

**Redes neuronales para la identificación de
enfermedades en el cultivo de tomate (*Solanum
lycopersicum* L.) en Honduras**

**Jorge Deninson Erazo Valdiviezo
Ken Sebastian Uema Higa**

**Escuela Agrícola Panamericana, Zamorano
Honduras**
Noviembre, 2020

ZAMORANO
CARRERA DE INGENIERÍA AGRONÓMICA

Redes neuronales para la identificación de enfermedades en el cultivo de tomate (*Solanum lycopersicum* L.) en Honduras

Proyecto especial de graduación presentado como requisito parcial para optar al título de Ingeniero Agrónomo en el Grado Académico de Licenciatura

Presentado por

Jorge Deninson Erazo Valdiviezo
Ken Sebastian Uema Higa

Zamorano, Honduras
Noviembre, 2020

Redes neuronales para la identificación de enfermedades en el cultivo de tomate (*Solanum lycopersicum* L.) en Honduras

Presentado por:

Jorge Deninson Erazo Valdiviezo
Ken Sebastian Uema Higa

Aprobado:



Carolina Avellaneda, Ph.D.
Asesora Principal



Rogel Castillo, M.Sc.
Director
Departamento de Ciencia y
Producción Agropecuaria



Josué Molina, Ph.D.
Asesor



Luis Fernando Osorio, Ph.D.
Vicepresidente y Decano Académico

Redes neuronales para la identificación de enfermedades en el cultivo de tomate (*Solanum lycopersicum* L.) en Honduras

Jorge Deninson Erazo Valdiviezo
Ken Sebastian Uema Higa

Resumen: En la actualidad el tomate (*Solanum lycopersicum* L.) es la hortaliza más importante productiva y económicamente en Honduras, siendo un insumo indispensable del comercio a nivel nacional y global, por lo cual es sumamente necesario contar con herramientas de asesoría agrícola, que ayuden en la productividad y calidad del cultivo. El proyecto aquí presentado consistió en la implementación de mecanismos de inteligencia artificial, de la creación de un sistema de redes neuronales que permitan diagnosticar enfermedades en el tomate de forma confiable y precisa. La codificación del sistema se realizó empleando códigos de programación obtenidos de la colorimetría de síntomas, usando el modelo de color RGB y el lenguaje de programación Python, el cual facilitó la implementación y entrenamiento de la red, para esto fue necesario crear una base de datos de la cual la red neuronal obtuvo las imágenes necesarias para realizar su entrenamiento y aprendizaje, aplicándolo luego en sus diagnósticos. El desarrollo del programa presentó algunas dificultades o limitaciones, lo que limitó el número de enfermedades a evaluar, por tal razón el estudio se centró en aquellas que presentaron síntomas bien definidos, y de fácil interpretación para la red neuronal, permitiendo extenderse a cualquier cantidad de enfermedades. Se realizó la codificación de las enfermedades y su implementación, logrando que el sistema tuviese la capacidad de diagnosticar correctamente dos ellas, mancha por *Septoria* sp. y tizón temprano.

Palabras clave: Inteligencia artificial, “Machine Learning”, Python.

Abstract: Currently, tomato (*Solanum lycopersicum* L.) is the most important vegetable productively and economically in Honduras, being an indispensable input of trade at the national and global level, for which it is highly necessary to have agricultural advisory tools, which help in the productivity and quality of the crop. The present study implemented artificial intelligence mechanisms for the creation of a neural networks system that allow diagnosing diseases in tomatoes in a reliable and precise way. The system coding was carried out using programming codes obtained from the colorimetry of symptoms using the RGB color model and Python programming language, which facilitated the implementation and training of the network. For this, it was necessary to create a database of which the neural network obtained the necessary images to carry out its training and learning, applying it later in its diagnoses. The development of the program presented some difficulties or limitations, which limited the number of diseases to be evaluated for the reason the study focused on those that presented well-defined symptoms, and easily interpreted by the neural network, allowing it to be extended to any number of diseases. The coding of the diseases and their implementation were carried out, ensuring that the system had the ability to correctly diagnose two of them, *Septoria* leaf spot and early blight.

Key words: Artificial intelligence, “Machine Learning”, Python.

