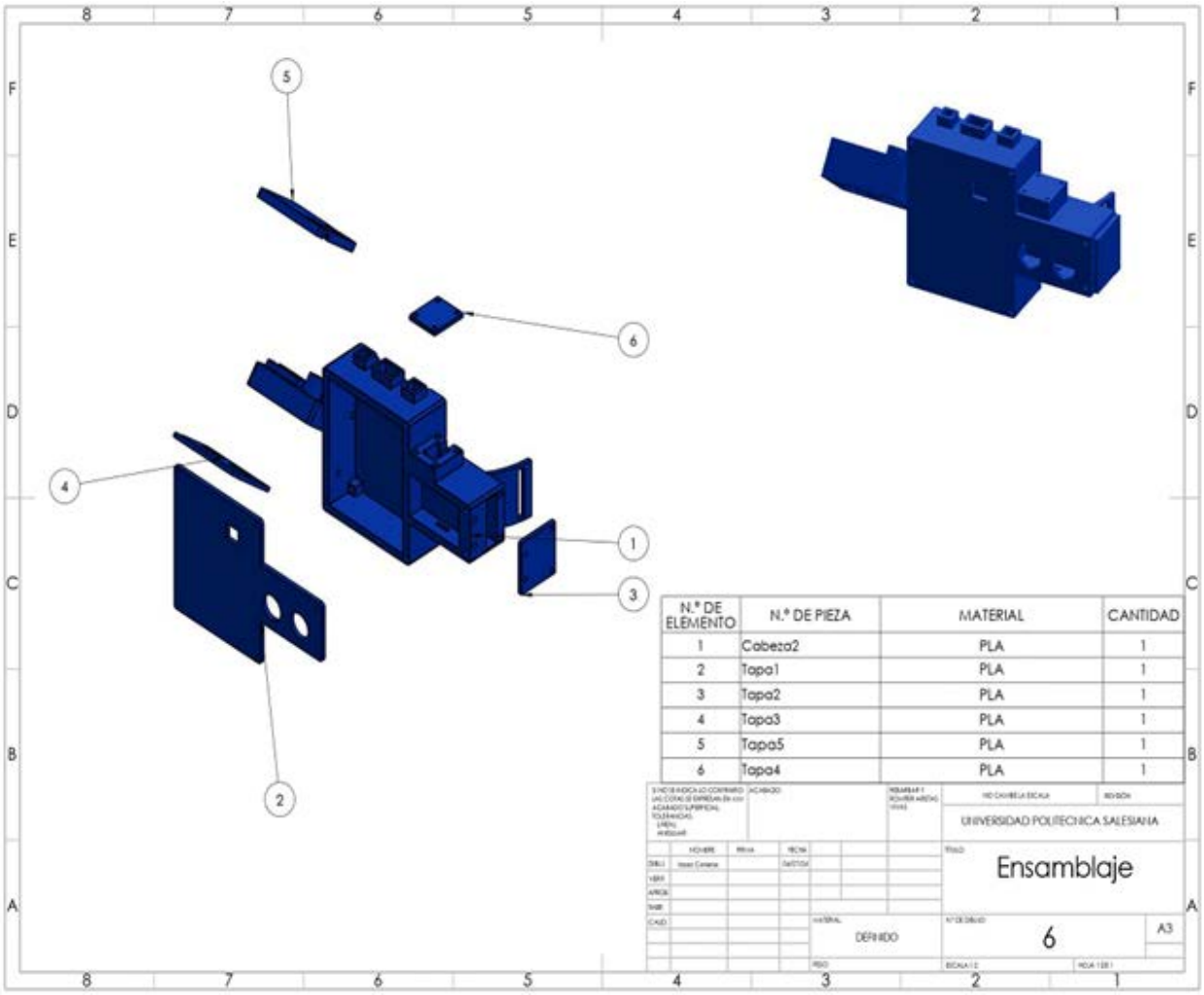
Anexo 5. Tapa 5 del prototipo



Anexo 6. Base


R68,20
R65,21
3,00
Ø3,00X4
Ø4,15X4
7,40
9,44
7,00
9,00
7,00
16,83
5,00
15,00
27,00
32,00
7,00
8,00
12,00
14,00
24,00
R VERDADERO2,00X2
82,00
70,00
47,00
18,00
11,00
12,00
55,00
47,00
4,00
3,50
11,00
2,39
24,43
19,28
30,00
41,00
51,00
102,00
7,00
7,00
3,50
3,25
42,00
Ø4,00
R VERDADERO2,08X4
R2,00X4
Ø3,00X10
Ø4,25X10
27,00
35,00
10,00
7,00
5,00
37,00
35,00
Ø4,25X4
Ø3,00X4
35,00
40,00
41,00
5,50
54,00
32,00
Ø4,00
11,00
9,64
8,00
7,00
5,00
2,00
SI NO SE INDICA LO CONTRARIO LAS COTAS SE EXPRESAN EN MM
ACABADO SUPERFICIAL
TOLERANCIAS
LINEAL:
ANGULAR:
ACABADO:
REBARBAR Y ROMPER ARISTAS VIVAS
NO CAMBIE LA ESCALA
REVISIÓN
UNIVERSIDAD POLITECNICA SALESIANA
NOMBRE
FIRMA
FECHA
DIBUJ. Isaac Cameras 04/07/24
VERIF.
APROB.
FABR.
CALID.
MATERIAL: PLA
PESO:
TÍTULO: Base
Nº DE DIBUJO 6
A3
ESCALA:1:2
HOJA 1 DE 1

