

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



ESTIMACIÓN DE ALTURA DE COLUMNA DE GASES VOLCÁNICOS MEDIANTE
LA UTILIZACIÓN DE SISTEMAS EMBEBIDOS EN LAS EMANACIONES DEL
VOLCÁN CUYANAUSUL, AHUACHAPÁN

TRABAJO DE GRADO PARA OBTENER EL TÍTULO DE:
INGENIERO ELECTRICISTA

PRESENTADO POR:
JORGE REYES, NESTOR ALEXANDER
MARTINEZ REYES, GERARDO EDENILSON

CIUDAD UNIVERSITARIA, SAN SALVADOR ABRIL 2026

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. JUAN ROSA QUINTANILLA

SECRETARIO GENERAL:

LIC. PEDRO ROSALÍO ESCOBAR CASTANEDA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

MSC. LUIS SALVADOR BARRERA MANCÍA

SECRETARIO:

ARQ. RAÚL ALEXANDER FABIÁN ORELLANA

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR:

ING. WERNER DAVID MELÉNDEZ VALLE

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

TRABAJO DE GRADUACIÓN PREVIO A OPCIÓN DE GRADO DE:
INGENIERO ELECTRICISTA

TÍTULO:

ESTIMACIÓN DE ALTURA DE COLUMNA DE GASES VOLCÁNICOS MEDIANTE
LA UTILIZACIÓN DE SISTEMAS EMBEBIDOS EN LAS EMANACIONES DEL
VOLCÁN CUYANAUSUL, AHUACHAPÁN

PRESENTADO POR:

NÉSTOR ALEXANDER JORGE REYES
GERARDO EDENILSON MARTINEZ REYES

TRABAJO DE GRADUACIÓN APROBADO POR:

DOCENTE ASESOR:

PhD. CARLOS OSMÍN POCASANGRE JIMÉNEZ

SAN SALVADOR, ABRIL DE 2026

TRABAJO DE GRADUACIÓN APROBADO POR:

DOCENTE ASESOR:

PhD. CARLOS OSMÍN POCASANGRE JIMENEZ

NOTA Y DEFENSA FINAL

En esta fecha, jueves 11 de diciembre de 2025, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 10:00 a.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Werner David Meléndez Valle
Director


Firma

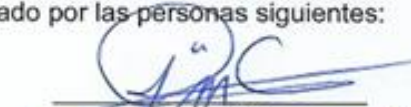


2. MSc. José Wilber Calderón Urrutia
Secretario



Firma

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

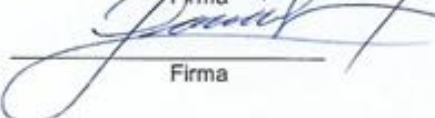
- DR. CARLOS OSMIN POCASANGRE JIMÉNEZ
(Docente Asesor)


Firma

- ING. WERNER DAVID MELÉNDEZ VALLE


Firma

- ING. LUIS ERNESTO ESCOBAR BRIZUELA


Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

ESTIMACIÓN DE ALTURA DE COLUMNA DE GASES VOLCÁNICOS MEDIANTE LA UTILIZACIÓN DE SISTEMAS EMBEBIDOS EN LAS EMANACIONES DEL VOLCÁN CUYANAUSUL, AHUACHAPÁN

A cargo de los Bachilleres:

- JORGE REYES NÉSTOR ALEXANDER
- MARTÍNEZ REYES GERARDO EDENILSON

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final:

8.0

(Ocho punto Cero)

Agradecimientos

Nestor Alexander Jorge Reyes

En el silencio de las noches de estudio, en los retos de cada día, cuando el cansancio y la duda parecían imponerse, encontré en Dios la paz que sostuvo mi ánimo, esa luz que iluminaba mis pensamientos, ese sentimiento que me acompañaba en cada logro es por esto y mucho más, que doy todas las gracias a él.

A mi madre Cristina que siempre estuvo ahí apoyándome, dándome ánimos, demostrando que uno es capaz de todo cuando se lo propone, inculcando valores que me guiaron en todos los años de estudio.

A mi padre Juan cuya trayectoria profesional ha sido una guía y un ejemplo constante de superación y por su apoyo en mis necesidades académicas que me permitieron llegar hasta aquí.

A mis hermanas Débora e Itzel, a mis tíos Carlos y Vilma, a mis abuelos Cristina, Angela, Rutilio y hasta el cielo a Salvador, que fueron pilares fundamentales, sus aportaciones me hicieron más grande como persona y profesional.

A mis amigos que estuvieron desde un inicio Vanessa, Neris, Madeline, Steven y Alejandra que se volvieron hermanos, que me demostraron que algo complicado puede volverse sencillo si se está con las personas correctas.

A Belen, Rosibel y Gicela quienes llegaron durante el camino, decidieron quedarse a acompañar hasta este momento y fueron ese último impulso que se necesitaba para seguir adelante.

Al Doctor Carlos Pocasangre, por su increíble conocimiento y profesionalismo que lo caracterizan que nos estuvo guiando durante el proceso, a Reina por su apoyo y bondad que nos a lo largo de mi trayectoria.

Gerardo Edenilson Martínez Reyes

A lo largo del camino académico se presentan momentos de dificultad en los que el cansancio, las dudas y los desafíos pueden hacer pensar en rendirse. Sin embargo, cada obstáculo también representa una oportunidad para aprender, crecer y continuar avanzando hacia las metas propuestas. En más de una ocasión el proceso pudo parecer complejo, pero la perseverancia, el apoyo de quienes estuvieron presentes y la fe permitieron seguir adelante hasta alcanzar este logro.

En primer lugar, agradezco a Dios por brindarme la fortaleza, la sabiduría y la perseverancia necesarias para superar cada reto presentado durante el desarrollo de este trabajo de graduación y a lo largo de mi formación académica. Su guía ha sido fundamental para no perder la esperanza en los momentos difíciles y para poder culminar esta etapa tan importante.

A mi padre Gerardo y mi madre Blanca, que aún en momentos difíciles nunca han dejado de darme todo su apoyo, por esas palabras de aliento, por cada una de las veces que han dicho que están orgullosos de mí y demostrarme que cada cosa que me propongo puedo conseguirla.

A mis hermanos, Fernando, Fabiola y Alma por que han sido fundamentales en mi vida y en este proceso y por todo el apoyo que me han brindado a lo largo del camino; A mis abuelos: Bertha y Carlos, Cristina y especialmente a mi abuelo Rodrigo que estaría apoyándome como siempre lo hizo, y a mi familia en general porque cada uno ha aportado significativamente su apoyo.

A Katya por creer en mí, por apoyarme incondicionalmente, por cada muestra de cariño y palabras de aliento en los momentos en que rendirse parecía la opción más fácil.

También expreso mis agradecimientos al Dr. Carlos Pocasangre por su valiosa orientación, paciencia y disposición para compartir sus conocimientos a lo largo del desarrollo de esta investigación, sus recomendaciones fueron fundamentales para fortalecer el contenido de este proyecto y para lograr su culminación de manera satisfactoria

De igual manera, agradezco a los docentes que fueron parte de mi proceso de formación académica, quienes con su dedicación, enseñanza y experiencia contribuyeron al desarrollo de mis conocimientos y habilidades profesionales.

También a Reina Vides, por su bondad y amabilidad con cada gestión, dando palabras de aliento y dando su apoyo cada vez que fuera necesario. Gracias

Índice de contenido

Índice de contenido	1
Índice de figuras	3
Índice de tablas.....	4
Glosario	5
Introducción	9
Objetivos	10
General.....	10
Específicos.....	10
Planteamiento del problema.....	11
Alcances	12
Justificación.....	13
Capítulo 1	14
1.0 Marco teórico	14
1.1 Antecedentes proyecto de ingeniería	14
1.2 Vulcanología básica.....	19
1.3 Importancia del monitoreo:.....	19
1.4 Vulcanología en El Salvador.....	20
1.5 Redes Neuronales Convolucionales	21
1.6 Sistemas embebidos	22
Capítulo 2	24
2.0 Metodología	24
2.1 Data set	24
2.2 Data Augmentation	27
2.3 Normalización de datos.....	29
2.4 Detección de la fumarola.....	30
2.5 Software	31
2.6 Hardware	32
Capítulo 3	36

3.0 Entrenamiento del modelo	36
3.1 Entrenamiento del algoritmo	37
3.2 Proceso de construcción de modelo.....	38
3.3 presentación del modelo final.....	39
3.4 Análisis del rendimiento del modelo	41
3.5 Exportación del modelo.....	45
Capítulo 4	46
4.0 Diseño de interfaz de usuario	46
4.1 Iniciación y permisos.	46
Capítulo 5	48
5.0 Implementación del sistema embebido.	48
5.1 Proceso ESP-32	48
Capítulo 6	51
6.0 Ensamblaje final	51
Capítulo 7	54
7.0 Resultados	54
7.1 Visita de campo.	54
7.2 Pruebas de campo.....	58
7.3 Limitaciones del sistema	60
7.4 Posibles mejoras y trabajos a futuro	61
7.5 Discusión de resultados.....	61
Capítulo 8	63
8.0 Conclusiones	63
Capítulo 9	66
9.0 Referencias bibliográficas	66

Índice de figuras

Figura 1 : Generación de un marcador aruco con ID=0, cuadros 5x5, de lado 50mm, captura de pantalla de Aruco maker. Fuente. ArUco markers generator(Chev.me, s,f).....	16
Figura 2: Se presenta la medida estimada de una puerta con el equipo, en rojo se muestra el marcador aruco, se obtuvo una altura de 1.92 metros. Fuente. Elaboración propia	17
Figura 3 Medidas reales de una puerta de 192 cm. Fuente. elaboración propia	18
Figura 4: Medida real de un lapicero -Vs- medida mediante el sistema. Fuente. Elaboración propia.	18
Figura 5: Estructura básica de una red neuronal convolucional. Fuente. IONOS (2024).	22
Figura 6: Imagen correspondiente a una emisión geotermal con una altura registrada de 25 metros. Fuente. Atacama Rocks and tours. (s.f).	25
Figura 7: Imagen correspondiente a un volcán inactivo. Fuente. Ahora Chiapas. (2025).	26
Figura 8: Imagen correspondiente a una emisión volcánica con una altura registrada de 2,500 metros. Fuente. Radio 3 Cadena Patagonia. (s.f).	27
Figura 9 Aplicación de filtro a imágenes (Imagen real, imagen con filtro a escala de grises con modo espejo e imagen rotada levemente). Fuente: Elaboración propia.....	28
Figura 10: Imagen real de un geiser e imagen sintética de un geiser. Fuente. Atacama Tours (s.f).	28
Figura 11: Diferentes fumarolas con filtros y detección de contornos. Fuente. Elaboración propia.	30
Figura 12: Aplicación de un Umbral adaptativo, y como la pluma volcánica es más evidente. Fuente. Elaboración propia.	31
Figura 13: Placa de desarrollo ESP32. Fuente. Joy-IT (2025).....	33
Figura 14: Esquemático con pines del módulo LCD16x2. Fuente. ElectronicWings (2025).	33
Figura 15: Módulo I2C con esquema de pines. Fuente. UNIT Electronics (2025).....	34
Figura 16 : Placa de expansión con 38 pines para ESP32. Fuente. Didacticas Electronicas (2025).	35
Figura 17: Curva de dispersión, valores reales y valores estimados. Fuente. Google Colab (2025).	43
Figura 18: La curva azul muestra el entrenamiento del algoritmo y la línea naranja muestra la pérdida en la validación. Fuente. Elaboración propia.	44
Figura 19: Layout de cámara utilizado para la captura de imágenes. Fuente. Elaboración propia.	47
Figura 20: Diagrama de flujo del proceso de la ESP32. Fuente. Elaboración propia.	50
Figura 21: Diagrama de conexiones entre ESP32 y LCD. Fuente. Elaboración propia.	52
Figura 22: Instalación esp32 y tarjeta de expansión en una caja de proyectos Steren. Fuente. elaboración propia.	53
Figura 23: Instalación LCD y conexión con ESP32. Fuente. Elaboración propia.	53

Figura 24: Ensamblaje final del prototipo integrado como un sistema embebido. Fuente Elaboración propia.	54
Figura 25: Ruta cultural Asamblea de Dios. Fuente. Elaboración propia.	56
Figura 26: Ruta cultural Centro Escolar cantón el Tortuguero. Fuente. Elaboración propia.	56
Figura 27: Vista satelital del geiser El Respiradero desde Google Maps. Fuente. Google Maps (2025).	57
Figura 28: Ruta desde Atiquizaya hasta Geiser El Respiradero. Fuente. Elaboración propia.	58
Figura 29: Pruebas de campo fotografías tomadas del geiser El Respiradero. Fuente. Elaboración propia.	59
Figura 30: Pruebas de campo fotografías tomadas del geiser El Respiradero. Fuente. Elaboración propia.	59

Índice de tablas

Tabla 1: con la ubicación y tipo de volcanes que están activos en El Salvador. Fuente. MARN. (s,f).	21
Tabla 2: Pines de conexión entre la esp32 y el módulo I2C LCD. Fuente. Random Nerd tutorials (2019).	49
Tabla 3: valores de medición obtenidos. Fuente. Elaboración propia.	60

Glosario

A

Android Studio IDE: Entorno de desarrollo integrado oficial para crear aplicaciones Android, compatible con Java y Kotlin.

Arduino IDE: Entorno de desarrollo integrado para programar placas Arduino y compatibles.

ArUco Marker: Marcador cuadrado en blanco y negro con un identificador único, utilizado en visión por computadora para calibración de cámaras, seguimiento de objetos y aplicaciones de realidad aumentada.

B

Backlight: Iluminación trasera de una pantalla LCD que mejora la visibilidad en ambientes oscuros.

Batch size: Número de muestras procesadas simultáneamente durante una iteración de entrenamiento en redes neuronales.

C

Capas densas (Dense layers): Capas de una red neuronal donde cada neurona está conectada con todas las de la capa anterior.

Convolución: Operación matemática que aplica filtros a una imagen para extraer características.

D

Dataset: Conjunto de datos estructurados utilizados para entrenamiento o análisis.

Data Augmentation: Técnica para aumentar la diversidad de datos de entrenamiento mediante transformaciones.

DataFrame: Estructura de datos en forma de tabla, común en Python (pandas).

Dropout: Método de regularización que desconecta aleatoriamente neuronas durante el entrenamiento para evitar sobreajuste.

E

Epoch: Una pasada completa de todo el conjunto de datos de entrenamiento a través del modelo.

ESP32: Microcontrolador de doble núcleo con conectividad Wi-Fi y Bluetooth, ampliamente usado en IoT.

F

Fumarola: Emisión de gases volcánicos a través de grietas o conductos en la superficie del volcán.

G

Géiser: Fuente termal que expulsa agua y vapor de manera intermitente debido a la presión subterránea.

GND: Pin de referencia a tierra en circuitos electrónicos.

GPIO (General Purpose Input/Output): Pines de propósito general en un microcontrolador, configurables como entrada o salida digital.

I

I2C: Protocolo de comunicación serial que permite conectar múltiples dispositivos usando solo dos líneas.

IP (Internet Protocol): Dirección numérica que identifica un dispositivo dentro de una red.

IP estática: Dirección IP fija que no cambia con el tiempo, asignada manualmente a un dispositivo.

K

Keras: Biblioteca de alto nivel para construir y entrenar redes neuronales sobre TensorFlow.

Kotlin: Lenguaje de programación moderno, oficial para desarrollo en Android.

L

Learning rate: Parámetro que controla la magnitud de los ajustes en los pesos de la red neuronal durante el entrenamiento.

