

Red neuronal artificial para determinar la condición ecofisiológica de los helechos espada a partir de una imagen

Artificial neural network to determine the ecophysiological condition of sword ferns from an image

U. López Hernández^{2*}, E.M. Silva Cruz³, L.F. Barcelos Mendoza⁴

**Autor de correspondencia*

DOI: 10.56643/rcia.v4i2.224

Como citar este artículo: López Hernández et al. (2025). Red neuronal artificial para determinar la condición ecofisiológica de los helechos espada a partir de una imagen. *Revista Científica de Ingenierías y Arquitectura*. 4(2). 45-73. DOI: <https://doi.org/10.56643/rcia.v4i2.224>

RESUMEN

El presente trabajo se enfoca en analizar diversas técnicas de inteligencia artificial, como mnist, el aumento de datos, dropOut y, principalmente, la red neuronal convolucional (cnn) aplicada al campo de la ecofisiología en helechos. Esta investigación parte de reconocer la necesidad de ocupar herramientas tecnológicas que ayuden a identificar de forma automática el estado de salud de los helechos espada, debido a los cambios climáticos que han tenido lugar en los últimos años, pues han afectado a las especies endémicas de Oaxaca. Se identificó que el análisis de imágenes puede ser clave para detectar signos de estrés o deterioro de la vegetación.

En este artículo se presenta una metodología para el modelado de una red neuronal empleando Python en el entorno de trabajo de Google Colab y utilizando diferentes características para el cálculo de parámetros de las capas convolucionales.

2 Estudiante de Ingeniería de Software y Sistemas Computacionales. Grupo GIDi de aplicaciones STEAM, Universidad La Salle Oaxaca. ORCID: <https://orcid.org/0009-0004-3628-6079>. Correo de correspondencia: 014430576@ulsaoaxaca.edu.mx

3 Doctor en Electrónica. Docente en Grupo GIDi de aplicaciones STEAM, Universidad La Salle Oaxaca. ORCID: <https://orcid.org/0000-0002-0496-9682>. Correo electrónico: eric.silva@ulsaoaxaca.edu.mx

4 Maestro en Educación. Docente en Grupo GIDi de aplicaciones STEAM, Universidad La Salle Oaxaca. ORCID: <https://orcid.org/0000-0003-0285-650X>. Correo electrónico: ludwig.barcelosme@ulsaoaxaca.edu.mx
Recibido: 23/07/2025 | Aceptado: 09/09/2025 | Publicado: 15/12/2025

Los autores declaran no tener ningún conflicto de interés. .

Ello hace necesaria la implementación de filtros y estructuras propias del aprendizaje profundo, como también explicar cada etapa de la configuración. Los modelos diseñados fueron entrenados mediante un set de datos (*dataset*) de imágenes de helechos con distintos estados de salud, durante la época de estiaje.

Entre los resultados más relevantes se destaca la correcta construcción de una arquitectura de red neuronal, así como la adecuada compilación y ejecución de modelos, lo que permitió alcanzar una precisión de 96% y hace posible replicar y escalar el objetivo de este proyecto. Una de las aportaciones principales de la presente investigación es la integración de la inteligencia artificial como herramienta de apoyo para la realización de estudios ecofisiológicos, lo que puede mejorar el monitoreo del estado de las plantas endémicas de Oaxaca, por ejemplo, el helecho tipo espada, de forma automatizada.

PALABRAS CLAVE: Capas, Ecofisiología, Helecho espada, Inteligencia artificial, Modelos, Red neuronal.

ABSTRACT

This paper focuses on the analysis of various artificial intelligence techniques, such as MNIST, data augmentation, dropOut, and primarily the convolutional neural network (CNN), applied to the field of fern ecophysiology. This research stems from the recognition of the need for technological tools that help automatically identify the health status of sword ferns, especially due to the climate changes that have occurred in recent years and are affecting the endemic species of Oaxaca. It was identified that image analysis can be key to detecting signs of stress or vegetation deterioration.

A methodology is presented for modeling a neural network using Python in the Google Colab framework, applying different features to calculate the parameters of the convolutional layers. The implementation of filters and structures specific to deep learning was necessary, explaining each stage of the configuration. The models were trained using a *dataset* of images taken of ferns in different health states during the dry season.

Among the most significant results, the successful construction of a neural network architecture stands out, as well as the successful compilation and execution of models, achieving an accuracy of 96%, which allows for the replication and scaling of this project's objective. One of the main contributions of this research is the integration of artificial intelligence as a support tool for

ecophysiological studies, improving the automated monitoring of the status of endemic plants in Oaxaca, such as the Sword Fern.

KEYWORDS: Artificial Intelligence, Ecophysiology, Layers, Models, Neural Network, Sword Fern.

INTRODUCCIÓN

En la actualidad se ha producido un gran avance en el desarrollo de la inteligencia artificial. Según lo expuesto por Villagomez (2025), esta tecnología se ha consolidado como una fuerza innovadora, dando lugar a un impacto significativo en la sociedad. Su capacidad para desarrollar sistemas y ejecutar tareas que antes requerían de habilidades humanas responde, en gran parte, al aumento de la capacidad de procesamiento, al diseño de algoritmos cada vez más sofisticados y a la disponibilidad masiva de datos generados por la humanidad.

Actualmente, la mayoría de las personas hacen uso de la inteligencia artificial al emplear diferentes modelos creados por distintas empresas, como Chatgpt, Copilot, Claude, etc. Cada uno de estos modelos ayudó a las personas a solucionar problemas rápidamente, resolviendo las dudas que surgen en su quehacer diario y, cuando se trata de investigaciones, llegando a un punto en el que es posible resolver en unas cuantas horas el trabajo que antes se elaboraba en días. Un claro ejemplo se evidencia en la educación, como señalan Kroff, Coria y Ferrada (2024); ahora, los docentes universitarios cuentan con la posibilidad de ofrecer retroalimentación inmediata y personalizada a sus estudiantes. Asimismo, el uso de chatbots en los procesos administrativos permite a los alumnos automatizar tareas repetitivas, lo que libera tiempo para que los profesores se enfoquen en labores más estratégicas y orientadas a la enseñanza.

Las redes neuronales constituyen un componente esencial de la inteligencia artificial. De acuerdo con ibm (2021), pueden definirse como modelos computacionales de Machine Learning diseñados para emular la toma de decisiones del cerebro humano. Su funcionamiento se basa en procesos que reproducen la interacción de las neuronas biológicas, permitiéndole identificar patrones, evaluar distintas alternativas y formular conclusiones de manera autónoma.

Un ejemplo claro de cómo la inteligencia artificial ha contribuido significativamente en el ámbito científico se observa en el campo de la ecofisiología. Ejemplo de ello es el caso mencionado en Gole, Bedi, Marwaha, Haque y Kumar (2023), en el que se desarrolló TrIncNet, un Vision Transformer ligero que

logra identificar enfermedades en plantas a partir de imágenes de sus hojas. Dicho modelo fue entrenado con dos tipos de conjuntos de datos de plantas y demostró mayor precisión que otros modelos existentes, haciendo posible clasificar de manera eficiente los síntomas de estrés biótico en las plantas.

Debido a las necesidades ecofisiológicas de la flora endémica oaxaqueña, en la presente investigación se propone el estudio y aplicación de técnicas de inteligencia artificial mencionadas en la bibliografía, que poseen diferentes características y cuyas áreas de aplicación son diversas, para hacer frente a la problemática de cambio climático y la afectación que provoca en la flora de la ciudad de Oaxaca. En este caso, se pretende evaluar el estado de salud y la forma en que se ve afectada la condición ecofisiológica de los helechos espada.