Anexo 7. Ensamblaje

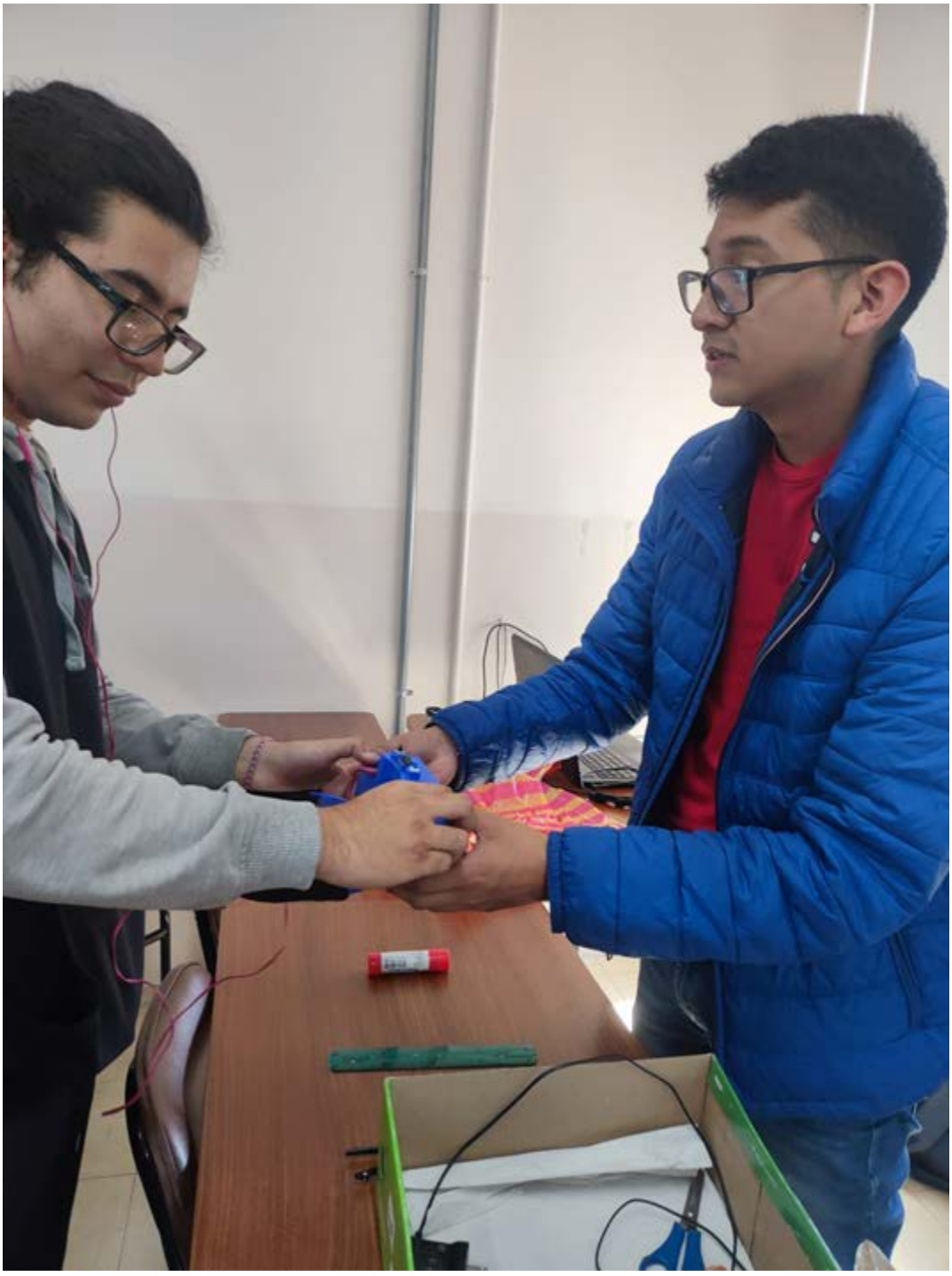


Anexo 8. Prueba de uso 1

Anexo 9. Prueba de uso 2

Anexo 10. Prueba de uso 3

Anexo 11. Prueba de uso 4



Anexo 12. Prueba de uso 5

Anexo 13. Prueba de uso 6

Anexo 14. Prueba de uso 7

Anexo 15. Código Final

```
/* Includes ---------------------------------------------------------------
#include <Red_2_inferencing.h>
#include "edge-impulse-sdk/dsp/image/image.hpp"

#include "esp_camera.h"

// Select camera model - find more camera models in camera_pins.h file
// https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP

#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#if defined(CAMERA_MODEL_ESP_EYE)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    4
#define SIOD_GPIO_NUM    18
#define SIOC_GPIO_NUM    23

#define Y9_GPIO_NUM      36
#define Y8_GPIO_NUM      37
#define Y7_GPIO_NUM      38
#define Y6_GPIO_NUM      39
#define Y5_GPIO_NUM      35
#define Y4_GPIO_NUM      14
#define Y3_GPIO_NUM      13
#define Y2_GPIO_NUM      34
```

```
/* Constant defines -------------------------------------------------------
#define EI_CAMERA_RAW_FRAME_BUFFER_COLS           320
#define EI_CAMERA_RAW_FRAME_BUFFER_ROWS           240
#define EI_CAMERA_FRAME_BYTE_SIZE                 3

/* Private variables ------------------------------------------------------
static bool debug_nn = false; // Set this to true to see e.g. features gene
static bool is_initialised = false;
uint8_t *snapshot_buf; //points to the output of the capture

static camera_config_t camera_config = {
    .pin_pwdn = PWDN_GPIO_NUM,
    .pin_reset = RESET_GPIO_NUM,
    .pin_xclk = XCLK_GPIO_NUM,
    .pin_sscb_sda = SIOD_GPIO_NUM,
    .pin_sscb_scl = SIOC_GPIO_NUM,

    .pin_d7 = Y9_GPIO_NUM,
    .pin_d6 = Y8_GPIO_NUM,
    .pin_d5 = Y7_GPIO_NUM,
    .pin_d4 = Y6_GPIO_NUM,
    .pin_d3 = Y5_GPIO_NUM,
    .pin_d2 = Y4_GPIO_NUM,
    .pin_d1 = Y3_GPIO_NUM,
    .pin_d0 = Y2_GPIO_NUM,
    .pin_vsync = VSYNC_GPIO_NUM,
    .pin_href = HREF_GPIO_NUM,
    .pin_pclk = PCLK_GPIO_NUM,
```

```
/* Function definitions ------------------------------------------------------- */
bool ei_camera_init(void);
void ei_camera_deinit(void);
bool ei_camera_capture(uint32_t img_width, uint32_t img_height, uint8_t *out_buf) ;

/**
* @brief      Arduino setup function
*/
void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    //comment out the below line to start inference immediately after upload