LCD16x2: Pantalla alfanumérica capaz de mostrar 16 caracteres por línea en dos filas.

M

MAE (Mean Absolute Error): Métrica de error que calcula el promedio de las diferencias absolutas.

MobileNet: Arquitectura ligera de red neuronal optimizada para dispositivos móviles.

MSE (Mean Squared Error): Métrica de error que calcula el promedio de los cuadrados de las diferencias.

N

Numpy: Biblioteca de Python para cálculo numérico y manejo de arreglos multidimensionales.

O

OpenCV: Biblioteca de visión por computadora para procesamiento de imágenes y video.

Overfitting: Situación en la que un modelo se ajusta demasiado a los datos de entrenamiento y pierde capacidad de generalización.

P

Precision: Métrica que mide la proporción de predicciones positivas correctas sobre el total de predicciones positivas.

Pooling: Técnica para reducir la dimensionalidad de una imagen manteniendo las características más relevantes.

Python: Lenguaje de programación de alto nivel, versátil y ampliamente usado en ciencia de datos.

R

R2 (Coeficiente de determinación): Métrica estadística que mide la proporción de varianza explicada por un modelo.

RGB: Modelo de color basado en la combinación de rojo (Red), verde (Green) y azul (Blue).

S

SCL: Línea de reloj en el protocolo I2C.

SDA: Línea de datos en el protocolo I2C.

Sistema embebido: Conjunto de hardware y software diseñado para realizar funciones específicas dentro de un dispositivo mayor.

T

TensorFlow: Biblioteca de código abierto para aprendizaje automático y redes neuronales.

TFLite (TensorFlow Lite): Versión optimizada de TensorFlow para dispositivos móviles y embebidos.

V

VCC: Pin de alimentación positiva en circuitos electrónicos.

Volcán: Estructura geológica formada por la acumulación de magma, gases y ceniza que emergen desde el interior de la Tierra.

Introducción

La vulcanología en El Salvador estudia un territorio marcado por la presencia de numerosos volcanes y manifestaciones geotérmicas, donde fenómenos como fumarolas, plumas volcánicas y géiseres forman parte del paisaje natural y del riesgo asociado a la actividad tectónica. Un ejemplo particular es el Cerro Cuyanausul, ubicado en el departamento de Ahuachapán, que presenta una grieta, producto de una falla geológica, que permite la expulsión de gases desde el subsuelo. Este tipo de manifestaciones fumarólicas es importante porque evidencia la interacción entre el magma profundo y los sistemas hidrotermales, ofreciendo señales de la dinámica interna de la corteza terrestre y constituyendo un recurso de estudio para comprender mejor la actividad volcánica y geotérmica en la región salvadoreña.

El desarrollo de modelos de aprendizaje automático para la detección y estimación de plumas volcánicas y fumarolas en géiseres requiere un enfoque integral que combine la construcción de un dataset representativo, técnicas de procesamiento y el uso de arquitecturas de red neuronal. En este trabajo se seleccionaron imágenes de volcanes con actividad fumarólica y géiseres, diferenciando las características cromáticas de las plumas (tonos grisáceos en erupciones volcánicas y tonos blancos en respiraderos hidrotermales), así como fotografías de volcanes sin actividad, con el fin de que el algoritmo aprende tanto la presencia como la ausencia de pluma. Dado el número limitado de imágenes disponibles, se aplicaron técnicas de Data Augmentation para ampliar el conjunto de datos y mejorar la capacidad de generalización del modelo.

El proceso de entrenamiento incluyó la normalización logarítmica de las alturas, la redimensión de las imágenes a 224×224 píxeles y la normalización de los valores RGB, lo que garantiza estabilidad y eficiencia en el procesamiento. Para la construcción del modelo se utilizó MobileNetV2 como base preentrenada, complementada con capas densas y técnicas de regulación para evitar el sobreajuste.

El entrenamiento se realizó en Google Colab con librerías como TensorFlow y Keras, empleando métricas como MSE y MAE, junto con estrategias de early stopping y fine-tuning para optimizar el rendimiento. Finalmente, el análisis de resultados mostró un buen desempeño en la predicción de alturas de plumas entre 0 y 3000 metros, aunque con limitaciones en valores superiores a 4000 metros debido a la escasez de ejemplos en el data set.

Objetivos

General

Diseñar e implementar un dispositivo basado en sistemas embebidos para la estimación de las dimensiones de la pluma volcánica del Volcán Cuyanausul.

Específicos

Desarrollar un prototipo capaz de identificar la altura de la pluma volcánica del cerro Cuyanausul.

Comprobar que el dispositivo creado pueda diferenciar distintas tonalidades de colores similares, con el fin de saber identificar el objeto de interés.

Comprobar que el dispositivo creado pueda diferenciar distintas tonalidades de colores similares, con el fin de saber identificar el objeto de interés.

Comprobar que el dispositivo creado pueda diferenciar distintas tonalidades de colores similares, con el fin de saber identificar el objeto de interés.

Planteamiento del problema

El Salvador se encuentra ubicado en una región de alta actividad tectónica y volcánica, formando parte del Cinturón de Fuego del Pacífico. Esta condición geográfica propicia la presencia de numerosos volcanes que, de manera constante o intermitente, presentan manifestaciones como emisiones de gases, fumarolas y, en algunos casos, erupciones que generan columnas de ceniza y gases de gran altura.

El monitoreo de estas manifestaciones es fundamental para comprender el comportamiento de los sistemas volcánicos y detectar posibles cambios en su actividad. Entre los parámetros de mayor relevancia se encuentra la altura de las plumas volcánicas, ya que su variación puede constituir un indicador de alteraciones en la dinámica interna del volcán y servir como apoyo para la evaluación de riesgos.

Actualmente, la estimación de la altura de estas columnas se realiza principalmente mediante observación directa, análisis de imágenes satelitales o mediante instrumentos especializados. Sin embargo, estos métodos pueden implicar altos costos de implementación, requerir infraestructura compleja o depender de recursos técnicos que no siempre están disponibles para un monitoreo continuo y accesible.

Ante esta situación, surge la iniciativa de desarrollar alternativas tecnológicas de bajo costo que permitan automatizar la estimación de la altura de las plumas volcánicas utilizando herramientas modernas de procesamiento de imágenes e inteligencia artificial. La integración de sistemas embebidos, visión por computadora y redes neuronales representa una oportunidad para crear soluciones accesibles, eficientes y adaptables a las necesidades del monitoreo volcánico.

Alcances

El presente trabajo de graduación se enfoca en el diseño y desarrollo de un prototipo de sistema embebido de bajo costo, el cual integra una cámara y un algoritmo basado en redes neuronales para la estimación de magnitudes físicas de longitud a partir de imágenes. El objetivo principal del sistema es contribuir al monitoreo de fenómenos geológicos, como emisiones de gases y plumas volcánicas.

El proyecto contempla el diseño del prototipo, la integración de componentes de hardware y software, así como el desarrollo y entrenamiento del algoritmo de procesamiento de imágenes que permita estimar la altura de las columnas de gases a partir de la información visual capturada. Asimismo, se busca la viabilidad de implementar tecnologías de visión por computadora utilizando plataformas de bajo costo.

De igual manera, el trabajo se orienta a la implementación de tecnologías innovadoras aplicadas al monitoreo volcánico, promoviendo soluciones accesibles que puedan complementar métodos tradicionales de observación.

Justificación

El Salvador se caracteriza por poseer una importante actividad volcánica debido a su ubicación dentro del Cinturón de Fuego del Pacífico. A lo largo de su territorio se encuentran diversos volcanes que presentan diferentes manifestaciones geológicas, entre ellas la emisión de gases y la formación de columnas o plumas volcánicas. Estas manifestaciones son indicadores relevantes para el estudio del comportamiento de los sistemas volcánicos, ya que las variaciones en la altura y características de dichas plumas pueden estar asociadas a cambios en la actividad interna del volcán.

El monitoreo de estos fenómenos resulta fundamental para el análisis y comprensión de la dinámica de los volcanes, así como para la identificación de posibles anomalías en su comportamiento, por tanto, la estimación de la altura de las plumas volcánicas puede proporcionar información útil para apoyar los procesos de observación y seguimiento de la actividad volcánica.

Ante esta necesidad, el presente trabajo de graduación propone el desarrollo de un prototipo basado en un sistema embebido de bajo costo que integra técnicas de visión por computadora y redes neuronales para estimar magnitudes físicas de longitud a partir del análisis de imágenes, desde el punto de vista tecnológico, este proyecto también promueve la aplicación de herramientas modernas de procesamiento de imágenes, aprendizaje automático y sistemas embebidos, integrando diferentes áreas del conocimiento para el desarrollo de soluciones innovadoras orientadas al análisis de fenómenos naturales.

Capítulo 1

1.0 Marco teórico

1.1 Antecedentes proyecto de ingeniería

“ESTIMACIÓN DEL TAMAÑO DE OBJETOS A BAJO COSTO UTILIZANDO ESP32-CAM Y VISIÓN POR COMPUTADORA”

El proyecto consistió en diseñar una solución de bajo costo empleando un módulo ESP32-CAM y marcadores ArUco previamente calibrados, utilizados como referencia de escala dentro de cada imagen capturada. A partir de las imágenes obtenidas, el sistema detecta los marcadores y convierte las distancias medidas en píxeles a unidades físicas reales, y así estimar con precisión las dimensiones de distintos objetos, el proyecto incluyó el análisis, montaje y validación experimental del sistema en condiciones controladas, demostrando que la metodología implementada puede alcanzar un error promedio inferior al 1.5%.

Metodología utilizada

A. Hardware

- ESP32-CAM: Este módulo integra un microcontrolador ESP32 con capacidades Wi-Fi y Bluetooth, acompañado de una cámara OV2640 de 2 megapíxeles. El ESP32 cuenta con un procesador de 32 bits que opera a 240 MHz, con 520 KB de SRAM interna y 4 MB de PSRAM externa. La cámara OV2640 ofrece una resolución máxima de 1600×1200 píxeles y soporta formatos de salida como JPEG y BMP. El módulo también incluye una ranura para tarjeta microSD de hasta 4 GB, facilitando el almacenamiento local de las imágenes capturadas.
- Sensores adicionales: Aunque el ESP32-CAM no incluye sensores ambientales integrados, su soporte para interfaces como UART, SPI e I2C permite la incorporación de sensores externos

para medir parámetros como temperatura, humedad y composición de gases, ampliando así las capacidades del sistema.

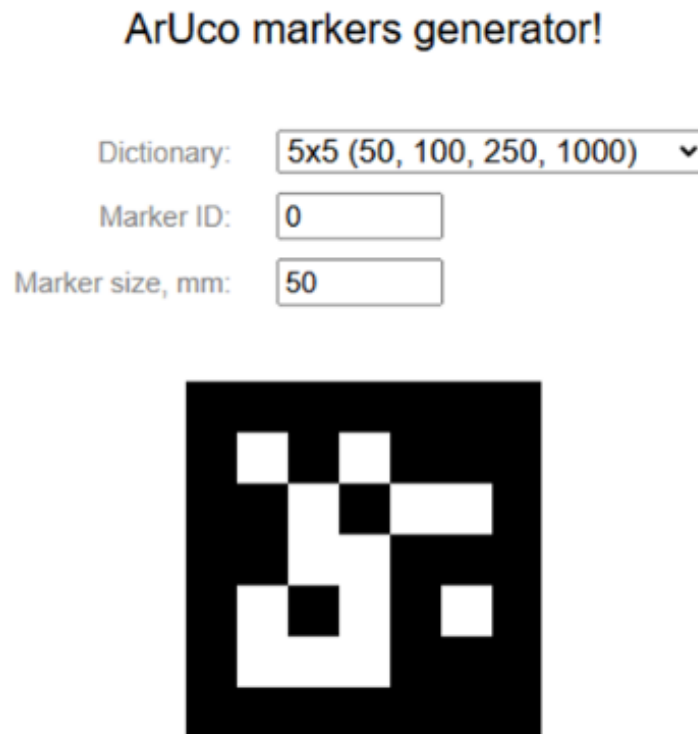
B. Software

- Python: Utilizado para procesamiento de imágenes y análisis de datos, aprovechando su amplia gama de bibliotecas científicas y de manipulación de imágenes.
- OpenCV: Esta biblioteca de código abierto proporciona herramientas para operaciones de procesamiento de imágenes como filtrado, detección de bordes, reconocimiento de características y seguimiento de objetos.
- NumPy: Biblioteca fundamental para cálculos numéricos en Python, facilitando operaciones matemáticas y manipulación de matrices, esenciales en procesamiento de imágenes
- Arduino IDE: Utilizado para programar y configurar el ESP32-CAM, permitiendo cargar firmware personalizado y gestionar la comunicación entre el hardware y el software de procesamiento.

C. Procesamiento de imágenes

1. Captura de imágenes: El ESP32-CAM se configura para tomar fotografías en tiempo real de los objetos, guardarlas en una tarjeta microSD y subirlas a un PC para un análisis adicional.
2. Detección de objetos: Las imágenes se procesan usando OpenCV para identificar y segmentar los objetos del fondo, aplicando técnicas de umbral y detección de contornos.
3. Escalas con marcadores ArUco: Se colocan marcadores ArUco, patrones binarios cuadrados fácilmente detectables en imágenes, en el entorno de la fumarola para establecer una referencia de escala. OpenCV proporciona funciones para detectar estos marcadores y calcular su orientación y posición en la imagen, en la figura 1 se muestra el marcador aruco utilizado en el proceso de toma de imágenes.

Figura 1 : Generación de un marcador aruco con ID=0, cuadros 5x5, de lado 50mm, captura de pantalla de Aruco maker. Fuente. ArUco markers generator(Chev.me, s.f).



4. Determinación de dimensiones reales: Conociendo las dimensiones físicas de los marcadores, se puede establecer la relación entre píxeles y unidades del mundo real, permitiendo estimar con precisión las dimensiones de los objetos.

Resultados

Esta sección presenta los resultados de pruebas realizadas con el sistema desarrollado para la estimación de dimensiones de objetos. Para evaluar la precisión y confiabilidad del sistema propuesto, se llevaron a cabo pruebas utilizando objetos cotidianos con dimensiones conocidas (ver Figuras 1 y 2). El objetivo fue comparar las dimensiones estimadas por el sistema con las mediciones reales y cuantificar el error bajo condiciones controladas.