ÍNDICE GENERAL

Portadilla.....	i
Página de firmas	ii
Resumen	iii
Índice General	iv
Índice De Cuadro Y Figuras	v
1. INTRODUCCIÓN.....	1
2. METODOLOGÍA.....	3
3. RESULTADOS Y DISCUSIÓN.....	6
4. CONCLUSIONES	12
5. RECOMENDACIONES	13
6. LITERATURA CITADA.....	14
7. ANEXOS	16

ÍNDICE DE CUADRO, FIGURAS Y ANEXOS

Cuadro	Página
1. Enfermedades seleccionadas de las que se recopilaron imágenes en la base de datos que causan daño foliar con una expresión clara de sus síntomas en la planta de tomate	7

Figuras	Página
1. Síntomas del tizón temprano del tomate	8
2. Síntomas de la mancha foliar por <i>Septoria</i>	10
3. Aplicación del algoritmo en las imágenes etiquetadas.....	10
4. Prueba de la efectividad del método utilizado para los diagnósticos	10
5. Comparación del desempeño de una red entrenada y otra sin entrenar	11
6. Prueba final de diagnóstico después de los entrenamientos y comprobada la efectividad del método que se utilizó	11

Anexos	Página
1. Código de implementación.....	16
2. Ejemplo de uso	20

1. INTRODUCCIÓN

El tomate (*Solanum lycopersicum* L.), es una de las hortalizas de mayor consumo en el mundo y la que cuenta con mayor valor económico. Su demanda y comercio aumenta día a día y en Honduras es la que mayor rendimiento presenta, viéndose por el rendimiento de países como Guatemala y México (FHIA 2017). A pesar de ser la hortaliza de más importancia en el país, no recibe el mejor manejo fitosanitario posible ya que no todos productores tienen los recursos para poder sembrar variedades resistentes a las enfermedades que pueden atacar el cultivo, otros tienen los conocimientos necesarios para poder realizar diagnósticos de las enfermedades por ellos mismos, y agregando la gran incidencia de estas hace que ninguna plantación esté exenta de infectarse. La gran variedad de enfermedades a las que es susceptible el cultivo de tomate se debe a las altas poblaciones de diferentes patógenos como hongos, bacterias, virus y oomicetos (Bernal 2010), los cuales al infectar una planta suelen causar daños como lesiones, manchas, halos, clorosis y otros síntomas. Dependiendo de la gravedad pueden ocasionar pérdidas económicas, reducción de la producción, y en casos extremos pérdida total del cultivo, lo cual se puede prevenir o revertir con un manejo adecuado de la enfermedad. El éxito de todo manejo integrado de enfermedades consiste en hacer un diagnóstico de estas en el momento en que aparecen los primeros síntomas, así poder tomar ventaja contra enfermedad y poder prevenir que esta continúe propagándose.

Considerando lo dicho anteriormente, se determinó que es necesario crear herramientas digitales que ayuden al agricultor a realizar diagnósticos de enfermedades, las cuales les permita determinar qué está atacando a su cultivo y qué la ocasiona. Con esto, determinar que tratamiento o medidas correctivas se va a usar, tratando de evitar una posible propagación hacia otras áreas del cultivo, y reducir el riesgo de sufrir mermas significativas de la calidad y el rendimiento final. Todo esto se ve reflejado al final del ciclo en la parte económica, al usar tratamientos de control se debe hacer de la mejor manera posible para evitar que las aplicaciones sean en vano, porque una mala aplicación puede ser incapaz de controlar el agente causal de la enfermedad. Hacer un mal diagnóstico lleva a usar tratamientos equivocados, permitiendo la propagación de la enfermedad, lo que hace necesario invertir más dinero en medidas correctivas.

En la actualidad, se han logrado muchos avances en el área de la agricultura, donde se han dado innovaciones que han revolucionado este sector, tienen como objetivo común optimizar en su máxima expresión la producción de cultivos, incluyendo la parte de productividad y también la parte fitosanitaria. Para ello se han desarrollado herramientas que sirven para realizar diagnósticos de enfermedades y sus síntomas en diferentes cultivos, algunas de estas son aplicaciones móviles desarrolladas usando mecanismos de inteligencia artificial, que poseen un rendimiento y funcionalidad de gran nivel, además estas son accesibles desde cualquier lugar mediante un dispositivo móvil lo que les da cierto grado de superioridad sobre otros métodos usados. En la actualidad este tipo de herramientas son escasas pero las existentes son muy eficientes, por ejemplo, “Plantix”, es una aplicación de asesoramiento de cultivos para los agricultores, fue creada por la empresa alemana “Progressive Environmental & Agricultural Technologies” (PEAT) la cual se dedica al desarrollo de inteligencia artificial, su propósito es revolucionar el sector agrícola con una aplicación totalmente capaz de detectar plagas y enfermedades en cultivos también brindan consejos sobre cómo tratarlas (PEAT 2020).