Para alcanzar este propósito, se plantea como objetivo específico analizar distintos tipos de redes neuronales a fin de seleccionar la que mejor se adapte a este proyecto; en este caso se realizarán pruebas iniciales de detección de números que permitan validar el funcionamiento de los modelos, para posteriormente aplicarlos a imágenes de plantas. Con la intención de validar el modelo en plantas, se escogió el helecho espada como especie referente de la flora oaxaqueña, por lo que se creará un *dataset* de imágenes que será entrenado y se evaluará el modelo seleccionado, a fin de que pueda aportar una estimación sobre el estado de salud de las plantas.

El presente proyecto se divide en dos partes; la primera implica la creación de redes neuronales que identifiquen los números 0 al 9 escritos por el ser humano; con estos códigos podremos realizar uno nuevo que nos ayude a identificar la calidad de vida de los helechos espada a partir de ciertas características. En este caso, nosotros, como seres pensantes, ya tenemos el entrenamiento para relacionar figuras y colores, lo que nos permite asignarles un significado, pues desde pequeños, en la escuela o en la vida diaria hemos aprendido a formar diferentes patrones para ir conformando toda nuestra inteligencia; sin embargo, la computadora requiere empezar de cero, mandando una base de datos y diciendo qué proceso tendrá que realizarse a partir de ésta.

MATERIALES Y MÉTODOS

Marco teórico

Para iniciar el desarrollo de una red neuronal es necesario establecer las bases de conocimiento, a fin de que pueda entenderse lo que se pretende hacer. En este caso debemos entender en primer lugar qué es inteligencia ar-

tificial (ia). La empresa ibm (2024) refiere que es una herramienta capaz de ayudar a los computadores o a cualquier otra máquina a tener la capacidad de hacer una simulación del aprendizaje humano; ésta comprende la resolución de problemas, la toma de decisiones, la comprensión de diferentes entornos, la creatividad y, especialmente, la autonomía para que ella misma genere sus propias conclusiones. Para ello, con diferentes herramientas, la inteligencia artificial puede lograr grandes cosas, por ejemplo, identificar objetos, aprender de nuevas experiencias, entender el lenguaje humano y tomar decisiones en momentos críticos.

Otro concepto relevante es el Machine Learning, también conocido como aprendizaje automático, el cual constituye una rama fundamental de la inteligencia artificial. En los últimos años, éste ha experimentado un notable crecimiento y reconocimiento debido a la gran cantidad de datos que han surgido; ello hace posible que la computadora identifique patrones respecto a ciertos parámetros y tome una decisión sin la necesidad de que esté programada, mejorando el rendimiento de diferentes procesos, lo que puede resultar en ahorro de tiempo y esfuerzo para los usuarios (Forero-Corba y Forero-Corba, 2024).

Teniendo claridad sobre estos dos conceptos es necesario profundizar en cómo las computadoras pueden llegar a tomar decisiones. Para ello, es necesario simular que opera como si fuera un cerebro humano, por lo que se crean redes neuronales artificiales. Como señalan Castañeda Sánchez, Polo Escobar y Vega Huincho (2023), éstas se han consolidado como una herramienta tecnológica de gran relevancia gracias a su aporte a la resolución de problemas que requieren procesos de aprendizaje, simulando el proceso de aprendizaje que tiene lugar en el cerebro humano. Mediante el uso de algoritmos o de programas informáticos se logra el propósito de que aprendan. Asimismo, los autores mencionan que la parte principal de una red son las neuronas. Una neurona biológica funciona como un sistema encargado de procesar señales y adquirir conocimiento mediante procesos sinápticos. Dichos procesos implican la conexión entre dos o más neuronas, mediante la cual la información es interpretada y tratada por medio de mecanismos electroquímicos.

En cambio, como exponen Garduño, Sagols y Wolf (2025), una neurona artificial, a la que también se puede llamar perceptrón, representa un modelo simplificado de neuronas cerebrales. Éstas no derivaron de estudios neurológicos, sino de nociones básicas sobre las neuronas y de la necesidad de implementarlas en computación. Los perceptrones adoptan la idea de recibir múltiples entradas, procesándolas en cada una de sus terminales de entrada y generando una salida.

Las neuronas pueden organizarse de diferentes formas, dando lugar a diversas arquitecturas de red, también conocidas como topologías, como el perceptrón simple o las redes neuronales multicapa. El perceptrón simple representa la estructura más básica, ya que solamente posee una capa de entrada que se conecta directamente con la salida. Por otro lado, las redes multicapa incorporan una o más capas ocultas ubicadas entre la entrada y salida, lo que permite realizar transformaciones más complejas de la información Vorobioff, Cerrotta, Morel y Amadio (2022).

De acuerdo con Chollet (2021), se han diseñado diversas tecnologías que facilitan el aprendizaje profundo, entre las cuales destaca Keras. Esta api de Deep Learning para Python, construida sobre TensorFlow, proporciona herramientas prácticas para diseñar y entrenar diferentes tipos de modelos, con el objetivo principal de lograr una experimentación más ágil. Keras incluye buenas prácticas que llegan a reducir la complejidad para los usuarios, ofreciendo flujos de trabajo consistentes y sencillos. Actualmente, numerosas empresas y organizaciones como Google, Uber, cern, Netflix y la nasa emplean Keras, aprovechando su flexibilidad y la variedad de patrones de uso que ofrece de manera eficiente y confiable.

Tecnologías por ocupar

La planeación para la realización del proyecto implica, en primer lugar, crear las diferentes redes neuronales y sus respectivos entrenamientos, para verificar la tasa de validación.

La programación se realizó en la plataforma de Google Colab, que ofrece herramientas para hacer entrenamientos de *Machine Learning* y brinda cierta cantidad de almacenamiento y de memoria ram de forma gratuita.

Para realizar nuestra red neuronal nos apoyaremos en TensorFlow, que sirve para el modelado de redes neuronales. A través de la página oficial de TensorFlow (2025) se pueden ver las características básicas que debe tener nuestro sistema; las versiones aceptadas de Python son de la 3.9 a la 3.12; esto debe tomarse en cuenta, ya que si se utiliza una versión de Python diferente, no se podrá instalar TensorFlow y no se podrá ejecutar el programa para realizar las diferentes predicciones.

También se ocuparán las siguientes librerías:

-*numpy*: para realizar operaciones numéricas y manejo de arrays.

-PIL (*Pillow*): para trabajar con imágenes (abrir, convertir, mostrar imágenes, etc.).

-*matplotlib*: para crear gráficos y visualizaciones.

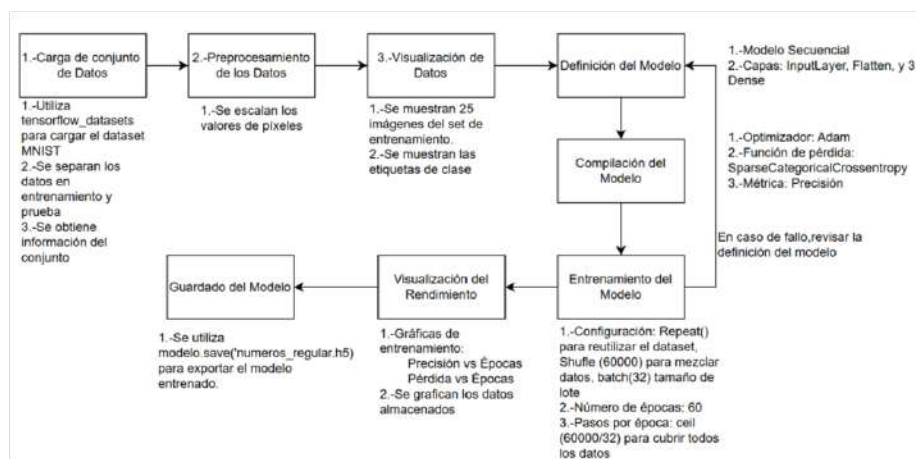
Creación de redes neuronales para la detección de números

Empleando el video Ringa Tech (2021), fue posible desarrollar diferentes redes neuronales capaces de detectar números escritos a mano. Ringa Tech explica tres estilos de desarrollo de redes neuronales: Red Neuronal Regular, Red Neuronal Convolutiva y otra con aumento de datos. Las diferencias entre una y otra estriban en que cada una es mejor que la anterior; la segunda posibilita el manejo de capas, mientras que la última permite realizar diferentes transformaciones a la imagen, por ejemplo, su rotación y traslación, haciendo que la predicción sea más efectiva.