Pruebas de medición:

Como parte de la experimentación se observa la figura 2 en la que se presenta la fotografía de una puerta en la que el sistema detectó sus dimensiones de altura y ancho:

Figura 2: Se presenta la medida estimada de una puerta con el equipo, en rojo se muestra el marcador aruco, se obtuvo una altura de 1.92 metros. Fuente. Elaboración propia



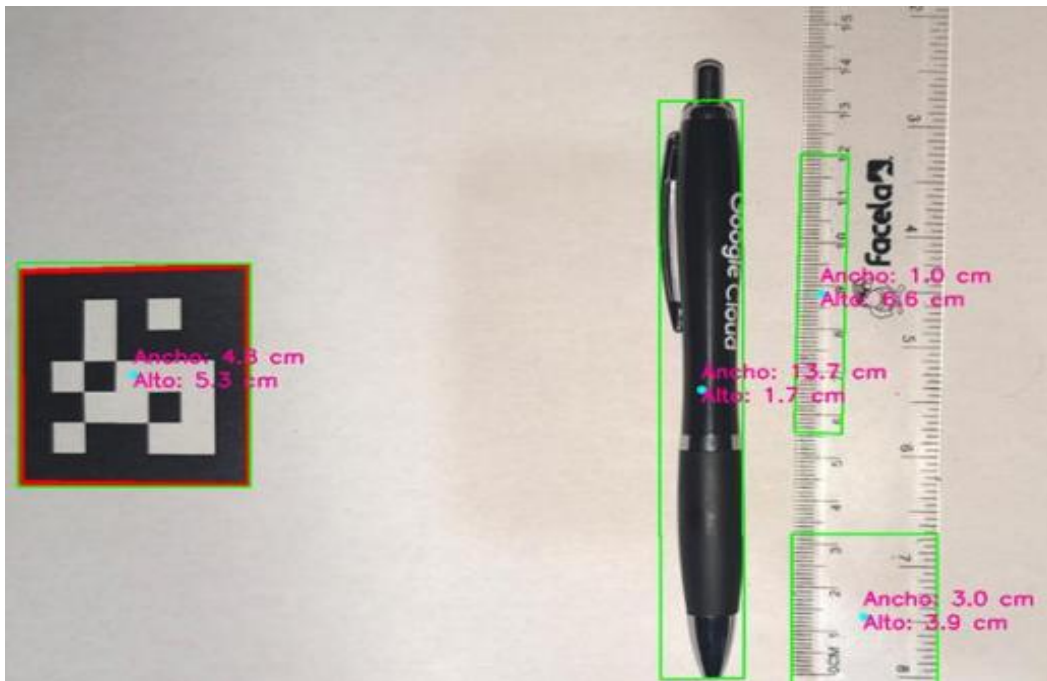
Como parte de comprobación de resultados, se tiene la medida real de la puerta, la cual se observa en la figura 3, dando un resultado de 1.92 metros.

Figura 3 Medidas reales de una puerta de 192 cm. Fuente. elaboración propia



La Figura 4 presenta una comparación entre la longitud real de un lapicero y la longitud estimada por el sistema desarrollado, con el propósito de validar la precisión del método de medición implementado. Esta prueba permite verificar la capacidad del algoritmo para convertir dimensiones en píxeles a unidades físicas reales mediante una referencia de escala previamente calibrada. La cercanía entre ambos valores demuestra que el sistema posee un error reducido y una adecuada exactitud en la estimación de longitudes en condiciones controladas.

Figura 4: Medida real de un lapicero -Vs- medida mediante el sistema. Fuente. Elaboración propia.



1.2 Vulcanología básica.

La vulcanología es la rama de la geología que estudia los volcanes, los procesos magmáticos y los fenómenos asociados a la actividad volcánica, como erupciones, fumarolas, fuentes termales y géiseres. En El Salvador, es de especial importancia debido a su ubicación dentro del Cinturón de Fuego, una de las zonas tectónicamente más activas del planeta.

Dentro de este contexto, la vulcanología en El Salvador no solo se enfoca en las erupciones, sino también en manifestaciones volcánicas secundarias como fumarolas, manantiales termales y campos geotérmicos, los cuales evidencian la presencia de calor interno en el subsuelo. Estas manifestaciones son claves para el estudio de los géiseres, que, aunque no son comunes en El Salvador, comparten el mismo origen geológico.

Los géiseres son emisiones periódicas de agua caliente y vapor que ocurren cuando el agua subterránea entra en contacto con rocas calentadas por el magma, acumulando presión hasta ser expulsada violentamente a la superficie. En El Salvador, las condiciones geotérmicas existen especialmente en zonas volcánicas y campos geotérmicos como Ahuachapán y Berlín.

Las fumarolas son respiraderos en la superficie terrestre donde se emiten gases como vapor de agua (H₂O), dióxido de azufre (SO₂), dióxido de carbono (CO₂) entre otros.

Geiser: es una abertura en la corteza donde recorre una fuente termal el cual expulsa columnas de agua hirviendo debido a altas presiones y temperatura en las fuentes subterráneas de agua.

Pluma volcánica: son columnas de gases, cenizas y otras partículas emitidas por volcanes durante una erupción.

1.3 Importancia del monitoreo:

Las erupciones volcánicas son uno de los agentes de cambio más dramáticos y violentos de la Tierra. Las potentes erupciones explosivas no solo pueden alterar drásticamente la tierra y el agua en decenas de kilómetros alrededor de un volcán, sino que las diminutas gotas de ácido sulfúrico que entran en erupción en la estratosfera pueden cambiar temporalmente el clima de nuestro planeta.

Debido a que contienen gases volcánicos, las fumarolas proporcionan información importante a los geólogos sobre los volcanes activos. Un aumento en la actividad de una fumarola a menudo se

asocia con disturbios y puede indicar que el magma se está moviendo dentro del volcán. Los vulcanólogos monitorean la temperatura de las fumarolas, así como la composición de los gases.

El monitoreo volcánico en El Salvador se realiza mediante diversas técnicas que permiten evaluar la actividad interna y superficial de los volcanes. Entre ellas destacan la vigilancia sísmica, que registra la ocurrencia de sismos volcánicos; el análisis de gases, que mide la concentración de dióxido de azufre (SO₂), dióxido de carbono (CO₂) y sulfuro de hidrógeno (H₂S); la química de aguas, que estudia cambios en pH y conductividad en fuentes cercanas; y la observación de fumarolas, deformaciones del cono y emisiones de ceniza. Estas metodologías, combinadas entre sí, proporcionan información crítica para la generación de alertas tempranas y la reducción de riesgos en comunidades cercanas a volcanes activos (SNET, 2025).

1.4 Vulcanología en El Salvador

Origen de los volcanes

El origen del vulcanismo en El Salvador se explica por la subducción de la placa de Cocos bajo la placa del Caribe. Este proceso tectónico provoca la fusión parcial de materiales en el manto superior, generando magma que asciende hacia la superficie y forma la cadena volcánica del país. La interacción constante de estas placas es la responsable de la alta densidad de volcanes activos y de la recurrencia de erupciones que han marcado la historia geológica del territorio salvadoreño (MARN, s.f.).

Principales volcanes activos

El Salvador cuenta con varios volcanes considerados activos, entre los que destacan Santa Ana, San Salvador, San Miguel, Izalco, San Vicente y la caldera de Ilopango. Estos volcanes, de tipo estratovolcán y caldera, presentan diferentes estilos eruptivos como actividad freatomagmática, estromboliana, efusiva y fumarólica. Sus erupciones, registradas en fechas recientes como 2005 en Santa Ana y 2013 en San Miguel, evidencian la constante interacción tectónica que caracteriza al país y explican la necesidad de un monitoreo permanente para prevenir riesgos a la población, estos datos se presentan en la tabla 1. (MARN, s.f.).

Tabla 1: con la ubicación y tipo de volcanes que están activos en El Salvador. Fuente. MARN. (s.f).

Volcán	Departamento	Altitud msnm	Tipo	Actividad	Ultima erupción
Santa Ana	Santa Ana - Sonsonate	2381	Estratovolcán	Freamamática	2005
San Salvador	San Salvador - La Libertad	1890	Estratovolcán	Freamamática	1917
San Miguel	San Miguel	2130	Estratovolcán	Estromboliana	2013
Izalco	Sonsonate	1950	Estratovolcán	Estromboliana	1966
San Vicente	San Vicente - La Paz San Salvador - Cuscatlán	2182	Estratovolcán	Fumarólica Pliniana fratoplíniana y	
Ilopango	- La Paz	450	Caldera	efusiva	1880

1.5 Redes Neuronales Convolucionales

Las redes neuronales constituyen un subconjunto del machine learning y representan la base de los algoritmos de deep learning. Su estructura está formada por capas de nodos interconectados, que incluyen una capa de entrada, una o varias capas ocultas y una capa de salida. Cada nodo procesa la información mediante pesos y umbrales, activándose únicamente cuando el valor resultante supera un límite definido, lo que permite la propagación de los datos a través de la red.

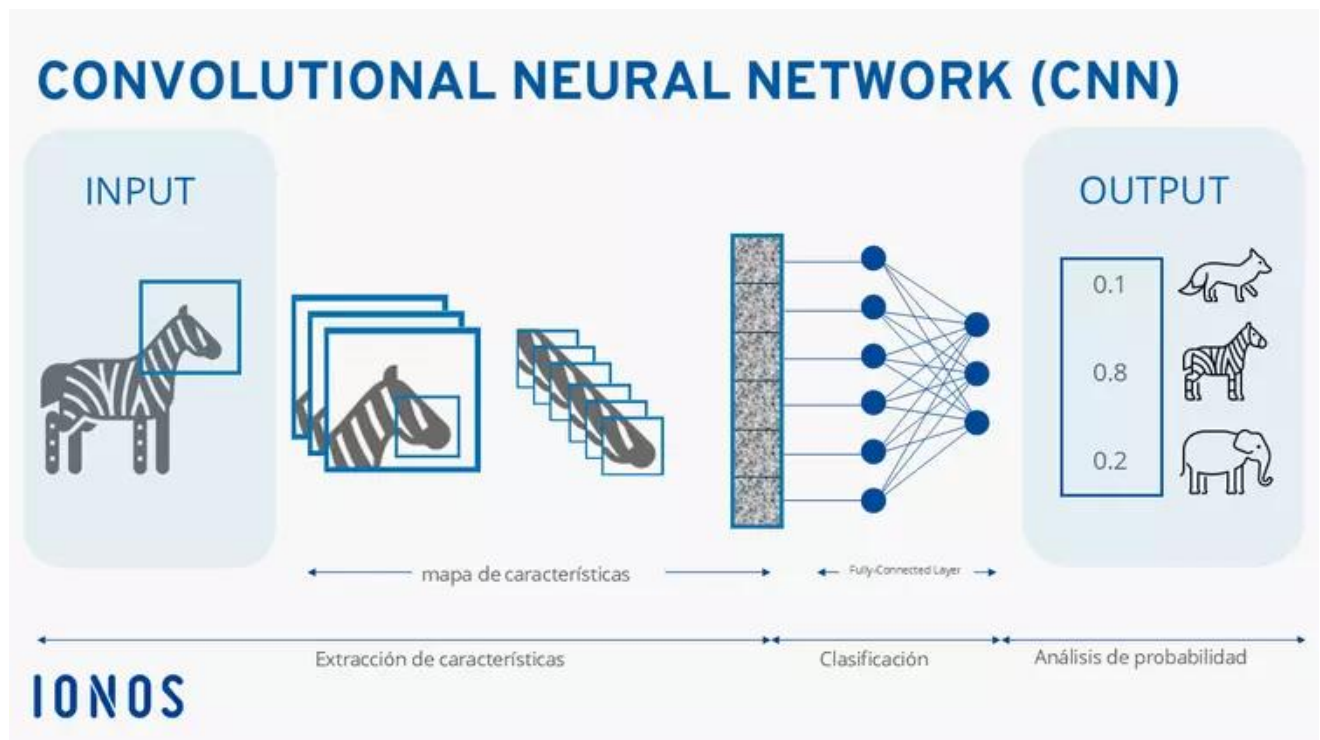
Existen diversos tipos de redes neuronales diseñadas para distintos tipos de datos y aplicaciones. Las redes neuronales recurrentes se emplean principalmente en tareas de procesamiento del lenguaje natural y reconocimiento de voz, mientras que las redes neuronales convolucionales son ampliamente utilizadas en clasificación de imágenes y visión por computadora. A diferencia de los métodos tradicionales de extracción manual de características, las redes neuronales convolucionales permiten identificar patrones de forma automática y escalable, utilizando principios de álgebra lineal, como la multiplicación de matrices, para el análisis de imágenes. No obstante, su entrenamiento suele requerir un alto consumo de recursos computacionales, especialmente el uso de unidades de procesamiento gráfico.

Las redes neuronales convolucionales se caracterizan por su alto rendimiento en el procesamiento de señales visuales, de audio y de voz. Su arquitectura se compone principalmente de capas convolucionales, capas de agrupación y capas totalmente conectadas. A medida que la información avanza por estas capas, la red incrementa progresivamente su nivel de abstracción, comenzando

con la detección de características simples, como bordes y colores, y evolucionando hacia el reconocimiento de formas complejas y objetos completos.

La Figura 5 ilustra la estructura básica de una red neuronal artificial, mostrando la organización de sus principales componentes: la capa de entrada, una o varias capas ocultas y la capa de salida. La capa de entrada recibe las variables iniciales del sistema, como las características extraídas de las imágenes, posteriormente, la información se procesa en las capas ocultas mediante neuronas interconectadas, donde cada conexión posee un peso que se ajusta durante el entrenamiento. Finalmente, la capa de salida genera la predicción correspondiente, en este caso, la estimación de la altura de la pluma volcánica.

Figura 5: Estructura básica de una red neuronal convolucional. Fuente. IONOS (2024).



1.6 Sistemas embebidos

Un sistema embebido puede definirse como un sistema electrónico y computacional diseñado para ejecutar una tarea específica en tiempo real, integrando en un mismo dispositivo el procesador, la memoria y las interfaces de entrada y salida. A diferencia de las computadoras de propósito general, que pueden ejecutar múltiples aplicaciones, los sistemas embebidos están optimizados para cumplir

una función concreta con alta eficiencia y confiabilidad, lo que los hace indispensables en ámbitos como la automatización, la comunicación y el control de procesos. (TRBL Services, 2021).

En el caso particular de este proyecto, el sistema embebido se ha orientado al procesamiento de imágenes y comunicación en redes, aplicado al monitoreo volcánico. El dispositivo desarrollado cumple la función de recibir imágenes del fenómeno volcánico y mostrar el dato de interés: la altura de la pluma volcánica en metros.

Para lograrlo, se ha implementado un ESP32, un microcontrolador con conectividad Wi-Fi y Bluetooth, que actúa como núcleo del sistema. Este se complementa con una pantalla LCD16x2 con módulo I2C, que permite visualizar los resultados de manera sencilla y eficiente. El procesamiento de imágenes se realiza mediante OpenCV y Numpy, mientras que el análisis inteligente se apoya en TensorFlow y Keras, utilizando arquitecturas optimizadas como MobileNetV2. El entrenamiento del modelo se llevó a cabo en Google Colab, aplicando técnicas de data augmentation para mejorar la precisión y algoritmos de optimización como Adam para ajustar los parámetros de la red neuronal.

La integración de estos componentes permite que el sistema embebido procese imágenes en tiempo real, identifique la pluma volcánica y calcule su altura, transmitiendo la información de manera confiable. De esta forma, el proyecto aporta una herramienta tecnológica que complementa las técnicas tradicionales de monitoreo volcánico en El Salvador, integrando visión por computadora, inteligencia artificial y comunicación en red dentro de un dispositivo compacto y especializado.

Capítulo 2

2.0 Metodología

2.1 Data set

El conjunto de datos utilizado en esta investigación está compuesto por imágenes que muestran volcanes y géiseres con actividad fumarólica. Estas imágenes fueron recopiladas de diferentes fuentes de internet y corresponden a diversos eventos eruptivos registrados en distintos volcanes alrededor del mundo. Para cada imagen se tomó como referencia la información reportada sobre la altura de la pluma volcánica de cada evento, la cual fue utilizada como dato de referencia dentro del proceso de análisis.

El objetivo de incluir diferentes tipos de imágenes fue permitir que el algoritmo pudiera distinguir entre distintos tipos de emisiones en sistemas volcánicos y geotermales. Por lo cual, se incorporaron imágenes que presentan fumarolas grisáceas asociadas a erupciones volcánicas, así como fumarolas blancas provenientes de respiraderos hidrotermales o géiseres.