Se decidió implementar algoritmos de inteligencia artificial mediante redes neuronales artificiales, las cuales son entrenadas a través de entradas que se obtienen a partir de escenarios externos o internos del sistema las cuales se reproducen por valores conocidos, asignados individualmente a cada uno de estos, para realizar una programación adecuada se necesita de una base de datos con imágenes de los daños ocasionado en hojas por las enfermedades también de hojas sanas, donde la red neuronal pueda obtener la información necesaria para su entrenamiento haciendo muchas repeticiones hasta que aprenda a hacer un diagnóstico eficaz. Las redes neuronales son parte de la inteligencia artificial, también han brindado resultados satisfactorios en las tareas de clasificación necesarias para la identificación de síntomas en condiciones de campo (Rivas y Mazón 2018).

Las redes neuronales, para sus entrenamientos y diagnósticos, se auxilian de una base de datos donde se recopilaron imágenes de las enfermedades seleccionadas para el proyecto, en las que se pueda apreciar claramente los daños causados en el follaje de la planta, deben tener alta resolución para que el programa no tenga problemas al analizarlas, luego hace una clasificación y descarte de aquellas que no encajen con los patrones de colorimetría que corresponden a hojas infectadas o sanas, patrones que sirven para obtener códigos de programación usando el modelo de color “Red, Green, Blue” (RGB) por sus siglas en inglés, este se usa para diseños digitales de “websites”, aplicaciones y banners, esos códigos se deben traducir a lenguaje de programación para implementarlos en la red neuronal. La implementación de los códigos y entrenamiento de la red se hizo en la plataforma de scripting llamada “Python”, es un lenguaje de programación de código libre de alto rendimiento, usado para programación disponible en diferentes sistemas operativos.

Se recopilaron imágenes de las enfermedades que fueron seleccionadas de acuerdo con la expresión de sus síntomas, considerando aquellos que permiten apreciar coloraciones anormales, esto es importante porque favorecen el entrenamiento del programa ya que así puede apreciar mejor los daños de las hojas, tales como cambios de color, halos, clorosis, canchales, mosaicos, cribados entre otros. Además de tomar en consideración el tipo de síntomas en las enfermedades, también se consideró la regularidad o frecuencia con que atacan al cultivo.

El desarrollo de programas de este tipo, que son escasos a nivel mundial, causan un impacto positivo en el sector agrícola, una red neuronal puede ser implementada en la programación de herramientas digitales como aplicaciones móviles similares a “Plantix”, que ayuda a productores con asesoramiento para sus cultivos permitiéndoles hacer diagnósticos de las enfermedades y así puedan tomar medidas correctivas, serían beneficiados productores sin los recursos económicos para contratar especialistas que puedan hacer monitoreo en sus cultivos o aquellos que no tienen el conocimiento necesario para identificar una enfermedad. Por otra parte, la elaboración de la base de datos es muy importante, al permitir que exista una fuente de información regional sobre las enfermedades más comunes del tomate, que pueda ser usada en futuras investigaciones. Los objetivos que se establecieron para el proyecto realizado considerando diferentes factores fueron:

- Elaborar una base de datos mediante la recolección de imágenes que permitan observar los síntomas expresados por enfermedades del follaje en la planta de tomate.
- Realizar diagnósticos de enfermedades en el follaje del cultivo de tomate a partir de una red neuronal completamente funcional.

2. METODOLOGÍA

El análisis divide el problema en tres etapas definidas como lectura y preparación de datos, donde se toma la base de datos de las hojas previamente etiquetadas con cierta enfermedad o como sanas. Luego se procede a la fase de extracción de ruido, donde la imagen es procesada y filtrada para eliminar cualquier perturbación que no pertenezca a la enfermedad requerida (región de interés), tomando sus valores de RGB y procesándolos de manera que se obtenga un valor numérico único para cada imagen, pudiendo recuperar su forma original siempre que fuera necesario. Una vez que se tiene bien definido el paso anterior, se procede a la aplicación del algoritmo de inteligencia artificial, para entrenar a la red neuronal con los pesos de cada imagen etiquetada como sana o infectada con determinada enfermedad. En este paso, se realizan múltiples cálculos hasta que se obtiene una condición de desempeño, que no es más que la relación entre la probabilidad de la comparación de la predicción con el conjunto de datos previamente etiquetados para, de esta manera, obtener la predicción exacta para una enfermedad específica.