Creación de Red Neuronal Regular

A partir de la primera red neuronal desarrollada, se realizó el diagrama metodológico presentado en la figura 1, que simplifica el entendimiento de la red. En López (Números Red Neuronal Regular, 2025) se puede apreciar todo el código explicado.

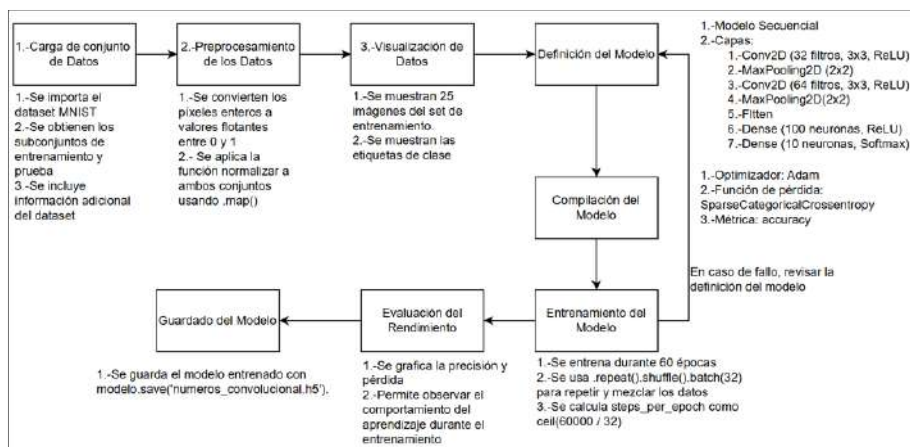
Figura 1. Diagrama metodológico para la creación de una Red Neuronal Regular para detección de números. Fuente: Elaborada por los autores.



Creación de Red Neuronal Convolutacional

Al crear la red neuronal convolutacional se modificó la definición del modelo, agregando más capas, como se puede apreciar en el diagrama metodológico de la figura 2. En López (Números Red Neuronal Convolutacional, 2025) se puede apreciar todo el código realizado.

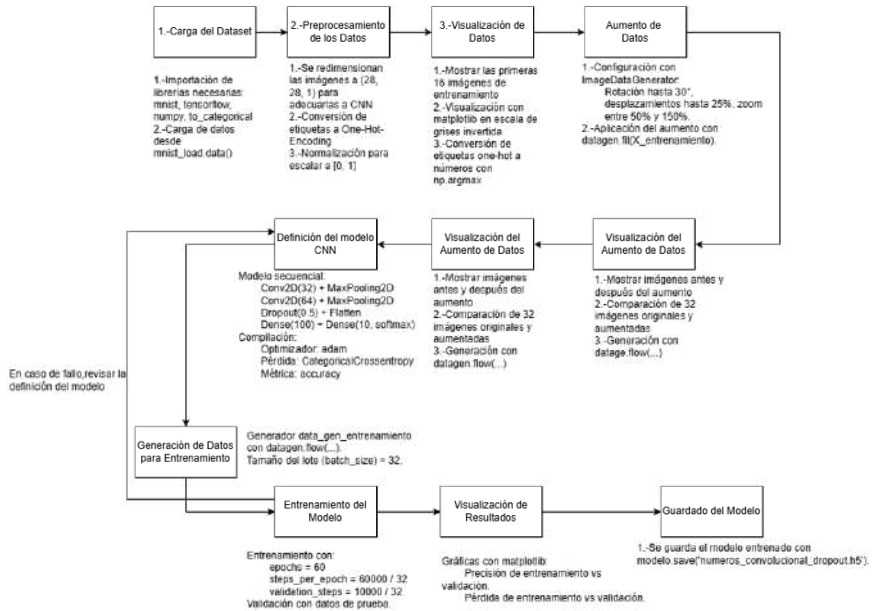
Figura 2. Diagrama metodológico para la creación de una Red Neuronal Convolutacional para detección de números. Fuente: Elaborada por los autores.



Creación de Red Neuronal Convolutacional con aumento de datos y DropOut

Retomando el código de la anterior red neuronal convolutacional, se procedió a agregar características para realizar diferentes transformaciones de las imágenes, por ejemplo, la rotación y la traslación, extendiendo el diagrama metodológico, como puede apreciarse en la figura 3. En López (Números Red Neuronal Convolutacional con Dropout, 2025) se puede ver el código explicado con todas las características implementadas.

Figura 3. Diagrama metodológico para la creación de una Red Neuronal Convolutional con aumento de datos y Dropout. Fuente: Elaborada por los autores



Selección de la planta endémica de Oaxaca

Para diseñar una red neuronal que detecte la salud de una planta endémica de Oaxaca, primero fue necesario elegir entre la gran diversidad existente. A través de diferentes investigaciones se eligió el helecho espada, debido a que, en Oaxaca, como documenta Molfino (2022), existe una riqueza excepcional de helechos, lo que hace que este estado sea un entorno privilegiado para su estudio, ya que alberga a más de 600 especies.

El helecho espada posee características muy particulares de acuerdo con el color de su hoja y su forma. Como menciona Pur Plant (2024), el color de sus hojas dice mucho sobre la salud de la planta; cuando la planta tiene un color muy verde y sus hojas no muestran deformaciones se considera que su estado es muy saludable; si la planta se está tornando amarilla y la forma de sus hojas está cambiando, se considera que se está enfermando; cuando el helecho tiene hojas café o marchitas, se considera que la planta ha muerto, como se aprecia en la figura 4, que muestra la transformación desde el estado saludable al muerto.

Figura 4. Estados de salud de los helechos (PictureThis, 2025; Craftologia, 2025)



Los principales factores que determinan que la planta cambie su estado de salud son el riego incorrecto, el suelo inadecuado, enfermedades y, principalmente, las excesivas temperaturas que se han presentado durante los últimos años.

Creación de *dataset* de fotografías del helecho espada

Después de elegir la planta que se ocuparía, se le tomaron fotos desde diferentes ángulos, con diferentes intensidades de luz y diferentes calidades, para lograr mejorar la capacidad de entrenamiento de nuestra red neuronal. Sólo se consiguieron dos plantas con diferentes estados de salud, saludable y muerta, por lo que, para contar con fotos del helecho enfermo, se recurrió a internet.

El *dataset* se cargó en la plataforma de almacenamiento Google Drive, que puede encontrarse en López, *Dataset de fotografías del helecho espada* (2025), clasificado en carpetas de acuerdo con el estado de salud.

Helecho enfermo: en la figura 5 se puede observar la carpeta en la que están almacenadas las imágenes del helecho, existiendo un total de 99 fotografías.

Helecho saludable: en la figura 6 se muestra la carpeta del helecho saludable, del cual se tomaron 373 fotografías.

Helecho muerto: en la figura 7 se muestra la carpeta del helecho muerto, del cual se tomaron 707 fotografías.