Para usar nuestro dataset necesitamos dos atributos de cada registro: el nombre y la altura. A partir del dataset se crea un DataFrame, que es una estructura bidimensional organizada en filas y columnas. Los datos suelen estar separados por comas y se almacenan en un archivo con extensión .csv, el cual puede ser importado y manipulado en entornos como Google Colab mediante la librería pandas.

Algunas de las imágenes utilizadas se muestran en la figura 6:

Figura 6: *Imagen correspondiente a una emisión geotermal con una altura registrada de 25 metros. Fuente. Atacama Rocks and tours. (s.f).*



Asimismo, dentro del conjunto de datos también se incluyeron imágenes de volcanes inactivos como se presenta en la figura 7, asignándoles un valor de altura de 0 metros. Esto permitió que el algoritmo aprendiera a diferenciar entre valores positivos, correspondientes a la presencia de una pluma volcánica, y valores nulos, que representan la ausencia de dicha pluma. De esta manera, el conjunto de datos proporciona una base variada de escenarios que contribuye al entrenamiento y evaluación del modelo propuesto en la investigación.

Figura 7: Imagen correspondiente a un volcán inactivo. Fuente. Ahora Chiapas. (2025).



La Figura 8 presenta una imagen correspondiente a una emisión volcánica con una altura de pluma registrada de 2,500 metros, utilizada como parte del conjunto de datos para el entrenamiento del modelo. Esta imagen representa un caso de actividad volcánica con una columna de gases claramente definida, lo que permite identificar características visuales relevantes como forma, densidad y extensión vertical. Desde el punto de vista del procesamiento de imágenes, este tipo de muestras es esencial para que la red neuronal aprenda a asociar patrones visuales con valores reales de altura.

Figura 8: Imagen correspondiente a una emisión volcánica con una altura registrada de 2,500 metros. Fuente. Radio 3 Cadena Patagonia. (s.f).



2.2 Data Augmentation

Con el objetivo de mejorar la capacidad de generalización del modelo y aumentar la cantidad de datos disponibles para el entrenamiento se aplicaron técnicas de data augmentation o aumento de datos al conjunto de imágenes. Este proceso consiste en generar nuevas muestras a partir de las imágenes originales mediante la aplicación de diferentes transformaciones, esto permite ampliar el conjunto de datos sin necesidad de recopilar nuevas imágenes, o como en este caso, por falta de datos de alturas de eventos registrados.

Las transformaciones aplicadas incluyen ajustes como rotación de imágenes, variaciones en el brillo y contraste, así como posibles cambios de escala y orientación. Estas modificaciones permiten simular diferentes condiciones de observación, tales como variaciones en la iluminación, ángulos de captura o condiciones atmosféricas, que pueden presentarse al momento de registrar imágenes de fenómenos volcánicos o hidrotermales.

La aplicación de estas técnicas contribuye a que el modelo tenga más aprendizaje frente a variaciones presentes en las imágenes, reduciendo el riesgo de sobreajuste y mejorando su

capacidad para reconocer patrones asociados a la presencia o ausencia de plumas volcánicas en diferentes escenarios. Como se muestra en la figura 9 (secuencia de imágenes) y figura 10.

Figura 9 *Aplicación de filtro a imágenes (Imagen real, imagen con filtro a escala de grises con modo espejo e imagen rotada levemente). Fuente: Elaboración propia.*



La figura 10 presenta una comparación entre una imagen real de un géiser y una imagen sintética generada a partir de técnicas de aumento de datos. Esta representación permite observar cómo a partir de una imagen original, es posible crear nuevas imágenes de muestra que conservan las características esenciales del fenómeno, como la forma y distribución de la pluma.

Figura 10: *Imagen real de un geiser e imagen sintética de un geiser. Fuente. Atacama Tours (s.f).*



2.3 Normalización de datos

Durante la etapa de preprocesamiento de los datos se aplicaron diferentes técnicas de normalización con el objetivo de facilitar el entrenamiento del modelo y mejorar la estabilidad del algoritmo. La normalización permite transformar los valores originales del conjunto de datos a escalas más manejables, lo que favorece la convergencia del modelo y mejora su capacidad para identificar patrones en la información.

En el caso de las alturas de las plumas volcánicas, se aplicó una normalización mediante la función logarítmica natural. Este tipo de transformación permite reducir la escala de valores muy grandes y estabilizar la distribución de los datos. La transformación utilizada se expresa en la Eq. 1 de la siguiente forma:

$$altura_log = \ln \ln (h) . \text{ Eq. 1.}$$

Sin embargo, debido a que dentro del conjunto de datos existen valores de altura iguales a cero (correspondientes a imágenes donde no existe pluma volcánica) fue necesario ajustar la expresión para evitar errores matemáticos, ya que el logaritmo natural de cero no está definido. Por esta razón se utilizó la siguiente transformación que se observa en la Eq. 2:

$$altura_log = \ln \ln (h + 1) \text{ Eq. 2.}$$

De esta manera se logra mantener los valores de cero dentro del conjunto de datos y al mismo tiempo reducir la diferencia entre valores muy pequeños y grandes, permitiendo que el algoritmo procesa la información de forma más estable.

Adicionalmente, las imágenes utilizadas en el conjunto de datos fueron redimensionadas a una resolución de 224×224 píxeles. Este procedimiento permite reducir el consumo de memoria y procesamiento durante el entrenamiento del modelo, manteniendo al mismo tiempo suficiente información visual para el análisis de las plumas volcánicas.

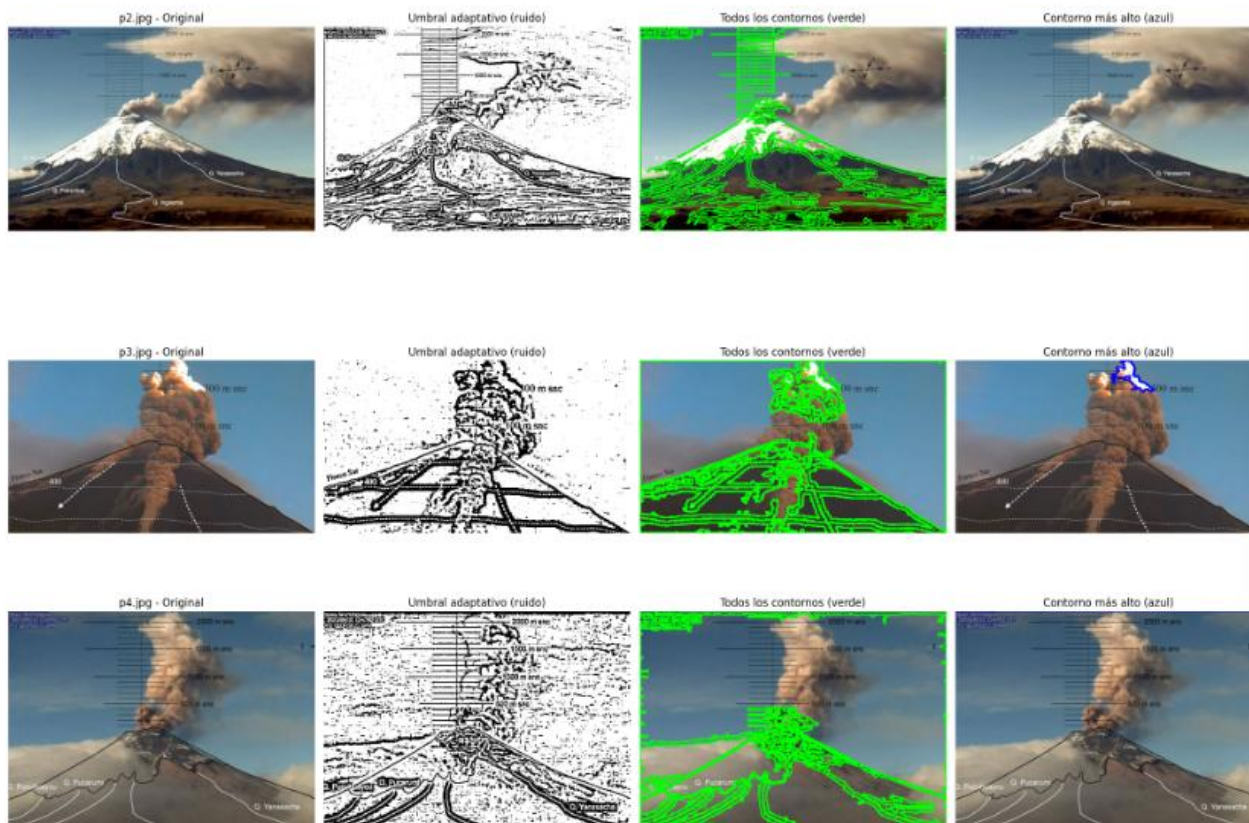
Posteriormente, se realizó la normalización de los valores de los píxeles de cada imagen. En una imagen digital, cada píxel posee valores RGB que normalmente se encuentran en un rango entre 0 y 255. Para facilitar el procesamiento por parte de la red neuronal, estos valores se escalaron a un rango entre 0 y 1, dividiendo cada valor entre 255. De esta forma, cada píxel queda representado por tres valores normalizados correspondientes a los canales RGB

2.4 Detección de la fumarola

Durante la construcción del primer dataset se seleccionaron imágenes que incluían una escala de altura visible dentro de la fotografía, con el propósito de garantizar que las imágenes de entrenamiento contarán con una referencia de altura verdadera. Sin embargo, este enfoque generó un inconveniente: la presencia de la escala introducía “ruido” en el procesamiento de la imagen, afectando la calidad de los datos.

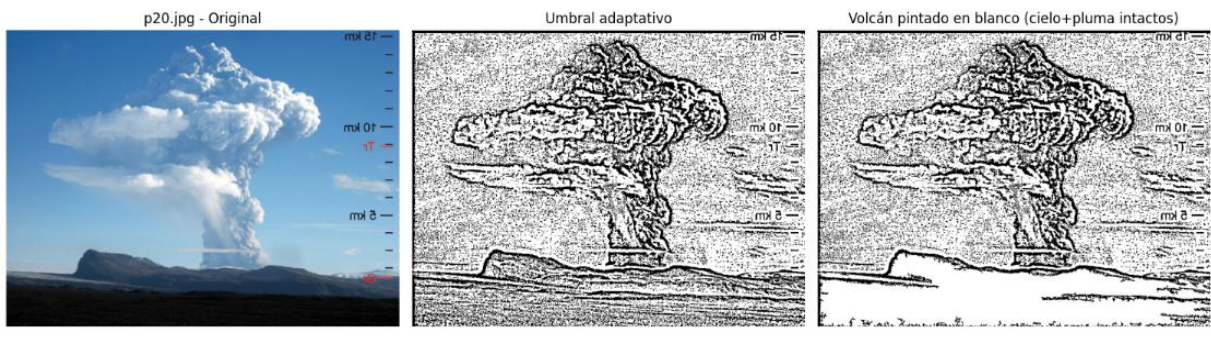
En la Figura 11 se observa cómo la escala dentro de la imagen interfiere en el análisis, añadiendo elementos no deseados que dificultan la correcta segmentación. Por esta razón, se decidió sustituir el dataset inicial por un conjunto de imágenes más limpio, en el cual se eliminaron dichas referencias visuales. El objetivo fue mantener únicamente los tres escenarios relevantes para el modelo: cielo, fumarola y volcán, logrando así un entrenamiento más robusto y libre de interferencias externas.

Figura 11: *Diferentes fumarolas con filtros y detección de contornos. Fuente. Elaboración propia.*



La Figura 12 muestra la aplicación de una técnica de umbralización adaptativa sobre una imagen volcánica. Este método de procesamiento divide la imagen en regiones y calcula un valor de umbral local para cada una, lo que permite resaltar elementos de interés incluso cuando existen variaciones de iluminación o contraste. En este caso, la pluma volcánica se vuelve más evidente y diferenciable del fondo, facilitando su identificación y segmentación.

*Figura 12: Aplicación de un Umbral adaptativo, y como la pluma volcánica es más evidente.
Fuente. Elaboración propia.*



2.5 Software

El proceso de entrenamiento del modelo se realizó en Google Colab, una plataforma de Google que permite ejecutar código en Python directamente desde el navegador sin necesidad de instalación local. Esta herramienta ofrece acceso gratuito a recursos de hardware como CPU, GPU y TPU, lo que facilita la implementación de algoritmos de aprendizaje automático y procesamiento de imágenes de manera eficiente. En este proyecto se emplearon librerías esenciales como NumPy, OpenCV, Keras y TensorFlow, que proporcionan estructuras de datos, funciones matemáticas avanzadas, herramientas de visión por computadora y frameworks de redes neuronales. Google Colab también ofrece memoria RAM, memoria de video (VRAM) y almacenamiento temporal, lo cual permitió realizar operaciones de normalización, aumento de datos y entrenamiento de modelos de clasificación sin necesidad de infraestructura física especializada (Abadi et al., 2016; Google Colab, 2024).

Para la captura de imágenes se utilizó Android Studio, el entorno oficial de desarrollo para aplicaciones móviles en Android. En este trabajo se configuró con el lenguaje de programación Kotlin y el plugin de Gradle versión 8.1.2.3, asegurando compatibilidad con dispositivos que ejecutan versiones de Android API 21 en adelante (Android 5.0). Se integraron librerías como CameraX, que permiten un control avanzado de la cámara del dispositivo, y módulos de

conectividad Wi-Fi para establecer comunicación con el microcontrolador ESP32. Android Studio proporcionó las herramientas necesarias para compilar, depurar y desplegar la aplicación móvil, garantizando la interacción fluida entre el sensor de cámara y el sistema de transmisión de datos ([Android Developers, 2025](#)).

Finalmente, la programación del microcontrolador ESP32 se realizó en Arduino IDE, un entorno de desarrollo ampliamente utilizado para proyectos de electrónica y sistemas embebidos. Arduino IDE ofrece una interfaz sencilla para escribir, compilar y cargar código en placas de desarrollo, soportando múltiples bibliotecas que facilitan la integración de sensores, módulos de comunicación y periféricos. En este trabajo se configuró el ESP32 como punto de acceso Wi-Fi con una dirección IP estática, lo que permitió la comunicación directa con el dispositivo móvil. La elección de Arduino IDE se debió a su compatibilidad con la comunidad de desarrollo de Espressif y a la disponibilidad de librerías específicas para el ESP32, lo que simplificó la implementación del sistema de transmisión de datos ([Arduino, 2025](#)).

2.6 Hardware

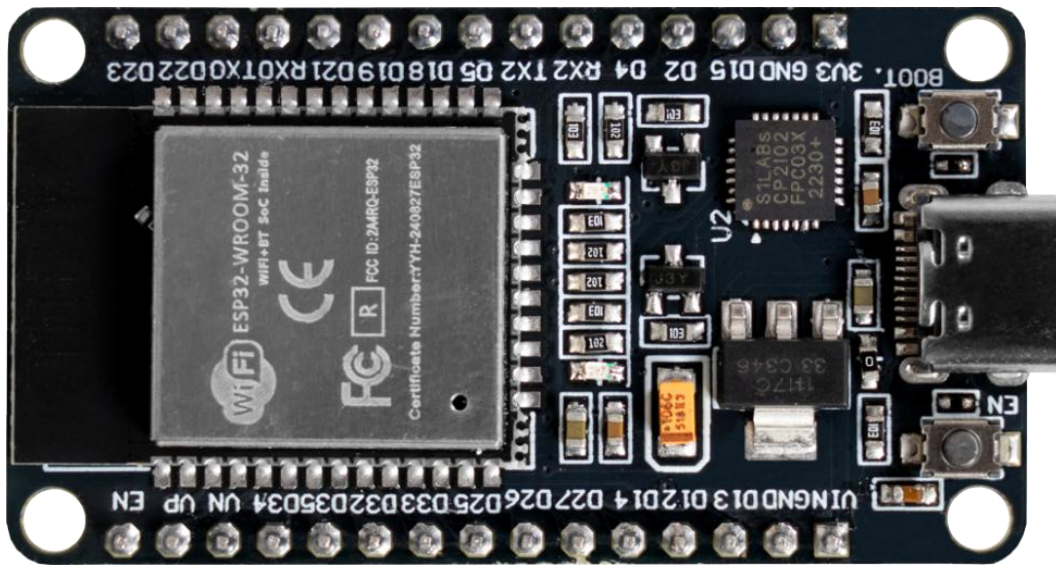
Esp32

Para la implementación del sistema se utilizó una placa de desarrollo ESP32 como la que se muestra en la Figura 13, que sirvió como núcleo del dispositivo. Esta placa se complementa con una placa de expansión, la cual permitió realizar las conexiones de manera más ordenada y estable, facilitando la integración de periféricos adicionales.