Una vez realizado el análisis, se realiza una validación del procedimiento, que consiste en: teniendo entrenada la red neuronal con los pesos correspondientes a individuos sanos e infectados, se le suministra una muestra con cierta cantidad de imágenes sin etiqueta, para que el algoritmo decida a qué clase pertenece cada una, sabiendo previamente el usuario a qué categoría pertenece cada imagen. Esta es una prueba a ciegas, puesto que el programa no conoce ninguna información a priori acerca de las imágenes, más que el entrenamiento que ha recibido con imágenes similares. Como fue mencionado, este proyecto demanda una enorme cantidad de información, en este caso imágenes para que las redes tengan información con la cual aprender a identificar las enfermedades, por lo que se creó una base de datos con imágenes de los daños en el follaje por eso se hizo la creación de una base de datos.

Base de datos

En informática, una base de datos es aquella colección de datos que se encuentran interrelacionadas de una u otra manera y que es necesario poseer un conjunto de diferentes programas con los que se pueda acceder a ellos (Ricardo 2009). En palabras sencillas, se podría decir que una base de datos es toda información que se encuentra acerca de algo en específico, que esta agrupada o estructurada para tener un acceso sencillo a esta (Gómez 2013). Una base de datos puede representar aspectos de la vida real, que puedan ser de interés para usuarios que están en busca o almacenando información o datos con un propósito específico (Gutiérrez Díaz 2009).

Esta base de datos de imágenes de las afecciones causadas por patógenos debe ser lo suficientemente extensa para que la red pueda tener información suficiente para realizar sus comparaciones y clasificaciones en los diagnósticos. Son necesarias 1,000 imágenes como mínimo por cada una de las enfermedades de esta manera tendrá suficiente información y arrojará un resultado más certero.

Para poder estructurar la red neuronal se necesitó de códigos de programación, los cuales se obtienen del escaneo de los espectros de color que las hojas presentan, para poder obtener estos códigos se usó el modelo de color RGB.

Modelo de color RGB (red, green, blue)

Es llamado así por sus siglas en inglés, este modo de color se utiliza para diseños digitales como “websites”, aplicaciones o banners. El modelo sirve para obtener patrones de códigos, los cuales se extraen con base en la colorimetría de los síntomas en las hojas afectadas escaneando las hojas y analizando los espectros de color que esta posee, luego de obtener los códigos de programación es importante traducirlo a un lenguaje de programación para que las redes neuronales puedan leerlos. Para la construcción del sistema de redes neuronales, donde se implementaron los códigos, también se sometió a entrenamientos y pruebas de diagnóstico con lo que aprendió a reconocer las enfermedades se usó el lenguaje de programación “Python”.

“Python”

Es un lenguaje de programación de alto nivel, orientado a objetos con funciones asociadas a extensas bases de datos y que tiene amplia agilidad algorítmica, además de poseer un alto nivel de semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con un tipado y enlaces dinámicos, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para ser usado como lenguaje de scripts o para conectar componentes existentes. La sintaxis simple y fácil de aprender del programa permite escribir códigos con alta legibilidad y por lo tanto reduce el costo de mantenimiento del programa (Marzal y Gracia 2014). El programa admite módulos y paquetes, lo que fomenta el modularidad del programa y la reutilización del código. El intérprete de “Python” y la extensa biblioteca estándar están disponibles en formato fuente o binario sin cargo para todas las plataformas principales y se pueden distribuir libremente (Challenge *et al.* 2014). La plataforma “Python” sirvió para la traducción a lenguaje de programación de los códigos obtenidos a través de la colorimetría, estos códigos se implementaron en el programa para estructurar la red neuronal.

Redes neuronales

Son una manera de hacer aprendizaje automático, en el cual una computadora aprende a realizar una tarea analizando ejemplos de entrenamiento donde, generalmente los casos reales se han etiquetado previamente. Un sistema de reconocimiento de objetos, por ejemplo, podría alimentar miles de imágenes etiquetadas de automóviles, casas, tazas de café, entre otros, y encontraría patrones visuales en las imágenes que se correlacionan consistentemente con etiquetas particulares.