Figura 5. Fotografías del helecho espada enfermo. Fuente: Elaborada por los autores

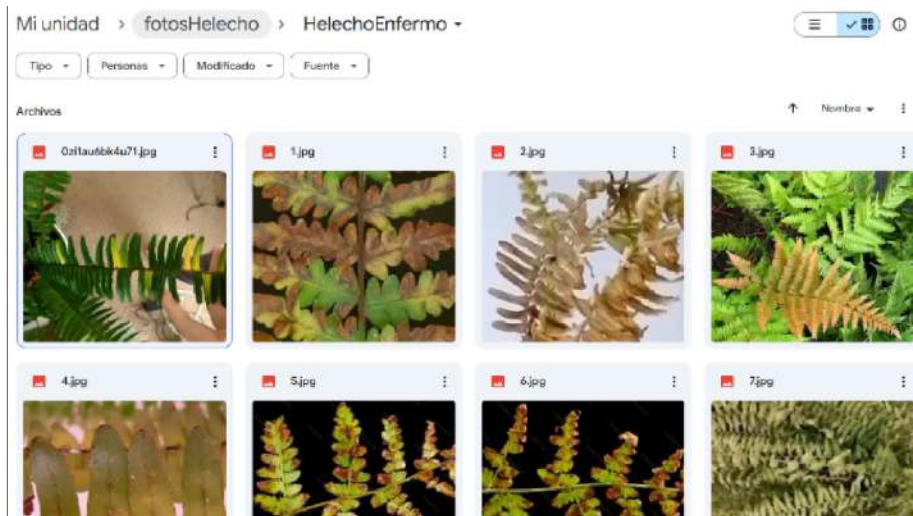


Figura 6. Fotografías del helecho espada muerto. Fuente: Elaborada por los autores

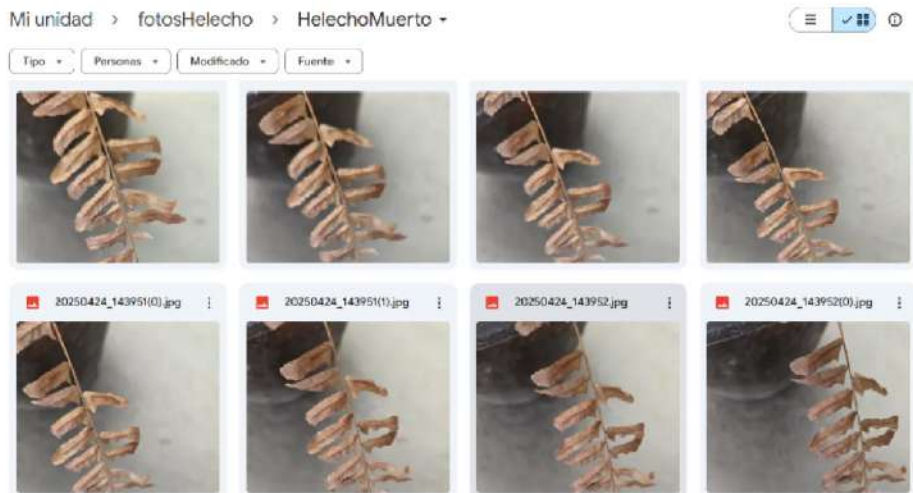
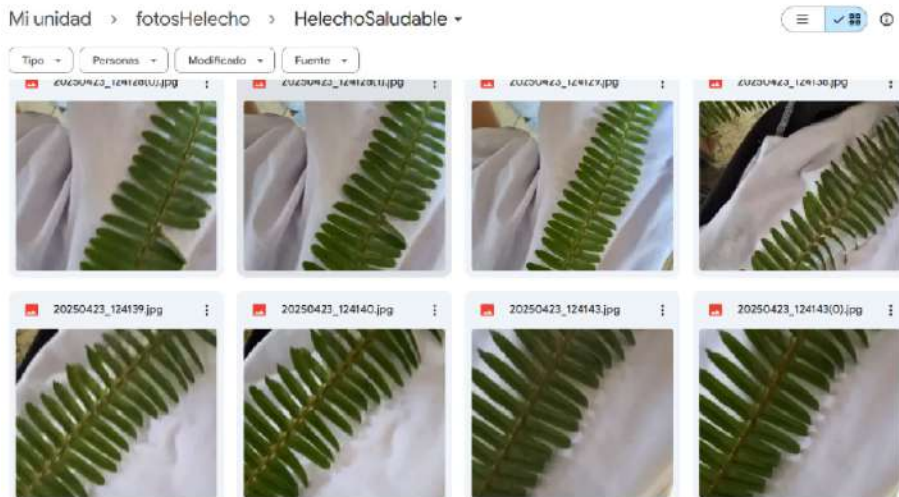


Figura 7. Fotografías del helecho espada saludable. Fuente: Elaborada por los autores



Creación de Red Neuronal Convolutiva para la detección de la salud de los helechos

A partir del análisis previo, se propone la creación de una red neuronal para la detección de la salud de los helechos espada. En este caso se decidió implementar una Red Neuronal Convolutiva, ya que el aumento del número de capas posibilita una mayor capacidad de aprendizaje y una mejor flexibilidad en comparación con una Red Neuronal Regular, que funciona mejor con datos tabulares. Esto responde a que, al trabajar con imágenes, es necesario tomar en cuenta la posición y los patrones de los píxeles, lo que hace que el aprendizaje computacional sea más eficiente.

Esta red fue alimentada con imágenes organizadas en tres categorías, helechos saludables, helechos enfermos y helechos muertos. Las imágenes fueron recolectadas y clasificadas manualmente y almacenadas en Google Drive para su acceso desde el entorno de Google Colab.

1.-Preprocesamiento de imágenes y aumento de datos

Para realizar el preprocesamiento de las imágenes, se empleó la clase `ImageDataGenerator` proporcionada por Keras, incluyendo:

Normalización de los píxeles a un rango [0,1]

Aumento de datos mediante transformaciones aleatorias

Rotaciones de hasta 30°

Desplazamientos horizontales y verticales

Escalamiento con zoom

Volteo horizontal

División del conjunto de datos en entrenamiento (80%) y validación (20%) mediante el parámetro `validation_split`

Se ajustaron las dimensiones de las imágenes a 224x224 píxeles y se las organizó en lotes de 32.

2.-Visualización de datos

Para verificar la correcta lectura y etiquetado de las imágenes cargadas, se implementó una función que muestra las imágenes de cada clase con su respectiva etiqueta, permitiendo observar la diversidad de los datos.

3.-Arquitectura del modelo

El modelo cnn fue construido secuencialmente, con la siguiente arquitectura:

Entrada: imágenes rgb de 224x224x3 píxeles.

Capas convolucionales y de pooling

Conv2D con 32 filtros y MaxPooling2D

Conv2D con 64 filtros y MaxPooling2D

Conv2D con 128 filtros y MaxPooling2D

Regularización: Dropout de 50% después de la última capa convolucional y la capa densa intermedia.