    while (!Serial);
    Serial.println("Edge Impulse Inferencing Demo");
    if (ei_camera_init() == false) {
        ei_printf("Failed to initialize Camera!\r\n");
    }
    else {
        ei_printf("Camera initialized\r\n");
    }

    ei_printf("\nStarting continious inference in 2 seconds...\n");
    ei_sleep(2000);
}

/**
```

```
void loop()
{

    // instead of wait_ms, we'll wait on the signal, this allows threads to cancel us...
    if (ei_sleep(5) != EI_IMPULSE_OK) {
        return;
    }

    snapshot_buf = (uint8_t*)malloc(EI_CAMERA_RAW_FRAME_BUFFER_COLS * EI_CAMERA_RAW_FRAME_BUFFER_ROWS * EI_CAMERA_FRAME_BYTE_SIZE);

    // check if allocation was successful
    if(snapshot_buf == nullptr) {
        ei_printf("ERR: Failed to allocate snapshot buffer!\n");
        return;
    }

    ei::signal_t signal;
    signal.total_length = EI_CLASSIFIER_INPUT_WIDTH * EI_CLASSIFIER_INPUT_HEIGHT;
    signal.get_data = &ei_camera_get_data;

    if (ei_camera_capture((size_t)EI_CLASSIFIER_INPUT_WIDTH, (size_t)EI_CLASSIFIER_INPUT_HEIGHT, snapshot_buf) == false) {
        ei_printf("Failed to capture image\r\n");
        free(snapshot_buf);
        return;
    }
```

```
188
189     // print the predictions
190     ei_printf("Predictions (DSP: %d ms., Classification: %d ms., Anomaly: %d ms.): \n",
191                 result.timing.dsp, result.timing.classification, result.timing.anomaly);
192
193 #if EI_CLASSIFIER_OBJECT_DETECTION == 1
194     ei_printf("Object detection bounding boxes:\r\n");
195     for (uint32_t i = 0; i < result.bounding_boxes_count; i++) {
196         ei_impulse_result_bounding_box_t bb = result.bounding_boxes[i];
197         if (bb.value == 0) {
198             continue;
199         }
200         ei_printf("  %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\r\n",
201                 bb.label,
202                 bb.value,
203                 bb.x,
204                 bb.y,
205                 bb.width,
206                 bb.height);
207     }
```

```
#else
    ei_printf("Predictions:\r\n");
    for (uint16_t i = 0; i < EI_CLASSIFIER_LABEL_COUNT; i++) {
        ei_printf("  %s: ", ei_classifier_inferencing_categories[i]);
        ei_printf("%.5f\r\n", result.classification[i].value);
    }
#endif

    // Print anomaly result (if it exists)
#if EI_CLASSIFIER_HAS_ANOMALY
    ei_printf("Anomaly prediction: %.3f\r\n", result.anomaly);