Una red neuronal consta de miles o incluso millones de nodos de procesamiento simples que están densamente interconectados. La mayoría de las redes neuronales actuales están organizadas en capas de nodos, en las cuales los datos se mueven a través de ellas en una sola dirección. Un nodo individual puede estar conectado a varios nodos en la capa debajo de él, desde donde recibe datos, y a varios nodos en la capa superior, a los que envía datos. Es decir, aprende desde las capas inferiores e informa a las capas superiores, donde igual se aprende y se envía retroalimentación a la capa inferior en un proceso continuo hasta que se alcanza una condición de desempeño óptima, lo cual indica que se tiene alta confiabilidad en la predicción realizada.

A cada una de sus conexiones entrantes, un nodo asignará un número conocido como peso. Cuando la red está activa, el nodo recibe un elemento de datos diferente, un número diferente, sobre cada una de sus conexiones y lo multiplica por el peso asociado. Luego suma los productos resultantes,

dando un solo número. Si ese número está por debajo de un valor umbral, el nodo no pasa datos a la siguiente capa. Si el número excede el valor de umbral, el nodo se dispara, lo que en las redes neuronales actuales generalmente significa enviar el número, la suma de las entradas ponderadas, a lo largo de todas sus conexiones salientes.

Cuando se entrena una red neuronal, todos sus pesos y umbrales se establecen inicialmente en valores aleatorios. Los datos de entrenamiento se envían a la capa inferior, la capa de entrada, y pasan a través de las capas siguientes, multiplicándose y sumando de maneras complejas, hasta que finalmente llegan, radicalmente transformadas, a la capa de salida. Durante el entrenamiento, los pesos y los umbrales se ajustan continuamente hasta que los datos de entrenamiento con las mismas etiquetas arrojan resultados similares de manera consistente.

Al tratarse de tareas como las de realizar cálculos numéricos, y realizar las comparaciones entre un cerebro humano y una computadora digital, se encuentra que el cerebro humano posee muchas ventajas sobre la computadora (Krogh 2008). El cerebro humano es capaz de reconocer un rostro humano o, por ejemplo, una habitación llena de objetos a poca luz aun si esta es vista de lado, puede entender fácilmente el habla, aun si es una voz de una persona desconocida, por ende, aun con muchos años de investigación una computadora está lejos de poder funcionar a ese nivel (Krogh 2008), con esto hace referencia a la importancia de realizar un proceso de codificación eficiente ya que es una inteligencia artificial no quiere decir que es 100% fiable, pues para al momento de brindar sus resultados esta se base en los porcentajes de proximidad que las imágenes presentan cuando fueron analizadas, esto significa que los procesos de entrenamiento y programación deben ser completamente exitosos.

3. RESULTADOS Y DISCUSIÓN

El resultado final del proyecto fue completamente satisfactorio ya que el sistema de redes neuronales es totalmente capaz de diagnosticar dos enfermedades de las más importantes en la región, que representan un riesgo económico para las zonas de producción a nivel nacional. Dichas enfermedades expresan una sintomatología muy bien marcada en la zona foliar de la planta de tomate, síntomas como los mencionados anteriormente como ser coloraciones anormales, halos necróticos, pudriciones, canchales entre otros, basándonos principalmente en las alteraciones de la coloración en las hojas se procedió a realizar un análisis de los espectros de color con el modelo de color RGB para la obtención de los códigos de programación que fueron usados para codificar la red neuronal.

Los procesos de clasificación, comparaciones y descarte durante los entrenamientos que la red neuronal hizo, fue utilizando las imágenes de la base de datos, que se conformó de más o menos 1,000 imágenes por cada una de las ocho enfermedades que fueron seleccionadas (Cuadro 1), estas enfermedades pertenecen a diferentes especies de patógenos como las que son causadas por hongos de las cuales son cuatro, producida por bacterias se seleccionó uno, ocasionadas por virus dos, y una última la cual es producida por Oomicetos, además en la base de datos existe una cantidad similar de hojas completamente sanas que sirven para que la red también pueda diferenciar entre aquellas que no están infectadas. Durante los entrenamientos la red neuronal analizó cada una de las imágenes de la base de datos en diferentes procesos donde, al finalizar cada uno esta descartaba aquellas que no coincidían con ninguna de las clases que se le otorgaron.