Capas densas finales:

Una capa Dense con 128 neuronas (ReLU)

Una capa Dense con tres salidas y la aplicación de la función *softmax* para la clasificación en escenarios multiclase

4.-Compilación

El modelo se compiló utilizando el optimizador Adam

Se empleó la función de pérdida categorical crossentropy (apropiada para clasificación multiclase con one-hot encoding)

La métrica utilizada para evaluar el modelo fue la precisión (accuracy)

5.-Entrenamiento

Se realizó durante 30 épocas, utilizando el generador de datos con aumento para el conjunto de entrenamiento

Generador normalizado para el conjunto de validación

6.-Evaluación del modelo

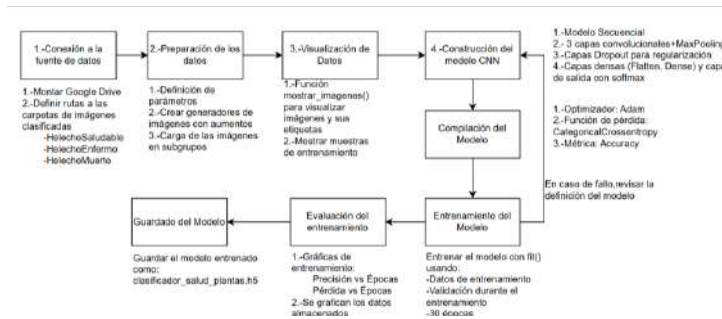
Se representaron gráficamente las curvas de precisión y de pérdida correspondientes al entrenamiento y validación, con el propósito de analizar el rendimiento del modelo a lo largo de las épocas.

7.-Guardado del modelo

El modelo se guarda en formato HDF5, el cual permite su posterior carga y reutilización sobre nuevos datos de helechos.

A continuación, en la figura 8 se puede apreciar el diagrama metodológico y en López (Detector de salud de helechos espada, 2025) se puede encontrar el código creado.

Figura 8. Diagrama metodológico del primer modelo de detección de salud en helechos.
 Fuente: Elaborada por los autores.



Creación de Red Neuronal Convolutiva mejorada a partir de la aplicación de filtros a las imágenes

A raíz de que existen áreas de mejora en el anterior código expuesto, se implementó la aplicación de diferentes filtros a la imagen para que tuviera un mejor desempeño al momento de predecir.

1.-Montaje del entorno y adquisición de datos

Se realizó la misma configuración que en el caso del anterior código explicado.

2.-Preprocesamiento de datos

Se establecieron dimensiones de 224x224 píxeles, acordes a las necesidades del modelo MobileNetV2. A continuación, se aplicó un preprocesamiento dual:

Se normalizaron las imágenes, estableciendo un rango de píxel de [0, 1]

Se transformaron las imágenes del espacio rgb al espacio hsv (Matiz, Saturación, Valor), más adecuado para el análisis de vegetación. Además, se calcularon histogramas normalizados de cada canal, generando un vector de características por imagen. Este vector de características puede ser útil para futuras tareas de análisis.

3.-Aumento de datos

Para optimizar la capacidad de generalización del modelo o mitigar el sobreajuste ocasionado por el tamaño reducido de datos, se aplicó un aumento de datos.

Rotaciones aleatorias (hasta 30°)

Desplazamientos horizontales y verticales (hasta 25%)

Zooms variables (0.5-1.5)

Inversión horizontal

Alteraciones en el brillo (0.7-1.3)

4.-Visualización y análisis exploratorio

Se implementó una función de visualización que, para un subconjunto de imágenes, muestra tanto la imagen original como sus canales hsv individuales. Esto facilitó el análisis visual de patrones de color y de degradación foliar asociada con las clases enfermas y muertas.

5.-Arquitectura del modelo

Para el núcleo del modelo se utilizó MobileNetV2, una red convolucional ligera preentrenada en ImageNet, la cual ayuda a los dispositivos cuyos recursos son limitados. Se empleó la arquitectura sin su capa de clasificación final (include_top=False), utilizando sus características internas como extractor de rasgos de forma y textura.

Las capas superiores del modelo fueron adaptadas con:

Una capa de GlobalAveragePooling2D para reducir dimensionalidad

Dos bloques Dense + *BatchNormalization* + *Dropout* para agregar capacidad de aprendizaje específica:

256 unidades (para integrar características visuales generales)

128 unidades (enfocadas en detalles más finos)

Capa de salida compuesta por tres neuronas con función de activación *softmax*, diseñada para realizar clasificación multiclase.

Las primeras 100 capas del modelo base se congelaron para preservar los patrones visuales básicos previamente aprendidos.

6.-Compilación

Se compiló utilizando el optimizador Adam y la función de pérdida categorical crossentropy, apropiada para tareas de clasificación multiclase con etiquetas codificadas mediante el esquema one-hot.

Durante el entrenamiento se utilizaron dos callbacks

Early stopping: interrumpe el entrenamiento cuando la pérdida en validación deja de mejorar después de cinco épocas consecutivas.

ReduceLRonPlateau: disminuye la tasa de aprendizaje si el modelo no mejora durante tres épocas consecutivas

7.-Entrenamiento

El entrenamiento se llevó a cabo durante un máximo de 30 épocas, empleando lotes de 32 imágenes.

8.-Evaluación del entrenamiento

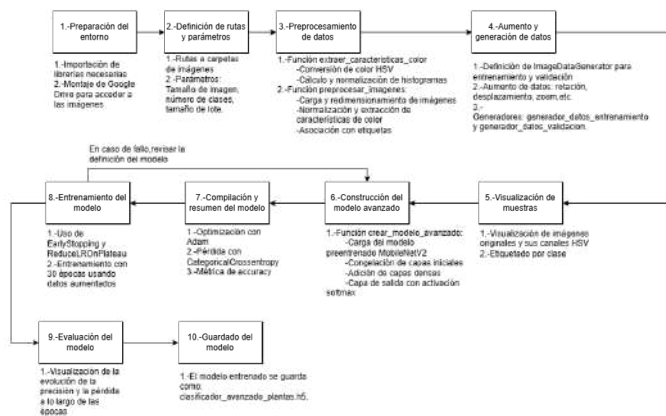
Se muestra un gráfico que da cuenta de la precisión y de la pérdida en los conjuntos de entrenamiento y validación.

9.-Guardado del modelo

Por último, el modelo entrenado fue guardado en formato hdf5, para permitir su reutilización o integración en sistemas de predicción.

A continuación, en la figura 9 se puede apreciar el diagrama metodológico y en López (Detector de salud de helechos espada mejorado, 2025) se encontrará el código realizado.

Figura 9. Diagrama metodológico del segundo modelo de detección de salud en helechos.
Fuente: Elaborada por los autores.



RESULTADOS

La creación de redes neuronales para el reconocimiento de números permitió identificar el modelo que mejor se adapta a las necesidades de esta investigación. Se seleccionó la Red Neuronal Convolutiva, debido a su capacidad para detectar patrones incluso en datos que no se encuentran explícitamente en el conjunto de entrenamiento.

En este caso, se eligió al helecho espada, una planta representativa de la flora oaxaqueña que presenta características particulares en sus distintos estados ecofisiológicos. Estos estados se reflejan directamente en el color y la forma de sus hojas, lo que facilitó su análisis e identificación.

Para el entrenamiento se recopilaron 1179 fotografías, que fueron clasificadas en tres categorías: saludable, enferma y muerta. Además, las imágenes se tomaron con distintas calidades, con el objetivo de que la red fuera capaz de reconocerlas sin importar la resolución o las condiciones de captura.

Finalmente, se construyó la Red Neuronal Convolutiva, incorporando como principales variables la detección de color y la morfología de las hojas, a fin de garantizar un modelo para la identificación de los estados de la planta.

Red Neuronal Convolutiva (primera versión)

Cuando se empezó a ejecutar, la primera red neuronal convolutiva pudo cargar exitosamente las imágenes y etiquetarlas en los tres distintos tipos previstos (saludable, enferma, muerta). Asimismo, en este caso se agregó una función para visualizar cómo se etiquetaban las imágenes, lo que se aprecia en la figura 10; además, éstas se normalizaron a cierto tamaño con algunas transformaciones.