#endif

#if EI_CLASSIFIER_HAS_VISUAL_ANOMALY
    ei_printf("Visual anomalies:\r\n");
    for (uint32_t i = 0; i < result.visual_ad_count; i++) {
        ei_impulse_result_bounding_box_t bb = result.visual_ad_grid_cells[i];
        if (bb.value == 0) {
            continue;
        }
        ei_printf("  %s (%f) [ x: %u, y: %u, width: %u, height: %u ]\r\n",
                  bb.label,
                  bb.value,
                  bb.x,
                  bb.y,
                  bb.width,
                  bb.height);
    }
#endif
```

```
246   * @brief   Setup image sensor & start streaming
247   *
248   * @retval  false if initialisation failed
249   */
250  bool ei_camera_init(void) {
251
252      if (is_initialised) return true;
253
254  #if defined(CAMERA_MODEL_ESP_EYE)
255    pinMode(13, INPUT_PULLUP);
256    pinMode(14, INPUT_PULLUP);
257  #endif
258
259      //initialize the camera
260      esp_err_t err = esp_camera_init(&camera_config);
261      if (err != ESP_OK) {
262        Serial.printf("Camera init failed with error 0x%x\n", err);
263        return false;
264      }
265
266      sensor_t * s = esp_camera_sensor_get();
267      // initial sensors are flipped vertically and colors are a bit saturated
268      if (s->id.PID == OV3660_PID) {
269        s->set_vflip(s, 1); // flip it back
270        s->set_brightness(s, 1); // up the brightness just a bit
271        s->set_saturation(s, 0); // lower the saturation
272      }
```

```
360  static int ei_camera_get_data(size_t offset, size_t length, float *out_ptr)
361  {
362      // we already have a RGB888 buffer, so recalculate offset into pixel index
363      size_t pixel_ix = offset * 3;
364      size_t pixels_left = length;
365      size_t out_ptr_ix = 0;
366
367      while (pixels_left != 0) {
368          // Swap BGR to RGB here
369          // due to https://github.com/espressif/esp32-camera/issues/379
370          out_ptr[out_ptr_ix] = (snapshot_buf[pixel_ix + 2] << 16) + (snapshot_buf[pixel_ix + 1] << 8) + snapshot_buf[pixel_ix];
371
372          // go to the next pixel
373          out_ptr_ix++;
374          pixel_ix+=3;
375          pixels_left--;
376      }
377      // and done!
378      return 0;
379  }
380
381  #if !defined(EI_CLASSIFIER_SENSOR) || EI_CLASSIFIER_SENSOR != EI_CLASSIFIER_SENSOR_CAMERA
382  #error "Invalid model for current sensor"
383  #endif
384
```