Características:

- Conexión Wi-Fi 802.11
- Conexión Bluetooth V4.2
- Ram 520 kB
- Flash externo hasta 16 MB
- Mas de 30 pines programables
- Aplicaciones IOT, domótica, transmisión de datos.
- Alimentación 3.3V o 5 V

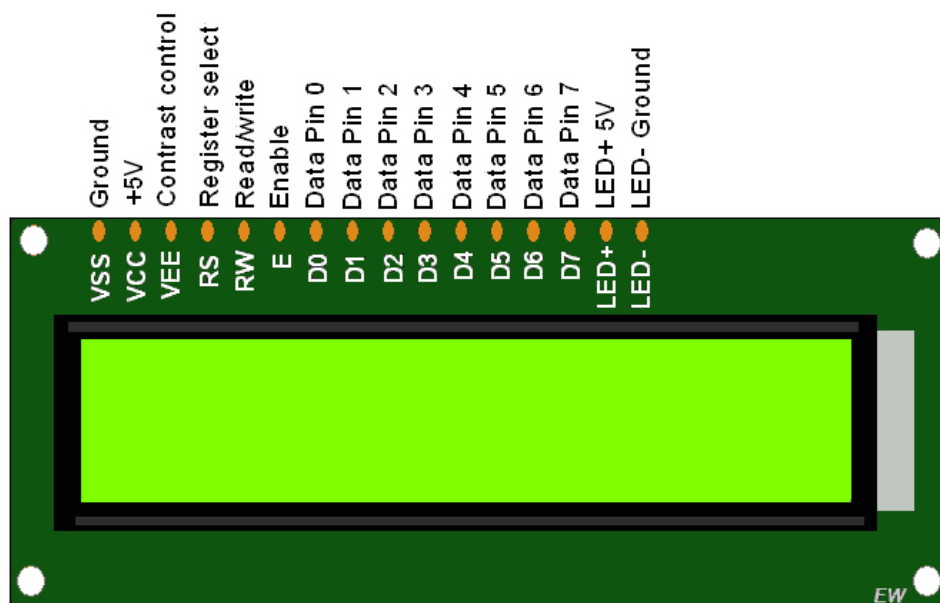
Figura 13: Placa de desarrollo ESP32. Fuente. Joy-IT (2025).



LCD1602

Destinada a mostrar información básica del sistema en tiempo real, como estados de conexión y mensajes de operación. Las conexiones eléctricas se realizaron mediante cables Dupont, que ofrecieron una solución práctica y flexible para enlazar la placa principal con la pantalla y otros componentes.

Figura 14: Esquemático con pines del módulo LCD16x2. Fuente. ElectronicWings (2025).



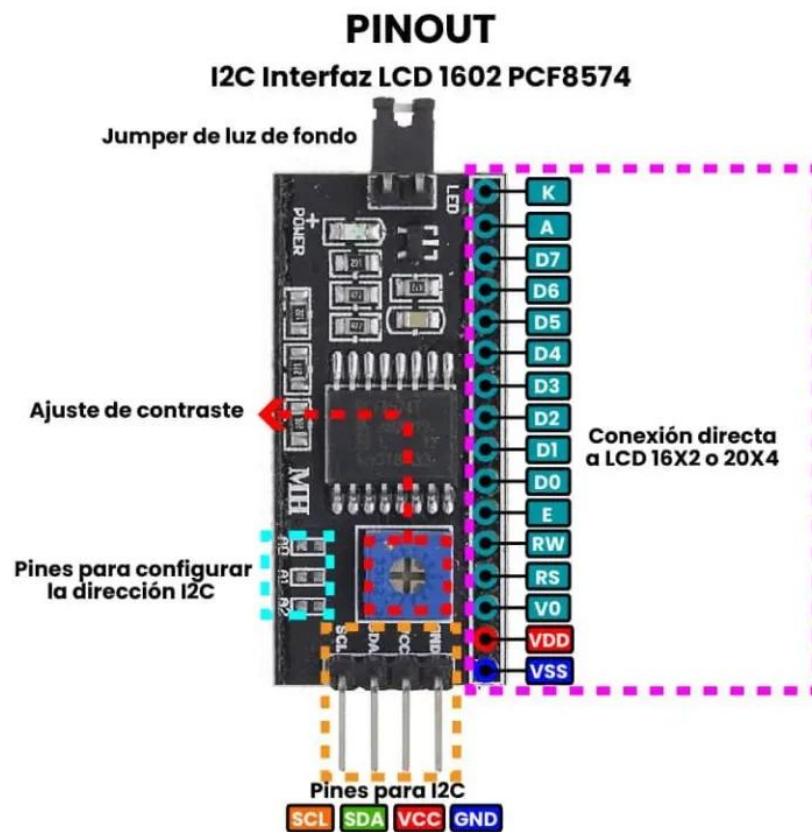
I2C

Es un adaptador que permite controlar pantallas LCD16x2, utilizando solamente 2 pines de comunicación SDA y SCL, lo cual reduce la utilización de pines, tal como se presenta en la figura 15.

Características:

- Comunicación I2C
- Voltaje de operación 5V
- Pines de conexión SDA, SCL, VCC y GND
- Compatibilidad LCD 16x2 y 20x4
- Reducción de pines
- Potenciómetro integrado para contraste.

Figura 15: Módulo I2C con esquema de pines. Fuente. UNIT Electronics (2025).



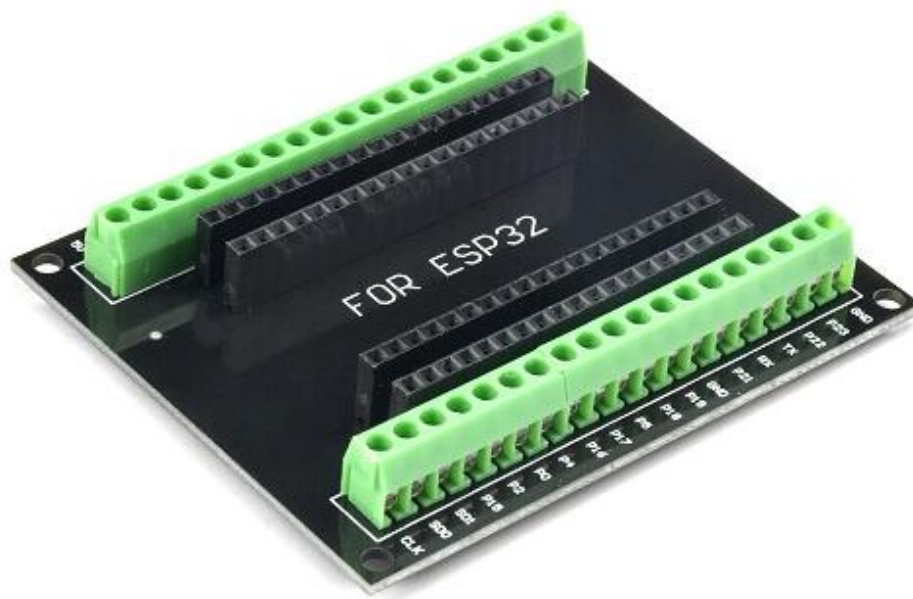
Base de expansión

Como se muestra en la Figura 16, la base de expansión es un módulo diseñado para facilitar la conexión de periféricos y sensores al microcontrolador, teniendo la oportunidad de realizar la conexión por cable DuPont o con bornera.

Características:

- Diseñada para ESP32
- Regleta de pines DuPont
- Bornera de tornillo
- Orificios para anclaje en bases

Figura 16 : Placa de expansión con 38 pines para ESP32. Fuente. Didacticas Electronicas (2025).



Finalmente, todo el conjunto se montó en una caja protectora, que aseguró la integridad física del sistema y permitió un uso más seguro en condiciones de campo. Este diseño buscó garantizar tanto la funcionalidad como la durabilidad del dispositivo.

El ESP32 fue el componente central del sistema. Se trata de un microcontrolador de bajo costo y alto rendimiento desarrollado por Espressif Systems, que integra conectividad Wi-Fi y Bluetooth,

un procesador dual-core de 32 bits y soporte para protocolos TCP/IP. Estas características lo convierten en una solución versátil para aplicaciones de Internet de las Cosas (IoT).

En este proyecto, el ESP32 se configuró como punto de acceso Wi-Fi con dirección IP estática, lo que permitió la comunicación directa entre el dispositivo móvil y el sistema de captura de imágenes. Su bajo consumo energético y capacidad de procesamiento lo hacen especialmente adecuado para proyectos que requieren transmisión de datos en tiempo real y autonomía operativa. La programación del ESP32 se realizó en Arduino IDE, un entorno de desarrollo ampliamente utilizado en sistemas embebidos, que ofrece compatibilidad con librerías específicas y una interfaz sencilla para compilar y cargar código en la placa ([Espressif Systems, 2025](#)).

Capítulo 3

3.0 Entrenamiento del modelo

Para el entrenamiento del modelo de red neuronal se utilizó Google Colab, Esta herramienta permite desarrollar y ejecutar código en lenguaje Python directamente desde un navegador web, sin necesidad de instalar software adicional en el equipo local.

Una de las principales ventajas de Google Colab es que proporciona acceso gratuito a recursos de hardware de alto rendimiento, como unidades de procesamiento gráfico (GPU) y, en algunos casos, unidades de procesamiento tensorial (TPU). Estos recursos son especialmente útiles en aplicaciones de aprendizaje profundo, ya que aceleran considerablemente el proceso de entrenamiento de redes neuronales al realizar operaciones matemáticas complejas de manera paralela.

Además, la plataforma ofrece un entorno preconfigurado con múltiples bibliotecas especializadas en ciencia de datos, visión por computadora y aprendizaje automático, entre las que destacan

NumPy, OpenCV, TensorFlow y Keras. Esto facilita la implementación del modelo, el procesamiento de imágenes y la evaluación de resultados.

3.1 Entrenamiento del algoritmo

División del dataset:

Se separa el conjunto de imágenes en dos partes: 80% para entrenamiento, donde el modelo aprende los patrones y características relevantes. 20% para validación, que permite evaluar el desempeño del modelo en datos no vistos y evitar el sobreajuste (overfitting).

Entrenamiento por lotes (batch training):

En lugar de procesar una sola imagen por iteración, se utilizan lotes de 8 imágenes seleccionadas aleatoriamente, esto mejora la eficiencia computacional, aprovecha mejor la memoria y estabiliza el cálculo de los gradientes durante la retro propagación.

Normalización de imágenes:

Los valores de cada canal de color RGB se transforman a un rango entre 0 y 1, esta técnica homogeneiza los datos de entrada, evita que valores altos dominen el entrenamiento y facilita la convergencia del algoritmo de optimización (por ejemplo, Adam).

Beneficios del proceso:

- Mayor estabilidad en el entrenamiento.
- Reducción del riesgo de sobreajuste.
- Mejor capacidad de generalización del modelo.
- Entrenamiento más eficiente y preciso.

```
1. # Transformación log1p (log natural de 1+x)
2. altura_log = np.log1p(altura)

3. # Dataset
4. dataset = tf.data.Dataset.from_tensor_slices((img_nombres, altura_log))

5. #preprocesamiento de imagenes, normalizacion
6. def abrir_img(path, label):
7. img = tf.io.read_file(path)
8. img = tf.image.decode_image(img, channels=3, expand_animations=False)
9. img = tf.image.resize(img, [224, 224]) / 255.0
10. return img, label
```

```

11. dataset = dataset.map(abrir_img)
12. # Dataset 80% entrenamiento y 20% validacion
13. entrenamiento_size = int(0.8 * len(df))
14. entrenamiento_ds = dataset.take(entrenamiento_size).shuffle(100).batch(8)
15. val_ds = dataset.skip(entrenamiento_size).batch(8)

```

3.2 Proceso de construcción de modelo

Con MobileNetV2 se aprovecha un modelo ya preentrenado en grandes bases de datos de imágenes (como ImageNet), lo que permite reutilizar el conocimiento adquirido en tareas de clasificación visual y aplicarlo como base para un nuevo algoritmo. Este enfoque se conoce como transfer learning. En la práctica, las capas convolucionales iniciales del modelo que ya han aprendido a detectar características generales como bordes, texturas y formas se congelan (es decir, sus pesos no se modifican durante el entrenamiento), de modo que actúan como un extractor de características fijo. Sobre estas capas se añaden nuevas capas densas o clasificadoras adaptadas al problema específico, en tu caso la medición de la pluma volcánica.

El flujo se organiza de la siguiente manera:

- Input shape: define el tamaño y la estructura de las imágenes de entrada (por ejemplo, 224×224 píxeles con 3 canales RGB).
- Weights: corresponden a los parámetros aprendidos por MobileNetV2 en su entrenamiento original; al congelarlos, se mantienen estables y sirven como base sólida.
- Capas adicionales: se agregan al final del modelo para ajustar la salida al nuevo objetivo, entrenándose únicamente estas capas con tu dataset específico.

```

1. # Base preentrenada con mobilenet
2. base_model = MobileNetV2(
3. input_shape=(224, 224, 3),
4. include_top=False,
5. weights='imagenet'
6. )
7. base_model.trainable = False

```

Seguidamente, en el modelo final se reduce el mapa de características a un vector, agregamos capas densas y conectadas para aprender patrones, regulación para evitar un sobreajuste y, en la salida, tenemos un solo valor: en este caso, la altura de la pluma volcánica.

3.3 presentación del modelo final

El modelo final se construye bajo un esquema secuencial, lo que permite apilar capas de manera ordenada y coherente. En primer lugar, se incorpora una capa de GlobalAveragePooling2D, cuya función es reducir el mapa de características generado por MobileNetV2 a un vector compacto. Este paso es crucial porque condensa la información espacial de la imagen en valores promedio, eliminando redundancias y facilitando que las siguientes capas trabajen con datos más manejables. Posteriormente, se añaden capas densas totalmente conectadas, que permiten aprender patrones más complejos y específicos relacionados con la altura de la pluma volcánica. Estas capas utilizan la función de activación ReLU, que introduce no linealidad y mejora la capacidad del modelo para capturar relaciones más sofisticadas. Para evitar el sobreajuste, se incluye la técnica de Dropout, que desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento, obligando al modelo a generalizar mejor. Finalmente, la capa de salida se diseña para producir un único valor numérico, correspondiente a la altura estimada de la pluma volcánica en metros.