Figura 10. Fotografías de los helechos correctamente cargadas y etiquetadas. Fuente: Elaborada por los autores



Posteriormente se hizo la compilación del modelo, por lo que la plataforma nos devolvió un resumen que muestra una descripción detallada de la arquitectura de la red neuronal convolucional construida. La tabla incluye información sobre cada capa del modelo, como el tipo, el tamaño de la capa de salida y la cantidad de parámetros entrenables, como se aprecia en la figura 11.

Layer (Type): indica el nombre y el tipo de la capa

Output Shape: da cuenta de la forma de los datos que salen en cada capa. Por ejemplo, None, 111, 111, 32 indica que esa capa produce un volumen de salida de 111x111 píxeles con 32 filtros o canales. None indica que el tamaño del lote (batch size) es variable.

Param # (Parámetros): es el número total de parámetros entrenables en esa capa. Para una capa Conv2D, se calcula:

$$(\text{filtro}_{\text{ancho}} * \text{filtro}_{\text{alto}} * \text{canales}_{\text{entrada}} + 1) * \text{número}_{\text{de filtros}}$$

Donde + 1 representa al término de sesgo (bias)

Al final se presenta un resumen global del modelo:

Total params: representa la suma de todos los parámetros del modelo, tanto los entrenables como los no entrenables

Trainable params: son aquellos parámetros que se modifican y optimizan durante el proceso de entrenamiento

Non-trainable params: son los parámetros que se mantienen fijos, generalmente por capas congeladas

Figura 11. Resumen de la compilación del primer modelo. Fuente: Elaborada por los autores.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
dropout (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11,875,712
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387

Total params: 11,169,347 (42.61 MB)
 Trainable params: 11,169,347 (42.61 MB)
 Non-trainable params: 0 (0.00 B)

Después empezó el entrenamiento, que concluyó satisfactoriamente, como se puede observar en la figura 12, llegando a las 30 épocas con una exactitud de 97%.

Figura 12. Entrenamiento del primer modelo concluido satisfactoriamente. Fuente: Elaborada por los autores.

Epoch 25/30	143s 5s/step - accuracy: 0.9804 - loss: 0.0884 - val_accuracy: 0.9828 - val_loss: 0.0536
Epoch 26/30	140s 5s/step - accuracy: 0.9894 - loss: 0.0414 - val_accuracy: 0.9612 - val_loss: 0.0736
Epoch 27/30	159s 5s/step - accuracy: 0.9798 - loss: 0.0772 - val_accuracy: 0.9267 - val_loss: 0.1462
Epoch 28/30	149s 5s/step - accuracy: 0.9717 - loss: 0.0966 - val_accuracy: 0.9612 - val_loss: 0.0930
Epoch 29/30	138s 5s/step - accuracy: 0.9725 - loss: 0.0838 - val_accuracy: 0.9569 - val_loss: 0.1045
Epoch 30/30	141s 5s/step - accuracy: 0.9798 - loss: 0.0740 - val_accuracy: 0.9784 - val_loss: 0.0703

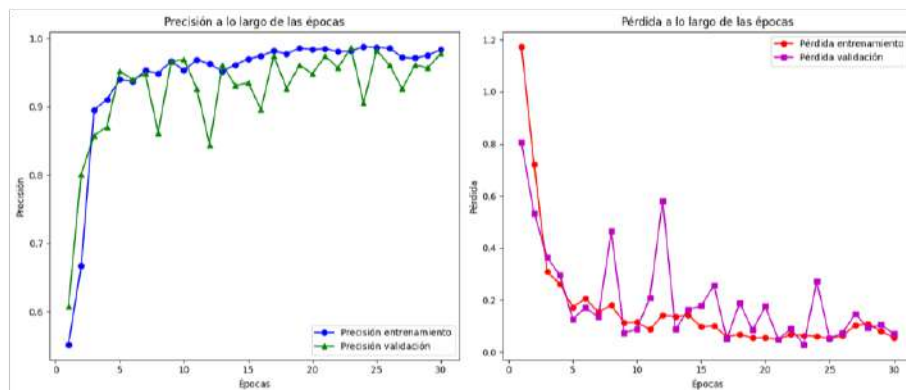
Por último, en la figura 13 se muestran dos gráficas que comparan la precisión y la pérdida a lo largo de las épocas del entrenamiento. Se observa que, aunque en general la precisión va en aumento y la pérdida tiende a disminuir, existen algunos picos irregulares. Estos picos indican que, durante algunas épocas, en el modelo no se generalizó correctamente, lo que puede deberse a varios factores:

Sobreajuste: el modelo aprende detalles propios del conjunto de entrenamiento que no se presentan en los datos de validación.

Complejidad del conjunto de datos: la alta variabilidad entre las imágenes puede dificultar que el modelo aprenda

Tamaño del *dataset*: conjunto de datos pequeños o balanceados (Google for Developers, 2025)

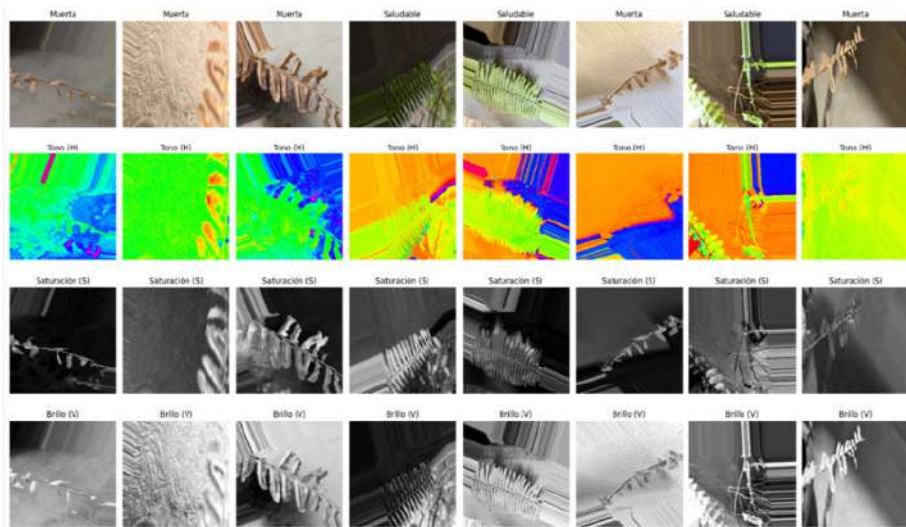
Figura 13. Gráfica de precisión y pérdida durante el entrenamiento del primer modelo.
Fuente: Elaborada por los autores.



Red Neuronal Convolutiva Mejorada

En el caso anterior se pudo apreciar que hubo picos irregulares durante el entrenamiento, por lo que se mejoró el código para que detectara más características a fin de definir mejor la curva de precisión y la de pérdida. Tras recuperar nuevamente las imágenes, se las transformó al canal hsv, como se puede apreciar en la figura 14, que muestra ejemplos de fotografías en sus diferentes canales.

Figura 14. Fotografías de helechos en canales HSV. Fuente: Elaborada por los autores.



La compilación del modelo se realizó en un modelo funcional, retomando la arquitectura basada en MobileNetV2 preentrenada sobre ImageNet. La tabla que muestra el resumen incluye información sobre cada capa del modelo, como el tipo, la dimensión de salida, la cantidad de parámetros entrenables y la capa a la que se encuentra conectada, como se aprecia en la figura 15.

En la tabla se pueden ver las siguientes columnas:

Layer (Type): nombre de la capa y tipo

Output Shape: la forma del tensor de salida que genera esa capa

Param #: números de parámetros entrenables (y no entrenables) en la capa

Connected to: qué capa entrega su salida como entrada a esta capa

Al final se presenta un resumen global del modelo:

Total params: es el total de parámetros entrenables y no entrenables del modelo

Trainable params: son los parámetros que se ajustan durante el entrenamiento

Non-trainable params: son los parámetros que se mantienen fijos, generalmente por capas congeladas

Figura 15. Resumen de la compilación del segundo modelo. Fuente: Elaborada por los autores.