Desarrollar este tipo de proyectos de programación obviamente requiere de un trabajo muy extenso y complicado, donde se necesita de extensas bases de datos con imágenes suficientes para cada una de las enfermedades, donde cada una de las enfermedades representadas con imágenes obtiene un valor específico único, valor que es muy difícil de obtener debido al gran número de imágenes, es por eso que momentáneamente se logró únicamente hacer la programación adecuada para el diagnóstico de dos enfermedades que son, el tizón temprano (*Alternaria solani* L.) y de la mancha foliar por septoria (*Septoria lycopersici*). Las redes neuronales fueron entrenadas, sometidas a diversos procesos de validación como métodos de aprendizaje con lo que la red fue adquiriendo la capacidad de realizar el diagnóstico para cada una, determinando la eficiencia en diferentes en los grados porcentuales de validación obtenidos, para brindar el resultado toma el porcentaje más alto que obtuvo en las repeticiones de validación.

Para poder obtener los grados porcentuales de validación esta evaluó una cantidad indeterminada de imágenes de las lesiones, que nuevamente fue sometiendo a comparaciones y clasificaciones con las demás enfermedades de la base de datos y de las hojas sanas, de esta manera puede diferenciar aquellas que no presentan ninguna afección, luego de finalizado el proceso brinda el resultado de acuerdo con la información que obtuvo. Una vez creados los parámetros para que la identificación de una enfermedad sea correcta, es cuestión de realizar unos pequeños ajustes para ir aumentando la capacidad de la red neuronal, pero para poder hacerlo en trabajar con un número más grande de enfermedades, pero la limitante verdadera es encontrar la cantidad de imágenes que se necesitan. Si se quisiera trabajar con un patrón de síntomas un tanto más complicados se requiere de otro tipo de algoritmo y metodología para poder hacer más eficiente el programa.

En el Cuadro 1 se representan las enfermedades de las que se recolectaron imágenes en la base de datos, cabe recalcar que de estas ocho enfermedades en total el sistema de redes neuronales cuenta con la capacidad de diagnosticar solo dos, esto es importante porque se posee información para continuar aumentando la capacidad de diagnóstico, tomando como base lo realizado hasta el momento.

Cuadro 1. Enfermedades seleccionadas de las que se recopilamos imágenes en la base de datos, que causan daño foliar con una expresión clara de síntomas en la planta de tomate.

Tipos de enfermedad	Nombre común	Agente causal
Enfermedades causadas por hongos	Tizón Temprano	<i>Alternaria solani</i>
	Mancha foliar por septoria	<i>Septoria lycopersici</i>
	Mancha anillada del tomate	<i>Corynespora cassiicola</i>
	Moho gris del tomate	<i>Passalora fulva</i>
Enfermedades causadas por bacterias	Mancha bacteriana	<i>Xanthomonas campestris pv vesicatoria</i>
Enfermedades causadas por virus	Mosaico del Tomate	Virus del mosaico del tomate
	Rizado amarillo del Tomate	Virus del rizado amarillo del tomate
Enfermedades causadas por Oomicetos	Tizón Tardío	<i>Phytophthora infestans</i>

Especificaciones de las enfermedades que pueden ser diagnosticadas por el sistema de redes neuronales, dentro de las cuáles se mencionan algunos de los síntomas más frecuentes de estas enfermedades, los métodos de diseminación y propagación que utilizan los patógenos para infectar plantas sanas y también las prácticas de manejo más recomendadas para cada una.

Tizón temprano, *Alternaria* (Early blight) - *Alternaria solani*

Sintomatología presentada por el tizón temprano. Los primeros síntomas de la enfermedad se comienzan a presentar en las hojas más viejas y maduras de la planta, presentando lesiones de color amarillo, pardo oscuro o negras, con bordes irregulares (Bernal 2010). Desarrollo de anillos concéntricos que presentan un halo clorótico, este síntoma se puede notar inclusive en los tallos de la planta, al verse afectados los tallos llegan a desarrollarse lesiones en forma de anillos, hendiduras, en formas de óvalos que llegan a presentar coloraciones marrones, negras, de blanco a grisáceo en el centro de las lesiones (Rybak y Rybak 2018; Bernal 2010). Alrededor de las heridas se presentan halos con coloraciones amarillas. En algunos casos las lesiones llegan a ser tan numerosas que se terminan uniendo, causando una necrosis total de la hoja iniciando en los bordes de la hoja y continúa expandiéndose hacia el centro de la hoja (Rosa 2007). Los frutos también se ven afectados donde la enfermedad se presenta con lesiones o heridas color café, donde también se forman anillos secos de gran tamaño en la superficie cerca del pedúnculo, en el centro de los anillos

se puede observar perfectamente una esporulación grisácea (PlantVillage 2020). Cuando la enfermedad infecta severamente a la planta produce una defoliación parcial o total, lo que termina exponiendo los frutos al sol causando una reducción del rendimiento de hasta un 30 a 50% (VISAR 2015).