En cuanto al proceso de optimización, se emplea el optimizador Adam con una tasa de aprendizaje de 1×10^{-4} , este valor se selecciona porque ofrece un equilibrio entre velocidad y estabilidad: valores más altos pueden acelerar el entrenamiento, pero tienden a generar inestabilidad en la convergencia; valores más bajos ralentizan el proceso, aunque aportan mayor estabilidad y precisión en el ajuste de los parámetros. Adam adapta dinámicamente la tasa de aprendizaje para cada parámetro, lo que lo convierte en una opción robusta para problemas de visión por computadora.

La función de pérdida principal utilizada es el Mean Squared Error (MSE), que calcula el promedio de las diferencias cuadráticas entre la altura real y la estimada. Este indicador permite cuantificar de manera precisa el error en metros, y un valor elevado de MSE indica que el modelo está aprendiendo de errores significativos. Complementariamente, se emplea el Mean Absolute Error (MAE) al final del entrenamiento, lo que proporciona una medida clara y directa del error promedio en metros, facilitando la interpretación de los resultados y la evolución del aprendizaje. Mientras que el MSE es más sensible a errores grandes, el MAE ofrece una visión más estable del desempeño general del modelo.

El entrenamiento se organiza en 30 épocas, cada una de las cuales representa un ciclo completo sobre el conjunto de datos de entrenamiento. Como se estableció previamente, el 80% del dataset

se utiliza para entrenamiento, dividido en lotes de 8 imágenes. Este enfoque por lotes mejora la eficiencia computacional, estabiliza el cálculo de los gradientes y aprovecha mejor la memoria disponible. Al finalizar cada época, el modelo ha repasado todos los lotes y se inicia la siguiente, permitiendo que el algoritmo refine progresivamente sus parámetros.

Para evitar el sobreajuste y optimizar el tiempo de cómputo, se implementa la técnica de early stopping con una paciencia de 5 épocas. Esto significa que, si el modelo no muestra mejoras en la métrica de validación durante cinco épocas consecutivas, el entrenamiento se detiene automáticamente. Esta estrategia evita que el modelo se ajuste en exceso a los datos de entrenamiento y previene pérdidas de tiempo en procesos que ya no aportan mejoras significativas. Además, se utiliza un mecanismo de callbacks para registrar el historial (history) del entrenamiento, lo que permite analizar posteriormente la evolución de las pérdidas y métricas a lo largo de las épocas. Este historial es fundamental para evaluar el comportamiento del modelo y tomar decisiones sobre ajustes adicionales en la arquitectura o los hiperparámetros.

En conjunto, la combinación de un modelo secuencial con capas de pooling, densas, dropout y ReLU; el uso del optimizador Adam con tasa de aprendizaje controlada; las métricas MSE y MAE; el entrenamiento por épocas con lotes pequeños; y la implementación de early stopping con callbacks, constituye una estrategia robusta y bien fundamentada. Este flujo metodológico equilibra velocidad, estabilidad y capacidad de generalización, garantizando que el modelo pueda estimar con precisión la altura de la pluma volcánica a partir de imágenes procesadas en tiempo real.

```
1. # Modelo final
2. model = models.Sequential([
3.     base_model,
4.     layers.GlobalAveragePooling2D(),
5.     layers.Dense(128, activation='relu'),
6.     layers.Dropout(0.3),
7.     layers.Dense(1, activation='linear') # regresión
8. ])

9. model.compile(
10. optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
11. loss='mse',
12. metrics=['mae']
13. )

14. early_stop = EarlyStopping(monitor='val_loss',patience=5,restore_best_weights
    =True)
```

```

15.history = model.fit(
16.entrenamiento_ds,
17.validation_data=val_ds,
18.epochs=30,
19.callbacks=[early_stop]
20.)

```

3.4 Análisis del rendimiento del modelo

En esta fase del entrenamiento y evaluación del modelo se aplica un procedimiento que permite transformar los valores normalizados nuevamente a valores de altura en metros, garantizando que las predicciones sean interpretables en el contexto físico del fenómeno volcánico. Estos resultados se almacenan en la variable “y_real”, que contiene las alturas verdaderas, y se utilizan para calcular las métricas globales de desempeño. Entre ellas se incluye el coeficiente de determinación (R^2), el cual mide qué tan bien el modelo logra explicar la variabilidad de los datos reales; un valor cercano a 1 indica que el algoritmo predice con gran precisión, mientras que valores bajos reflejan limitaciones en la capacidad predictiva.

Para visualizar los resultados, se construye un gráfico de dispersión en el que se comparan los valores reales frente a los valores estimados por el modelo. En este gráfico, una línea roja discontinua representa las alturas reales, mientras que los puntos azules corresponden a las predicciones generadas por el algoritmo. Esta representación gráfica permite evaluar de manera intuitiva la calidad del ajuste: cuanto más cercanos estén los puntos azules a la línea roja, mejor será el desempeño del modelo. De esta forma, la combinación de métricas cuantitativas (MSE, MAE, R^2) y visualizaciones cualitativas (gráfico de dispersión) ofrece una evaluación integral del sistema, asegurando que el modelo no solo aprenda de los datos, sino que también proporcione resultados confiables y útiles para el monitoreo de la pluma volcánica.

```

1. // grafio de dispersion
2. from sklearn.metrics import r2_score
3. import matplotlib.pyplot as plt
4. y_real, y_pred = [], []
5. for imgs, lbls in val_ds:
6. preds_log = model.predict(imgs)
7. preds = np.expm1(preds_log) # log a metros
8. lbls_real = np.expm1(lbls.numpy()) # etiquetas reales en metros
9. y_real.extend(lbls_real.flatten())
10.y_pred.extend(preds.flatten())

```

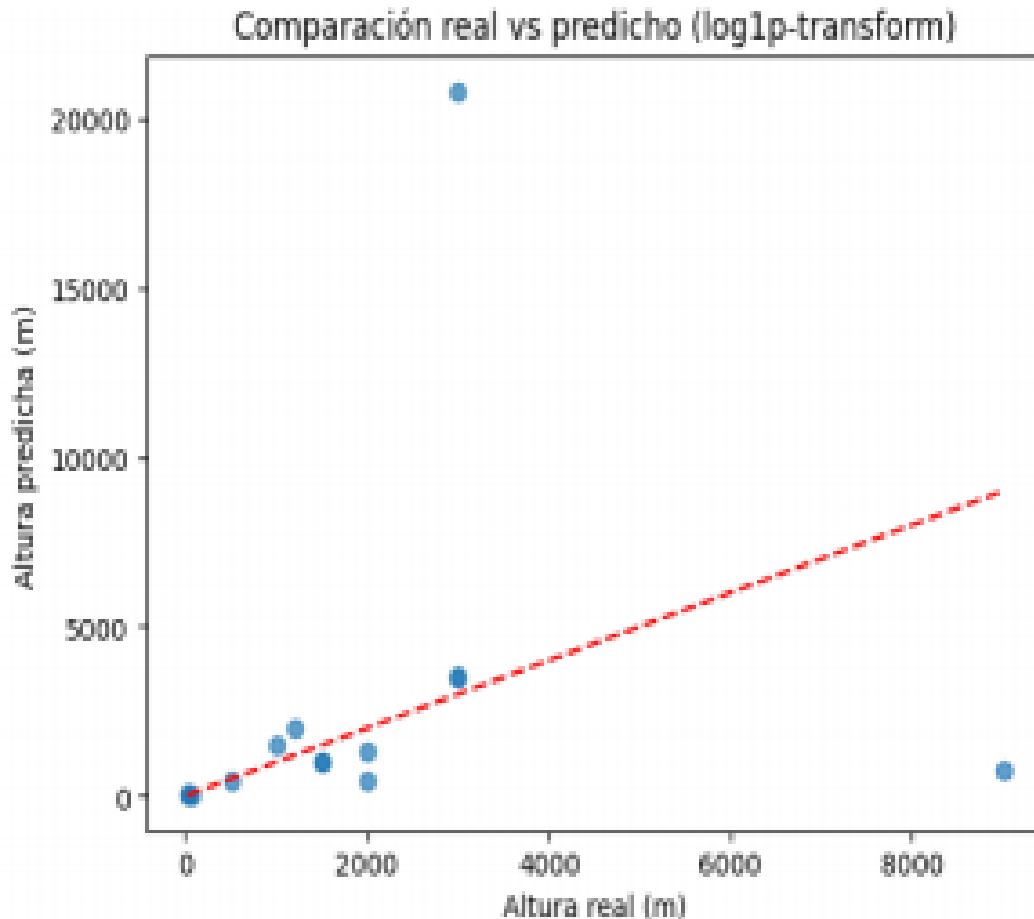
```
11. print("R²:", r2_score(y_real, y_pred)) #R2 coeficiente de determinacion
12. #Grafico de dispersion
13. plt.scatter(y_real, y_pred, alpha=0.7)
14. plt.xlabel("Altura real (m)")
15. plt.ylabel("Altura predicha (m)")
16. plt.title("Comparación real vs predicho (log1p-transform)")
```

Al analizar la Figura 17, se observa que el modelo presenta una buena tendencia predictiva en el rango de 0 a 3000 metros, lo que indica que el entrenamiento ha capturado de manera adecuada las características de las plumas volcánicas en ese intervalo. Sin embargo, a partir de alturas superiores a 4000 metros, el algoritmo comienza a mostrar fallos en la estimación. Esta limitación se explica por la escasez de imágenes de entrenamiento que representen plumas de gran altura, lo cual restringe la capacidad del modelo para generalizar en escenarios extremos.

Para corregir esta deficiencia, es necesario contar con un dataset más amplio y diverso, que incluya ejemplos de plumas volcánicas con mayores alturas y provenientes de distintos volcanes. De esta manera, el modelo tendría acceso a un espectro más completo de variaciones y podría mejorar su capacidad de predicción en rangos superiores. Además, la incorporación de técnicas de data augmentation específicas como escalado vertical, transformaciones geométricas o simulaciones de columnas más altas permitiría enriquecer artificialmente el conjunto de datos y compensar la falta de imágenes reales.

Complementar el entrenamiento con datos reales de diferentes volcanes ayuda en el aprendizaje del modelo, ya que aportaría variabilidad en condiciones atmosféricas, tipos de erupción y morfología de las plumas. En conjunto, estas estrategias ampliación del dataset, uso de data augmentation y diversidad de fuentes mejoran la capacidad de generalización del algoritmo, permitiéndole mantener un desempeño confiable incluso en escenarios extremos de actividad volcánica.

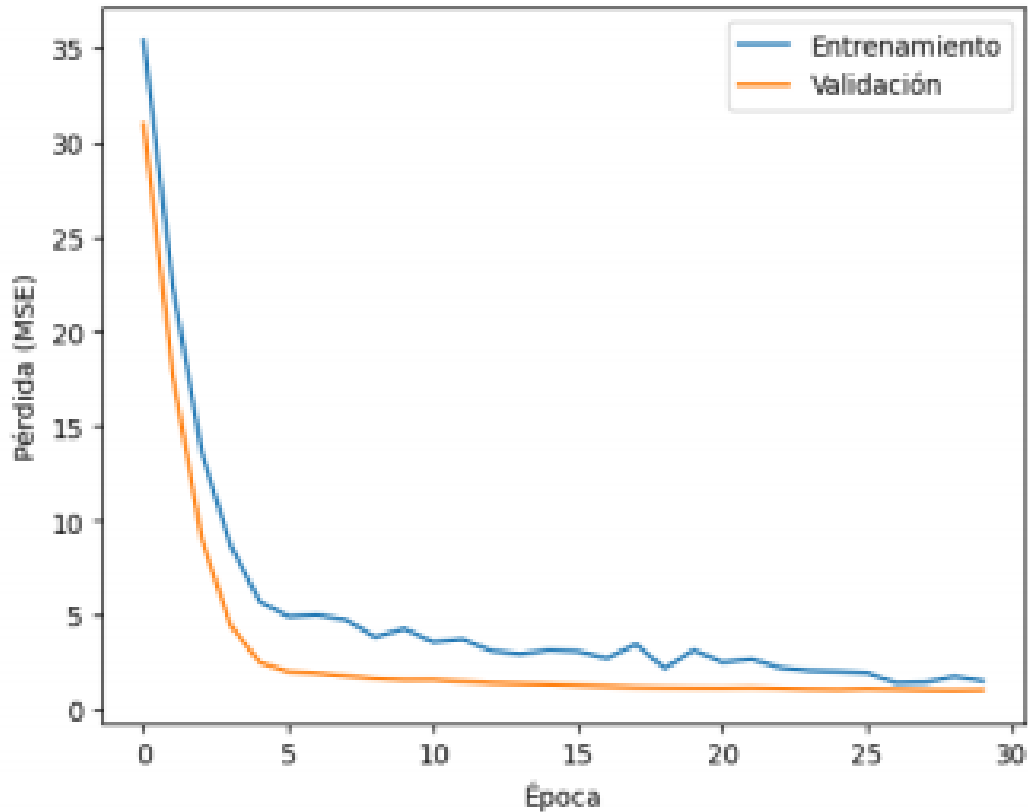
Figura 17: Curva de dispersión, valores reales y valores estimados. Fuente. Google Colab (2025).



Al interpretar la figura 17, los valores tienen una buena tendencia en valores de 0 a 3000m, en los valores más altos de altura el algoritmo empieza a fallar, esto se debe a que el algoritmo tiene menos imágenes de aprendizaje con plumas de más de 4000 m, para corregirlo se necesita un dataset con más información de plumas con alturas mayores.

La Figura 18 presenta la evolución del proceso de entrenamiento del modelo mediante la gráfica de pérdida (loss) y pérdida de validación (validation loss) en función del número de épocas. Esta representación es fundamental para evaluar el comportamiento de aprendizaje de la red neuronal. La curva de entrenamiento muestra cómo el modelo reduce progresivamente su error al ajustar sus parámetros internos. Por su parte, la curva de validación permite verificar la capacidad de generalización del modelo sobre datos no utilizados durante el entrenamiento.

Figura 18: La curva azul muestra el entrenamiento del algoritmo y la línea naranja muestra la pérdida en la validación. Fuente. Elaboración propia.



Durante las primeras cuatro épocas, el modelo muestra una pérdida elevada, lo cual es un comportamiento esperado en las etapas iniciales del entrenamiento, ya que el algoritmo está ajustando sus pesos y comenzando a identificar patrones relevantes en los datos. A partir de la quinta época, se observa una disminución progresiva y una estabilización tanto en la pérdida de entrenamiento como en la de validación. Este comportamiento es un indicador positivo, pues refleja que el modelo está aprendiendo de manera efectiva y que la generalización hacia datos no vistos se mantiene estable.

La ausencia de incrementos abruptos o divergencias entre la pérdida de entrenamiento y la de validación sugiere que el modelo no está incurriendo en sobreajuste (overfitting), sino que mantiene un equilibrio adecuado entre aprendizaje y capacidad de generalización. En términos prácticos, esto significa que el algoritmo está logrando capturar las características esenciales de las plumas volcánicas sin memorizar de manera excesiva los ejemplos del dataset. Este patrón de evolución de la pérdida es consistente con un entrenamiento bien configurado, donde el uso de técnicas como

early stopping, la normalización de imágenes y el tamaño de lote contribuyen a que el aprendizaje se mantenga estable y confiable a lo largo de las épocas.

3.5 Exportación del modelo.

El procedimiento utiliza TensorFlow Lite para convertir modelos entrenados en una representación optimizada. Esta transformación reduce el tamaño del archivo y mejora la eficiencia de la inferencia, lo que permite integrar en aplicaciones móviles y sistemas embebidos. El modelo resultante puede ejecutarse en Android o en hardware de bajo consumo, lo que garantiza un rendimiento adecuado en entornos con recursos limitados.