Layer (type)	Output Shape	Params	Connected to			
input_layer (InputLayer)	(None, 224, 224, 3)	0	-	block_16_project (Conv2D)	(None, 7, 7, 320)	307,200
Conv1 (Conv2D)	(None, 112, 112, 32)	384	input_layer[0][0]	block_16_project_BN (BatchNormalizati...	(None, 7, 7, 320)	1,280
bn_Conv1 (BatchNormalizati...	(None, 112, 112, 32)	0	conv1[1][0]	Conv_1 (Conv2D)	(None, 7, 7, 1280)	488,600
Conv2_relu (Conv2D)	(None, 56, 56, 32)	8	bn_Conv1[1][0]	Conv_1_bn (BatchNormalizati...	(None, 7, 7, 1280)	5,120
expanded_conv_dept_ (DepthwiseConv2D)	(None, 56, 56, 32)	384	conv2_relu[1][0]	out_relu (ReLU)	(None, 7, 7, 1280)	0
expanded_conv_dept_ (BatchNormalizati...	(None, 56, 56, 32)	0	expanded_conv_de...	global_average_poo... (GlobalAveragePool...	(None, 1280)	0
expanded_conv_dept_ (Conv2D)	(None, 56, 56, 32)	128	expanded_conv_dept_...	dense (Dense)	(None, 256)	327,856
expanded_conv_dept_ (BatchNormalizati...	(None, 56, 56, 32)	0	expanded_conv_dept_...	batch_normalization (BatchNormalizati...	(None, 256)	1,024
block_1_expand (Conv2D)	(None, 112, 112, 32)	1,280	expanded_conv_dept_...	dropout (Dropout)	(None, 256)	0
block_1_expand_BN (BatchNormalizati...	(None, 112, 112, 32)	0	block_1_expand[0][0]	dense_1 (Dense)	(None, 128)	32,896
block_1_expand_relu (ReLU)	(None, 112, 112, 32)	8	block_1_expand_BN[0][0]	batch_normalizatio... (BatchNormalizati...	(None, 128)	512
block_1_pool (MaxPooling2D)	(None, 56, 56, 32)	0	block_1_expand_relu[0][0]	dropout_1 (Dropout)	(None, 128)	0
block_1_depthwise (DepthwiseConv2D)	(None, 56, 56, 32)	384	block_1_pool[0][0]	dense_2 (Dense)	(None, 3)	387
block_1_depthwise_BN (BatchNormalizati...	(None, 56, 56, 32)	0	block_1_depthwise...	Total params: 3,620,739 (10.80 MB) Trainable params: 2,223,479 (8.48 MB) Non-trainable params: 357,312 (1.52 MB)		
block_1_depthwise_relu (ReLU)	(None, 56, 56, 32)	8	block_1_depthwise_BN[0][0]			
block_1_project (Conv2D)	(None, 56, 56, 320)	3,360	block_1_depthwise_relu[0][0]			

Posteriormente se realizó el respectivo entrenamiento, como se muestra en la figura 16, no generándose ningún tipo de error y llegando a una exactitud de 96%.

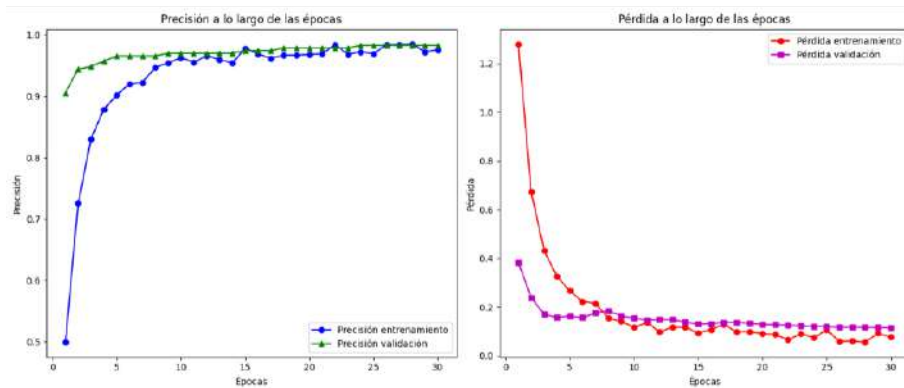
Figura 16. Entrenamiento del segundo modelo concluido satisfactoriamente. Fuente: Elaborada por los autores.

```

Epoch 1/100: 239s 8s/step - accuracy: 0.4700 - loss: 0.3289 - val_accuracy: 0.9530 - val_loss: 0.2282 - learning_rate: 1.0000e-04
Epoch 5/100: 235s 8s/step - accuracy: 0.9017 - loss: 0.2731 - val_accuracy: 0.9573 - val_loss: 0.2063 - learning_rate: 1.0000e-04
Epoch 6/100: 234s 8s/step - accuracy: 0.9396 - loss: 0.1653 - val_accuracy: 0.9487 - val_loss: 0.2785 - learning_rate: 1.0000e-04
Epoch 7/100: 232s 8s/step - accuracy: 0.9385 - loss: 0.1675 - val_accuracy: 0.9487 - val_loss: 0.2818 - learning_rate: 1.0000e-04
Epoch 8/100: 229s 7s/step - accuracy: 0.9516 - loss: 0.1579 - val_accuracy: 0.9487 - val_loss: 0.3061 - learning_rate: 1.0000e-04
Epoch 9/100: 230s 8s/step - accuracy: 0.9035 - loss: 0.1357 - val_accuracy: 0.9530 - val_loss: 0.2362 - learning_rate: 5.0000e-05
Epoch 10/100: 217s 7s/step - accuracy: 0.9632 - loss: 0.1188 - val_accuracy: 0.9530 - val_loss: 0.2018 - learning_rate: 5.0000e-05
    
```

Tras los cambios realizados, la figura 17 muestra que la gráfica registra una curvatura sin tanta irregularidad; ello permite inferir que el modelo está recibiendo datos consistentes y bien preprocesados.

Figura 17. Gráficas de precisión y de pérdida durante el entrenamiento del segundo modelo. Fuente: Elaborada por los autores.



CONCLUSIONES

Este trabajo permitió integrar conceptos teóricos y prácticos de la inteligencia artificial para aplicaciones en el ámbito de la ecofisiología, específicamente en la evaluación del estado de salud de los helechos espada, mediante la creación de una red neuronal convolucional. Partiendo de los primeros modelos creados para la detección de números (mnist), se logró adaptar esta metodología al reconocimiento de patrones en imágenes de plantas de manera de realizar su monitoreo.

En el modelo desarrollado fue necesario construir un *dataset* con imágenes de la planta que escogimos para representar a la flora oaxaqueña, el helecho espada. Se recopilaron 1 179 imágenes, utilizadas para el entrenamiento de la red neuronal.

En el caso de los helechos enfermos, el número de imágenes disponibles fue reducido, porque las mismas tuvieron que obtenerse de internet al no contar con ejemplares en esa condición. En contraste, para los demás estados se disponía de la planta físicamente, por lo que se tomaron fotografías propias desde distintos ángulos y con diversas calidades. Esta estrategia permitió que la red neuronal aprendiera a reconocer la planta en diferentes condiciones y con diversas características visuales.

El primer modelo de red neuronal se basó en una arquitectura secuencial sencilla y un conjunto limitado de imágenes, logrando alcanzar una precisión

de validación de 97%. Sin embargo, se observaron irregularidades durante el entrenamiento, reflejadas en las gráficas en que se visualizan la pérdida y la precisión; esto responde, sobre todo, a la cantidad limitada de datos, su variabilidad y la presencia de posibles sesgos en el conjunto de entrenamiento. Además, se identificó que, ante imágenes no contenidas en el *dataset*, el modelo podía generar predicciones erróneas, lo cual enfatiza la necesidad de contar con un *dataset* limpio y balanceado.