Propagación y diseminación. Se puede dispersar por acción del viento y por la lluvia, también se puede dispersar por salpicadura de la lluvia, este puede sobrevivir en tejidos enfermos del suelo, usar material de siembra contaminado, las herramientas y maquinaria de trabajo que usan los obreros (Lopez 2017). Otras especies de la familia de las solanáceas infectadas puede ser una fuente de inóculo, este puede sobrevivir y estar presente en el suelo, también puede propagarse por los obreros que realizan las actividades.

Control y manejo de la enfermedad. La aplicación de fungicidas es de mucha ayuda para el control de la enfermedad (Bernal 2010), solarizar el suelo donde se va a sembrar ayuda a eliminar el inóculo, es importante asegurarse de voltear el suelo uniformemente para que el control sea más eficaz. Se pueden usar variedades actualmente que son resistentes a esta enfermedad, esto sin duda alguna ayuda significativamente el control (Lopez 2017), respetar el distanciamiento entre plantas y entre hileras de siembra adecuado para hacer que exista aireación en el cultivo y de esta manera la humedad relativa baje para evitar la propagación de la enfermedad. Prácticas de control como la poda de las que están infectadas ayuda a reducir el foco de infección y además de esto permite una buena circulación del aire, otras prácticas culturales como el desmalezado, un manejo adecuado del material vegetativo infectado retirándolo del área cultivada y de esta manera reducir las fuentes de infección (López 2017). El estaquillado permite que las plantas puedan permanecer completamente erectas y organizadas permitiendo un espaciamiento adecuado del cultivo y que el aire tenga una buena circulación reduciendo, la humedad relativa y de la planta en general para que el hongo no tenga un ambiente propicio para su desarrollo (Bernal 2010). Aplicar el riego en horas de la mañana puede ser muy importante también, para que el exceso de humedad que pueda existir en el cultivo puede disminuir durante las horas del día (Luna 2019). Para la aplicación de fungicidas es importante que estos contengan ingredientes activos como: el clorotalonil, mancozeb, pyraclostrobin, una combinación pyraclostrobin + boscalid, flutriafol, trifloxystrobin (Luna 2019).



Figura 1. Sintomatología del Tizón temprano en el tomate: A) Hojas con amarillamiento y halos necróticos. B) Tallos con presencia de halos necróticos y manchas grisáceas. C) Fruto con lesión en el pedúnculo que provoca el desprendimiento.

Fuente: PlantVillage 2020 A y B, IFAS/UF Florida 2013 C.

Mancha foliar por septoria (Septoria leaf spot) - *Septoria lycopersici*

Sintomatología presentada por la mancha foliar por septoria. El inicio esta enfermedad se presenta con numerosas manchas acuosas pequeñas, en la parte foliar de la planta específicamente en las hojas, viéndose afectadas las hojas más viejas, las cuales también se encargan de la diseminación del patógeno, la planta suele ser más susceptible después de que los frutos estén cuajados (Bernal 2010; INTAGRI 2015). Las manchas desarrolladas presentan una coloración marrón claro o negra, que presentan una forma circular, por lo general alrededor de estas manchas se desarrollan halos concéntricos de color amarillo o blanco (Bernal 2010; INTAGRI 2015), en ocasiones los ataques de esta enfermedad pueden llegar a ser críticas, donde se presenta defoliación de la planta porque los síntomas más graves de esta enfermedad incluyen el secado y caída de las hojas (Bernal 2010). La enfermedad muchas veces suele ser confundida por la *Alternaria*, ya que presentan síntomas similares, pero llegan a diferenciarse por el tamaño y las manchas por septoria presentan halos alrededor de las manchas y son de menor tamaño que las del tizón temprano (Bernal 2010; INTAGRI 2015), generalmente en el centro de las manchas se presentan unas estructuras pequeñas color negro que reciben el nombre de picnidios, que son las estructuras distintivas de esta (Bernal 2010). Como consecuencia de la defoliación de la planta, los frutos al quedar expuestos al sol causan quemaduras en estas, lo que a su vez trae como consecuencia una reducción del rendimiento general del cultivo (Bernal 2010), cuando se encuentra en un ambiente con alta humedad, y la enfermedad se encuentra en un estado avanzado se comienzan a desarrollar los cuerpos fructíferos del hongo conocidos como picnidios estos al madurar en su totalidad producen muchas esporas, al liberarse estas esporas tienden a penetrar las hojas usando como acceso las estomas de las hojas (Bernal 2010; INTAGRI 2015). Este hongo por lo general no puede vivir mucho tiempo en el suelo al aire libre, pero si puede sobrevivir en el suelo gracias a los restos de otras plantas infectadas que se encuentran incorporadas al suelo de igual manera puede sobrevivir en las malezas de la familia de las solanáceas (Bernal 2010), se desconoce si la enfermedad tiene la capacidad de transmitirse a través de la semilla (Bernal 2010), el tallo, peciolo y el cáliz presentan lesiones que tienen una forma alargada, los cuales también desarrollan picnidios en el centro de estas y tienden a doblar los tallos (INTAGRI 2015).