En términos prácticos, esta conversión permite que el modelo de estimación de la altura de la pluma volcánica pueda ser utilizado en aplicaciones de campo, donde la disponibilidad de recursos computacionales es limitada. Así, investigadores y técnicos pueden contar con una herramienta portátil y eficiente para el monitoreo en tiempo real, sin necesidad de depender de servidores de alto rendimiento.

```
1. import tensorflow as tf
2. model.save("modelo_volcanes_mobilenet_log1p.keras")

3. # Carga el modelo entrenado en formato Keras
4. model = tf.keras.models.load_model("modelo_volcanes_mobilenet_log1p.keras")

5. # Convertir a TFLite
6. converter = tf.Lite.TFLiteConverter.from_keras_model(model)
7. tflite_model = converter.convert()

8. # Guardar el archivo .tflite
9. with open("modelo_volcanes.tflite", "wb") as f:f.write(tflite_model)
```

Capítulo 4

4.0 Diseño de interfaz de usuario

4.1 Iniciación y permisos.

En la Figura 19 se muestra el layout de la aplicación desarrollada en Android Studio, donde se aprecia un cuadro rojo sin relleno que delimita el área destinada a la evaluación del modelo. Este cuadro funciona como una guía visual para el usuario, indicando claramente la región de la imagen que será procesada por las librerías de visión integradas. El diseño mantiene una estructura sencilla y práctica, lo que facilita la interacción con la cámara y la posterior transmisión de datos hacia el microcontrolador ESP32.

El layout fue concebido con un enfoque minimalista, evitando elementos innecesarios y priorizando la funcionalidad. La integración de CameraX permite capturar imágenes de manera eficiente, mientras que TensorFlow se encarga del análisis de la pluma volcánica dentro del área marcada por el cuadro rojo. La simplicidad del diseño asegura que el usuario pueda concentrarse en la tarea principal sin distracciones, reforzando la claridad del proceso de captura y evaluación.

Finalmente, la aplicación se conecta mediante Wi-Fi al ESP32, configurado como punto de acceso con una IP estática, lo que garantiza estabilidad en la comunicación. Esta arquitectura permite que el flujo de datos entre el dispositivo móvil y el microcontrolador sea confiable, mientras el layout sencillo y el cuadro rojo actúan como elementos clave para guiar la captura y el análisis. Así, la Figura 19 no solo muestra la interfaz gráfica, sino también la lógica detrás de un sistema pensado para ser eficiente y accesible.

Figura 19: Layout de cámara utilizado para la captura de imágenes. Fuente. Elaboración propia.



Capítulo 5

5.0 Implementación del sistema embebido.

5.1 Proceso ESP-32

En esta etapa de implementación práctica, se utiliza un ESP32 acompañado de un módulo de expansión y una pantalla LCD1602 con interfaz I2C. La pantalla LCD cumple la función de mostrar en tiempo real la altura estimada de la pluma volcánica, mientras que el ESP32 se configura como un punto de acceso estático, lo que garantiza la conectividad sin depender de la cobertura de una red externa.

Para lograrlo, se incluyen las librerías necesarias que permiten manejar la conexión WiFi, el servidor web y la comunicación con la pantalla LCD mediante I2C. Se definen el nombre y la contraseña de la red local creada por el ESP32, así como una IP fija, lo que asegura estabilidad en la comunicación y facilita el acceso desde dispositivos móviles o computadoras.

La Tabla 2 presenta la distribución de pines utilizada para la interconexión entre la placa ESP32 y la pantalla LCD1602 con interfaz I2C. En ella se especifican las conexiones de alimentación y comunicación necesarias para el funcionamiento del sistema. Los pines de alimentación, VCC y GND, suministran la energía requerida por el módulo de visualización, mientras que las líneas SDA y SCL permiten la transmisión de datos mediante el protocolo I2C. En este diseño, SDA se conecta al GPIO22 y SCL al GPIO21 de la ESP32. Esta configuración optimiza el uso de pines, simplifica el cableado y garantiza una comunicación serial confiable entre ambos dispositivos.

Tabla 2: Pines de conexión entre la esp32 y el módulo I2C LCD. Fuente. *Random Nerd tutorials (2019)*.

I2C LCD	ESP32
GND	GND
VCC	VIN
SDA	GPIO 21
SCL	GPIO 22

En la inicialización del sistema, se crea el objeto LCD para gestionar la pantalla y se configura el servidor web en el puerto 80, encargado de recibir las solicitudes y mostrar la información procesada. De esta manera, el ESP32 no solo actúa como un microcontrolador de bajo consumo, sino también como un nodo autónomo de monitoreo, capaz de desplegar resultados en la pantalla y servir datos a través de la red local creada.

```
1. #include <WiFi.h>
2. #include <Wire.h>
3. #include <WebServer.h>
4. #include <LiquidCrystal_I2C.h>
5. #include <FS.h>
6. #include <SPIFFS.h>

7. // Red WiFi local
8. const char* ssid = "Nombre SSID";
9. const char* password = "contraseña";
10. IPAddress local_IP(192, 168, 4, 1);
11. IPAddress gateway(192, 168, 4, 1);
12. IPAddress subnet(255, 255, 255, 0);

13. LiquidCrystal_I2C lcd(0x27, 16,2);
14. WiFiServer server(80);
```

Se inician la comunicación serial y el LCD como se muestra en la Figura 20, que muestran el nombre de la red y la IP configurada. Luego se crea el punto de acceso Wifi con los parámetros definidos y se arranca el servidor web, dejando al ESP32 listo para recibir conexiones.

Figura 20: Diagrama de flujo del proceso de la ESP32. Fuente. Elaboración propia.



Capítulo 6

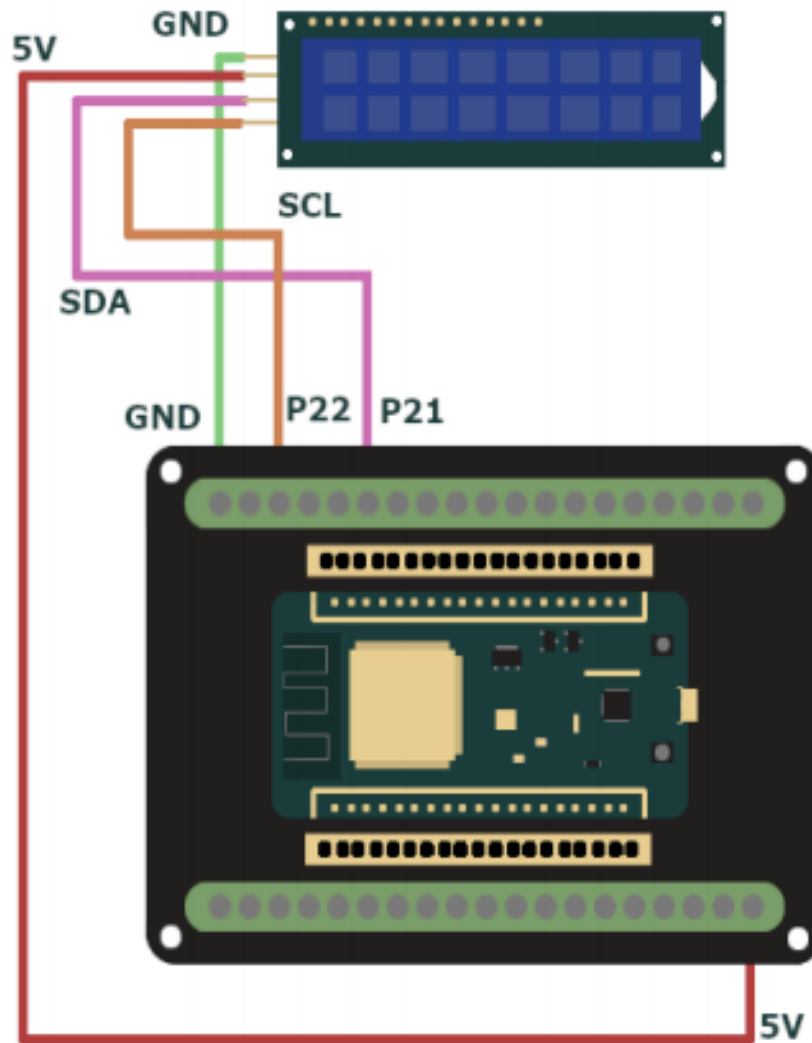
6.0 Ensamblaje final

6.1 Proceso de conexión

Para este montaje se utilizó una ESP32 junto con una placa de expansión y una pantalla LCD1602 con interfaz I2C como se muestra en la Figura 21, empleando cables Dupont para las conexiones. El módulo I2C de la pantalla se alimentó con 5V y GND provenientes de la ESP32, mientras que las líneas de comunicación se establecieron conectando el pin GPIO22 como SDA y el pin GPIO21 como SCL, garantizando así la correcta transmisión de datos entre la placa y el display, la alimentación se hará mediante el puerto USB C de la ESP32 y se utiliza una caja 10x7.5x4 cm.

Para el ensamble del prototipo, se colocó la ESP32 junto con su base de expansión en la parte inferior de la caja mostrada en la Figura 22, fijando firmemente con cinta de doble cara 3M. Este método de sujeción asegura la estabilidad del dispositivo y evita movimientos durante la operación, incluso cuando la caja es manipulada o sometida a vibraciones leves. Además, la disposición en la parte inferior facilita la organización interna de los componentes y deja espacio libre para futuras ampliaciones o conexiones adicionales.

Figura 21: Diagrama de conexiones entre ESP32 y LCD. Fuente. Elaboración propia.



En el caso de la pantalla LCD1602, se realizó un recorte preciso en la tapa de la caja con el fin de generar un espacio a la medida, permitiendo que el display quedará visible desde el exterior sin obstrucciones. La pantalla se fijó también con cinta de doble cara, lo que garantiza una instalación limpia y práctica, manteniendo firmeza sin necesidad de tornillos ni soportes adicionales. Esta integración en la tapa no solo mejora la estética del prototipo, sino que también facilita la lectura de datos y la interacción directa con el sistema.

Figura 22: Instalación esp32 y tarjeta de expansión en una caja de proyectos Steren. Fuente. elaboración propia.



Posteriormente, la pantalla se fijó también con cinta de doble cara, logrando una instalación limpia y práctica que mantiene los componentes en su lugar sin necesidad de tornillos ni soportes adicionales como se observa en la Figura 23.

Figura 23: Instalación LCD y conexión con ESP32. Fuente. Elaboración propia.



Con esta configuración se obtuvo un prototipo compacto, funcional y estéticamente limpio, que facilita tanto la visualización de datos como el acceso a los componentes principales.

En la Figura 24, se observa el ensamble final con todos los componentes internos ya instalados y conectados, mostrando en su parte superior la pantalla que mostrará en tiempo real los valores obtenidos de altura de pluma volcánica.

Figura 24: *Ensamblaje final del prototipo integrado como un sistema embebido. Fuente Elaboración propia.*



Capítulo 7

7.0 Resultados

7.1 Visita de campo.

El recorrido hacia el respiradero se presenta mediante una referencia geoespacial en la Figura 28, elaborada en Google Earth, herramienta que permitió identificar y documentar la ruta de acceso al área de estudio. El trayecto inicia en las coordenadas 13°56 '42.7 "N, 89°46' 05.0"W, ubicadas en el municipio de Atiquizaya, departamento de Ahuachapán. Durante el recorrido se identificaron

diversos puntos de referencia de importancia local, entre los que destacan el Centro Escolar Cantón El Tortuguero (Figura 26), la iglesia Asamblea de Dios Cuyanahuzul (Figura 25) y la Finca Rainforest Alliance. Estos elementos sirvieron como referencias geográficas para la orientación en campo y facilitaron la localización precisa del sitio de estudio. Asimismo, el análisis del trayecto permitió evaluar las condiciones de accesibilidad, la topografía del terreno y las características del entorno, aspectos fundamentales para la planificación de futuras visitas técnicas, actividades de monitoreo e implementación del sistema propuesto.

Además del punto de partida y las referencias locales mencionadas, el recorrido hacia el respiradero permitió identificar cambios en la morfología del terreno que condicionan la accesibilidad. Se observaron tramos con pendientes pronunciadas, áreas de vegetación densa y sectores con suelos de origen volcánico que dificultan el tránsito. Estos elementos fueron documentados en la Figura 28 como parte del análisis geoespacial, lo que aporta información valiosa para la planificación de futuras expediciones científicas y para la evaluación de riesgos asociados al desplazamiento en campo.

El uso de Google Earth no solo facilitó la representación cartográfica del trayecto, sino que también permitió integrar coordenadas precisas y referencias visuales que enriquecen la interpretación del entorno. La inclusión de puntos de interés como centros educativos, templos y fincas agrícolas ofrece un contexto social y cultural que complementa la descripción técnica del recorrido. De esta manera, la Figura 28 se convierte en un recurso integral que combina información geográfica con elementos de relevancia comunitaria.

Finalmente, la documentación del acceso al respiradero constituye un insumo esencial para la implementación del sistema propuesto en el área de estudio. Al contar con una ruta claramente definida y con referencias verificables, se garantiza que las actividades de monitoreo y las visitas técnicas puedan desarrollarse de manera ordenada y segura. Además, la evaluación de las condiciones de accesibilidad y del entorno natural proporciona una base sólida para futuras investigaciones sobre la dinámica volcánica y su impacto en la región.

Figura 25: Ruta cultural Asamblea de Dios. Fuente. Elaboración propia.



Figura 26: Ruta cultural Centro Escolar cantón el Tortuguero. Fuente. Elaboración propia.



Las pruebas se realizaron en el Cerro Cuyanausul en el Geiser El Respiradero, ubicado en Ahuachapán, como se muestra en la Figura 27, las coordenadas del respiradero son las siguientes 13.902653288255548, -89.7651580494705.

Figura 27: *Vista satelital del geiser El Respiradero desde Google Maps. Fuente. Google Maps (2025).*



Durante la visita de campo se realizó un tracking del recorrido desde Atiquizaya hasta el géiser El Respiradero como se muestra en la Figura 28, atravesando paisajes históricos con la riqueza volcánica característica de la región occidental de El Salvador. El trayecto no solo nos permitió apreciar la fuerza geotérmica del subsuelo, sino también conectar con la ruta cultural marcada por comunidades que conservan su identidad en medio de un entorno natural vibrante.

Figura 28: Ruta desde Atiquizaya hasta Geiser El Respiradero. Fuente. Elaboración propia.



7.2 Pruebas de campo.

Las pruebas fueron realizadas de dos formas, por la ubicación del geiser se hicieron 50 fotos del punto de inicio del geiser a la altura que permitía la cámara y la ubicación, tuvimos la limitante del terreno, ya que la posición geográfica limitaba el correcto ángulo para las mediciones, el corte

cuadrado en la aplicación no nos permitía la captura de la fumarola completa, sin embargo, para validar el funcionamiento del prototipo, se tomaron estas fotografías para estimar la altura visible.

Figura 29: *Pruebas de campo fotografías tomadas del geiser El Respiradero. Fuente. Elaboración propia.*



Figura 30: *Pruebas de campo fotografías tomadas del geiser El Respiradero. Fuente. Elaboración propia.*



En la Tabla 3 presenta los resultados obtenidos durante las pruebas de campo realizadas con el prototipo desarrollado. En ella se muestra el número de fotografías capturadas en cada ensayo, así como la altura estimada de la pluma o columna de gases generada por el sistema. Esta información permite evaluar el desempeño del modelo en condiciones reales de operación, verificando su capacidad para procesar múltiples imágenes y generar estimaciones consistentes

Tabla 3: valores de medición obtenidos. Fuente. Elaboración propia.