Las limitantes de la primera red neuronal llevaron a implementar el análisis de color hsv, lo que permitió extraer características más representativas de las imágenes, especialmente en los aspectos relacionados con el color y la iluminación. Esta mejora redujo las irregularidades observadas durante el entrenamiento, logrando una precisión de 96%, con mayor estabilidad y generalización.

Asimismo, el cambio de la arquitectura secuencial a un modelo funcional hizo posible una mayor versatilidad en el diseño estructural de la red, lo que ayudó a la adaptación del modelo. Además, esta modificación facilitó la incorporación de múltiples entradas o ramas en futuras versiones.

PRÓXIMOS AVANCES

Como próximas mejoras a este proyecto se estará realizando el diseño de una interfaz que permita ocupar los entrenamientos realizados para predecir la calidad de vida de la planta; además, se podrá mejorar el código y el *dataset* para que sea capaz de determinar con mayor exactitud.

AGRADECIMIENTOS

Al Grupo de Investigación, Desarrollo e Innovación-GI+D+i Aplicaciones steam de la Universidad La Salle Oaxaca, por su colaboración y respaldo en el desarrollo de esta investigación.

REFERENCIAS

- Castañeda Sánchez, W. A., Polo Escobar, B. R., y Vega Huincho, F. (2023). Redes neuronales artificiales: una medición de aprendizajes de pronósticos como demanda potencial. *Universidad, Ciencia y Tecnología*, 27(118), 51-60. doi:<https://doi.org/10.47460/uct.v27i118.686>
- Chollet, F. (2021). *Deep Learning with Python*. MANNING. ISBN: 9781617296864
- Craftologia (2025). *Helecho enfermo [imagen]*. Obtenido de Craftologia: <https://www.craftologia.com/tips/hogar/jardineria/como-quitar-el-hongo-de-un-helecho-sin-marchitarlo>
- Forero-Corba, W., y Forero-Corba, W. (2024). Técnicas y aplicaciones del Machine Learning e Inteligencia Artificial en educación: una revisión sistemática. *RIED-Revista Iberoamericana de Educación a Distancia*, 27(1), 209-253. doi:<https://doi.org/10.5944/ried.27.1.37491>
- Garduño Alvarado, T., Sagols Troncoso, F., y Wolf, G. (2025). Las neuronas artificiales y su papel central en la inteligencia artificial. *CViencia*, 76(1), 68-75. Obtenido de https://www.revistaciencia.amc.edu.mx/images/revista/76_1/PDF/13_76_1_1582_NeuronasArtificiales.pdf
- Gole, P., Bedi, P., Marwaha, S., Haque, A., y Kumar, C. (2023). TrIncNet: a lightweight vision transformer network for identification of plant diseases. *Frontiers in Plant Science*, 14. doi:<https://doi.org/10.3389/fpls.2023.1221557>
- Google for Developers (2025). Sobreajuste: Interpretación de curvas de pérdida. Obtenido de Conceptos de AA: <https://developers.google.com/machine-learning/crash-course/overfitting/interpreting-loss-curves?hl=es-419>
- IBM (2021). *¿Qué son las redes neuronales?* Obtenido de ibm: <https://www.ibm.com/mx-es/think/topics/neural-networks>
- IBM (2024). *¿Qué es la inteligencia artificial (IA)?* Obtenido de ibm: <https://www.ibm.com/mx-es/think/topics/artificial-intelligence>

- Kroff, F., Coria, D., y Ferrada, C. (2024). Inteligencia Artificial en la educación universitaria: Innovaciones, desafíos y oportunidades. *Revista Espacios*, 45(5), 120-135. doi:<https://doi.org/10.48082/espacios-a24v45n05p09>
- López, U. (2025). *Dataset de fotografías del helecho espada*. Obtenido de Google Drive: <https://drive.google.com/drive/folders/1SIJ7wtH42b97UZn0iy-IDp-mCRvdEXIbZ?usp=sharing>
- López, U. (2025). *Detector de salud de helechos espada*. Obtenido de Google Colab: https://colab.research.google.com/drive/16ShNUuKSG_3kKMqlkFwbYnd-nVmJZDvCJ?usp=sharing
- López, U. (2025). *Detector de salud de helechos espada mejorado*. Obtenido de Google Colab: <https://colab.research.google.com/drive/1jrvORMeuNI8F-BUY-nUnygE3qrigyORs9?usp=sharing>
- López, U. (2025). *Números Red Neuronal Convolutiva*. Obtenido de Google Colab: <https://colab.research.google.com/drive/17CSsuN5NjWDsKrR9IU-oZSkSOL-nM3eKN?usp=sharing>
- López, U. (2025). *Números Red Neuronal Convolutiva con Dropout*. Obtenido de Google Colab: https://colab.research.google.com/drive/1gmW_f3uym-M7ce_aaaQ4XLIYbRQBOENxG?usp=sharing
- López, U. (2025). *Números Red Neuronal Regular*. Obtenido de Google Colab: <https://colab.research.google.com/drive/1DKUKTabHyPFSIIRNhjW69Z-0JMv6UxafW?usp=sharing>
- Molfino, F. (2022). Oaxaca, el destino menos imaginado donde abundan los helechos. Obtenido de Travesías: <https://www.travesiasdigital.com/destinos/mexico/el-universo-oaxaqueno-que-fascino-a-oliver-sacks/>
- PictureThis (2025). Helecho espada [imagen]. Obtenido de PictureThis: https://www.picturethisai.com/es/care/Nephrolepis_exaltata.html
- Pur Plant. (2024). Problemas comunes en plantas helechos. Obtenido de Pur Plant: <https://www.purplant.es/guia-problemas-plantas/helecho/?srsltid=AfmBOoq-Z4MDi5q9kqC6p6yZzxXr9uslMEGu-0qskbzOqu6VvAoxw- ma>

Ringa Tech (2021). *Redes Neuronales Convolucionales - Clasificación avanzada de imágenes con IA / ML (cnn)* [Video]. Obtenido de YouTube: <https://www.youtube.com/watch?v=4sWhhQwHqug&list=LL&index=7&t=1275s>

TensorFlow (2025). *Instalar TensorFlow con pip*. Obtenido de TensorFlow: <https://www.tensorflow.org/install/pip?hl=es>

Villagomez Palacios, A. H. (2025). El impacto de la Inteligencia Artificial en la Sociedad: Una Revisión Sistemática de su Influencia en Ámbitos Sociales, Económicos y Tecnológicos. *Ciencia Latina*, 9(1), 8150-8172. doi:https://doi.org/10.37811/cl_rcm.v9i1.16468

Vorobioff, J., Cerrotta, S., Morel, N. E., y Amadio, A. (2022). *Inteligencia Artificial y Redes Neuronales: Fundamentos, Ejercicios y Aplicaciones con Python y Matlab*. edUTecNe. ISBN 978-987-4998-82-8

Derechos de Autor© 2025 López Hernández, Uziel; Silva Cruz, Eric Mario; Barcelos Mendoza, Ludwig Federico.



Este texto está protegido por una licencia [Creative Commons 4.0](https://creativecommons.org/licenses/by/4.0/). Usted es libre para Compartir, copiar y redistribuir el material en cualquier medio o formato y adaptar el documento, remezclar, transformar y crear a partir del material— para cualquier propósito, incluso para fines comerciales, siempre que cumpla la condición de: Atribución: Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.