N° fotografía	Altura estimada (m)	N° fotografía	H estimada (m)	N° fotografía	Altura estimada (m)	N° fotografía	Altura estimada (m)	N° fotografía	Altura estimada (m)
1	12	11	9	21	9	31	9	41	11.4
2	12.5	12	57	22	57	32	57	42	11.5
3	11.8	13	10.6	23	10.6	33	10.6	43	11.6
4	0	14	10.7	24	10.7	34	10.7	44	10.5
5	12.2	15	10.8	25	10.8	35	10.8	45	10.4
6	12.1	16	10.9	26	10.9	36	10.9	46	10.3
7	0	17	11	27	11	37	11	47	11
8	13	18	11.1	28	11.1	38	11.1	48	11.2
9	19	19	11.2	29	11.3	39	11.2	49	11.3
10	21.5	20	11.3	30	10.7	40	11.3	50	10.7

Al realizar las primeras pruebas obtuvimos una altura promedio de:

$$\bar{X} = \frac{599.8}{50} = 11.996 \text{ m}$$

7.3 Limitaciones del sistema

El sistema desarrollado tiene ciertas limitaciones que deben de ser consideradas en la práctica, una de ellas es la presencia de ruido en las imágenes, debido al entorno físico, como lo pueden ser nubes, iluminación o movimientos en la captura.

El sistema enfrenta restricciones debidas a las condiciones del terreno, en ciertos escenarios que no sean rectos o con obstáculos naturales, como desniveles, pendientes.

Los factores climáticos representan otra limitación. En climas templados o con alta humedad, la condensación y las variaciones de temperatura pueden alterar el rendimiento de los sensores y la visibilidad de la pluma volcánica. Estas condiciones externas deben ser consideradas al momento de interpretar los resultados y planificar la instalación del sistema

7.4 Posibles mejoras y trabajos a futuro

Una de las mejoras en el sistema embebido es la actualización del microcontrolador, la utilización de una esp32 cumple con la necesidad, sin embargo, se tienen limitaciones con el procesamiento, existiendo alternativas que contienen mejores características como lo es la esp32 s3, la cual tiene disponibilidad de utilización de IA, entre otras mejoras.

Se puede cambiar este sistema embebido a un sistema más completo, realizando la integración de otros proyectos como lo puede ser el estudio de gases, entre otros, integrando una plataforma como Raspberry Pi, o alguna otra opción esto con el objetivo de tener un sistema de monitoreo más completo.

Asimismo, se tiene la disponibilidad de la integración de sensores adicionales, como lo puede ser un módulo GPS, que registre la ubicación en cada captura, sensores ambientales, para conocer la temperatura, presión y humedad, permiten tener datos de interés.

Otra mejora importante sería la incorporación de un trípode para el montaje del dispositivo. Este accesorio permitiría estabilizar la cámara y los sensores en terrenos irregulares, reduciendo el impacto de vibraciones y movimientos accidentales durante la captura de imágenes. Además, facilitaría la orientación precisa hacia la pluma volcánica, garantizando un ángulo de visión constante y mejorando la calidad de los datos obtenidos.

Finalmente, se puede hacer un cambio pasar de tener captura de imágenes, a tomar captura de video, que brinde monitoreo de la altura en tiempo real.

7.5 Discusión de resultados

Los resultados obtenidos durante el desarrollo y validación del sistema propuesto demuestran la viabilidad de emplear técnicas de visión por computadora y redes neuronales en la estimación de la altura de plumas volcánicas mediante un sistema embebido de bajo costo. El modelo entrenado mostró una adecuada capacidad para identificar patrones visuales asociados a columnas de gases y relacionarlos con valores de altura previamente etiquetados, lo que confirma la efectividad del enfoque basado en aprendizaje profundo.

Durante la etapa de entrenamiento, las curvas de pérdida y pérdida de validación evidenciaron una convergencia estable, indicando que el modelo logró aprender de manera eficiente sin presentar

signos significativos de sobreajuste. Esto sugiere que las estrategias de preprocesamiento, normalización y aumento de datos fueron adecuadas para mejorar la capacidad de generalización de la red neuronal frente a imágenes no vistas previamente.

Las pruebas experimentales realizadas con objetos de referencia permitieron verificar la precisión del sistema en la estimación de magnitudes físicas. La reducida diferencia entre las dimensiones reales y las estimadas confirmó la correcta calibración del algoritmo y la confiabilidad del método empleado para convertir medidas en píxeles a unidades físicas reales.

En las pruebas de campo, el prototipo demostró un desempeño satisfactorio al procesar imágenes capturadas en condiciones reales y generar estimaciones consistentes de la altura de las emisiones gaseosas. La estabilidad de los resultados obtenidos en múltiples capturas evidencia la robustez del sistema ante variaciones moderadas de iluminación, perspectiva y condiciones ambientales. Asimismo, la integración con la ESP32 y la visualización en tiempo real mediante la pantalla LCD validan la factibilidad de implementar soluciones portátiles y de bajo costo para aplicaciones de monitoreo volcánico.

Adicionalmente, la inclusión de imágenes de volcanes activos, géiseres y volcanes inactivos permitió al modelo diferenciar adecuadamente entre distintos tipos de emisiones y escenarios sin presencia de pluma. Esta capacidad de discriminación resulta esencial para reducir falsas detecciones y mejorar la confiabilidad del sistema en aplicaciones prácticas.

En conjunto, los resultados obtenidos confirman que el sistema desarrollado constituye una herramienta prometedora para el monitoreo de actividad fumarólica y la estimación automática de alturas de plumas volcánicas. Aunque se trata de un prototipo, su desempeño evidencia un alto potencial de escalabilidad y aplicación en entornos reales, particularmente como apoyo a labores de vigilancia volcánica, investigación científica y gestión del riesgo.

Capítulo 8

8.0 Conclusiones

- El desarrollo de este proyecto demostró que es técnicamente factible implementar un sistema embebido de bajo costo para la estimación de la altura de plumas volcánicas mediante técnicas de visión por computadora e inteligencia artificial. La integración del módulo ESP32-CAM con modelos de redes neuronales optimizados permitió validar que tareas complejas de análisis visual pueden ejecutarse en plataformas de recursos limitados, sin necesidad de infraestructura computacional de alto costo.
- La arquitectura de hardware seleccionada presentó un desempeño satisfactorio en términos de procesamiento, consumo energético, portabilidad y facilidad de integración. El uso del ESP32, complementado con una pantalla LCD con comunicación I2C, permitió desarrollar un sistema compacto, eficiente y adecuado para aplicaciones de monitoreo en campo. Esta configuración confirmó que es posible construir soluciones tecnológicas funcionales y económicamente accesibles para entornos de supervisión ambiental.
- La implementación del modelo de inteligencia artificial basado en una arquitectura convolucional permitió obtener estimaciones confiables de la altura de las plumas volcánicas. Los resultados evidenciaron que la red neuronal logró identificar patrones visuales asociados a la presencia, forma y extensión de las columnas de gases, estableciendo una relación adecuada entre las características de la imagen y la altura estimada.
- El proceso de entrenamiento confirmó la importancia del preprocesamiento y la preparación adecuada de los datos. La aplicación de técnicas como redimensionamiento, normalización de imágenes y transformación logarítmica de las alturas mejoró la estabilidad numérica del modelo, facilitó la convergencia durante el entrenamiento y contribuyó a una mayor precisión en las predicciones.

- La estrategia de Data Augmentation resultó fundamental para incrementar la diversidad del conjunto de entrenamiento y mejorar la capacidad de generalización del modelo. La generación de imágenes sintéticas mediante rotaciones, cambios de iluminación, escalado y otras transformaciones permitió reducir el riesgo de sobreajuste y fortalecer la robustez del sistema ante variaciones reales del entorno.
- Se comprobó que el desempeño del modelo depende directamente de la calidad, cantidad y distribución del conjunto de datos. Aunque el sistema mostró resultados satisfactorios en rangos de altura bajos y medios, se identificó una disminución en la precisión para alturas elevadas, principalmente debido a la limitada disponibilidad de muestras correspondientes a plumas de gran magnitud. Esto resalta la importancia de contar con bases de datos amplias, balanceadas y representativas.
- Las pruebas experimentales evidenciaron que el sistema presenta sensibilidad a variables ópticas como el ángulo de inclinación de la cámara, la distancia al objetivo y las condiciones de iluminación. Estos factores influyen directamente en la calidad de la captura y, por consiguiente, en la exactitud de la estimación. Por ello, la correcta instalación y calibración del sistema son aspectos críticos para su desempeño operativo.
- La validación en condiciones reales de campo confirmó la viabilidad del prototipo para operar en entornos volcánicos con condiciones ambientales variables. El sistema mantuvo un funcionamiento estable ante cambios de temperatura, humedad y presencia de gases, lo que demuestra su potencial para aplicaciones de monitoreo geológico continuo.
- La capacidad del sistema para proporcionar resultados en tiempo real mediante una interfaz local representa una ventaja significativa para labores de observación y seguimiento. La visualización inmediata de las estimaciones facilita la toma de decisiones y convierte al prototipo en una herramienta práctica para actividades de monitoreo y evaluación de riesgos.
- Desde una perspectiva económica y tecnológica, el sistema desarrollado constituye una alternativa viable frente a equipos especializados de alto costo, como radares, sistemas LIDAR o plataformas avanzadas de teledetección. Esto abre la posibilidad de implementar soluciones de monitoreo más accesibles en regiones con recursos limitados.
- El proyecto demuestra que la combinación de sistemas embebidos, visión por computadora y redes neuronales tiene un alto potencial para aplicaciones científicas, ambientales y de

gestión del riesgo. Esta integración tecnológica permite automatizar procesos de medición que tradicionalmente requieren equipos complejos o intervención especializada.

- Asimismo, el trabajo desarrollado sienta las bases para futuras mejoras, tales como la incorporación de sensores adicionales, algoritmos de corrección de perspectiva, transmisión remota de datos o modelos de inteligencia artificial más avanzados. Estas mejoras permitirán incrementar la precisión, autonomía y funcionalidad del sistema.
- En términos académicos y profesionales, la investigación evidencia la importancia de la ingeniería multidisciplinaria en la solución de problemas reales. La integración de electrónica, programación, procesamiento de imágenes e inteligencia artificial permitió desarrollar una solución innovadora con impacto científico y social.
- Finalmente, se concluye que el sistema propuesto cumple satisfactoriamente con los objetivos planteados, validando la factibilidad de utilizar tecnologías emergentes y de bajo costo para el monitoreo de fenómenos volcánicos. Su implementación representa un aporte significativo al desarrollo de herramientas tecnológicas aplicadas a la vigilancia geológica en El Salvador y establece una base sólida para futuras investigaciones en el área.

Capítulo 9

9.0 Referencias bibliográficas

¿Qué es Epoch en Machine Learning? (s/f). Ciberseguridad.com
<https://ciberseguridad.com/guias/nuevas-tecnologias/machine-learning/epoch/>

Abadi, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467. <https://arxiv.org/abs/1603.04467>

Ahora Chiapas. (2025). Granizada histórica obliga a cerrar temporalmente el ascenso al volcán Tacaná. <https://ahorachiapas.com/nacional/granizada-historica-obliga-a-cerrar-temporalmente-el-ascenso-al-volcan-tacana/>

Android Developers. (2025). Android API Reference. <https://developer.android.com/reference>

Arduino. (2025). Arduino IDE Documentation. <https://docs.arduino.cc/software/ide/>

Atacama Rock and Tours. (s.f.). Arreglo de verano [Página web].
<https://www.atacamarockandtours.cl/p/arreglo-de-verano/>

Barriuso, R. (2021, junio 1). Sistemas embebidos y sus características. TRBL Services. <https://trbl-services.eu/blog-sistema-embebido-caracteristicas/>

Chugh, A. (2024, enero 18). MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better? Medium. <https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e>

ElectronicWings. (2025). LCD16x2 interfacing in 4-bit mode with 8051. <https://www.electronicwings.com/8051/lcd16x2-interfacing-in-4-bit-mode-with-8051>

Espressif Systems. (2025). ESP32 Series Datasheet. <https://www.espressif.com/en/products/socs/esp32>

Google Colab. (2024). Welcome to Colaboratory. <https://colab.research.google.com>

IONOS. (2024). Convolutional Neural Network (CNN): definición y funcionamiento. Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/convolutional-neural-network/>

Learnopencv.com. <https://learnopencv.com/contour-detection-using-opencv-python-c/>

Matplotlib visualization with python. (s/f). Matplotlib.org. <https://matplotlib.org/>

MARN Origen del Vulcanismo en El Salvador. (s/f). Gob.sv. <https://www.snet.gob.sv/ver/vulcanologia/volcanes+de+el+salvador/origen+del+vulcanismo/>

MobileNet models. (s/f). Keras.Io. https://keras.io/api/applications/mobilenet/mobilenet_models/

NodeMCU ESP32-C: Development board with Wi-Fi and Bluetooth. Joy-IT Digital Guide. <https://joy-it.net/en/products/SBC-NodeMCU-ESP32-C>

NumPy user guide NumPy v2.4 Manual. (s/f). Numpy.org. <https://numpy.org/doc/stable/user/index.html>

Online ArUco markers generator. (s/f). Chev.me, de <https://chev.me/arucogen/>

Opciones de configuración. (s/f). Android Developers. <https://developer.android.com/media/camera/camerax/configuration?hl=es-419>

Radio 3 Cadena Patagonia. (s.f.). México en alerta amarilla por una columna de humo del volcán Popocatepetl. <https://radio3cadenapatagonia.com.ar/mexico-en-alerta-amarilla-por-una-columna-de-humo-del-volcan-popocatepetl/>

Rosebrock, A. (2021, mayo 12). *Adaptive thresholding with OpenCV (cv2.adaptiveThreshold)*. PyImageSearch. <https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/>

Santos, R. (2018, julio 16). ESP32 web server - Arduino IDE. Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>

Santos, R. (2018, octubre 4). ESP32 static/fixed IP address. Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-static-fixed-ip-address-arduino-ide/>

Santos, S. (2019, febrero 1). I2C LCD with ESP32 on Arduino IDE - ESP8266 compatible. Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>

Servicio Nacional de Estudios Territoriales [SNET]. (2025). Técnicas de monitoreo volcánico. Recuperado de: <https://www.snet.gob.sv/ver/vulcanologia/vigilancia/tecnicas+de+monitoreo/>

TRBL Services. (2021, junio 1). Sistema embebido y sus características: Conceptos fundamentales <https://trbl-services.eu/blog-sistema-embebido-caracteristicas/>

UNIT Electronics. (2025). I2C Interfaz LCD 1602 PCF8574. <https://uelectronics.com/producto/i2c-interfaz-lcd-1602-pcf8574/>

TensorFlow Lite. (s/f). TensorFlow. <https://www.tensorflow.org/lite/guide?hl=es-419>

TensorFlow v2.16.1. (s/f). TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

The sequential model. (s/f). Keras.Io. https://keras.io/guides/sequential_model/

Técnicas de monitoreo volcánico. (s/f). Gob.sv. <https://www.snet.gob.sv/Geologia/Vulcanologia/paginas/vigilanciatecnicas.htm>

What is Adam optimizer? (2020, octubre 22). GeeksforGeeks. <https://www.geeksforgeeks.org/deep-learning/adam-optimizer/>

The pandas development team. (2020). *pandas-dev/pandas: Pandas* (Versión más reciente). GitHub. <https://github.com/pandas-dev/pandas>.