Propagación y diseminación. Las esporas se pueden diseminar por acción del viento el que lleva consigo las esporas del hongo a otras partes del cultivo, por el agua de riego que es transportada por el sistema de riego, también cuando llueve se puede dispersar por la salpicadura de lluvia, en labores de manejo de cultivo al usar herramientas y equipo de trabajo que están contaminadas con el inoculo del hongo (Bernal 2010). El intervalo de temperatura para el desarrollo del hongo varía entre 15 y 27 °C, siendo 25 °C la temperatura para un crecimiento óptimo (Plantix 2020).

Control y manejo de la enfermedad. Usar cultivares que sean resistentes a esta enfermedad es uno de los métodos de control más importantes de esta enfermedad en la actualidad en el pasado era un poco más difícil puesto que no existían este tipo de cultivares, (Bernal 2010), realizar acciones de control de malezas en las zonas de producción, la eliminación de restos de cultivos afectados anteriormente es prácticas que pueden ayudar a reducir los focos de infección del hongo (López 2017). Aplicaciones de fungicidas a base de mancozeb, clorotalonil, así como los que pertenecen a la familia de las strobilurinas, son muy útiles para el control de la enfermedad (Lopez 2017), la rotación de cultivos también es una de las practicas recomendadas, que se debe de realizar mínimo durante 3 años, ayudara a disminuir el inoculo (Semini 2020).



Figura 2. Síntomas de la mancha foliar por Septoria. A) Lesión en la hoja donde se alcanza a notar el centro blanco o gris, así como también los cuerpos fructíferos o esporulaciones fúngicas. B) Rama que presenta varias manchas causadas por la septoria. C) Lesión de la mancha por septoria en la hoja de tomate.

Fuentes: PlantVillage 2020.

Resultados de la identificación con la red neuronal

Se aplicó el algoritmo utilizando imágenes etiquetadas correctamente (Figura 3) que luego serían utilizadas para entrenar la red neuronal, como imágenes de prueba (Figura 4), que serían utilizadas para probar la red antes de ser entrenada y después de dicho entrenamiento, con el objetivo de comparar el desempeño con aprendizaje previo y en ausencia de este (Figura 5). Después del entrenamiento, se aplicó el algoritmo a 60 imágenes de prueba, para que la red neuronal decidiera clasificar a las imágenes con la etiqueta adecuada, de acuerdo con los pesos obtenidos en el entrenamiento. Se obtuvo que en la totalidad de los casos la red neuronal consiguió identificar correctamente a las imágenes de acuerdo con su categoría (Figura 6), lo cual representa un rendimiento elevado y una eficiencia prácticamente impecable.



Figura 3. Aplicación del algoritmo en las imágenes etiquetadas.

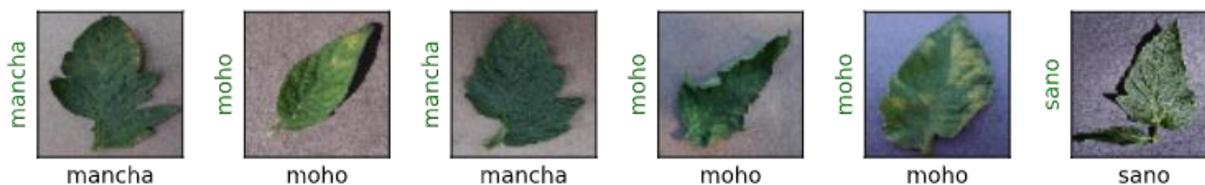


Figura 4. Prueba de la efectividad del método utilizado para los diagnósticos.

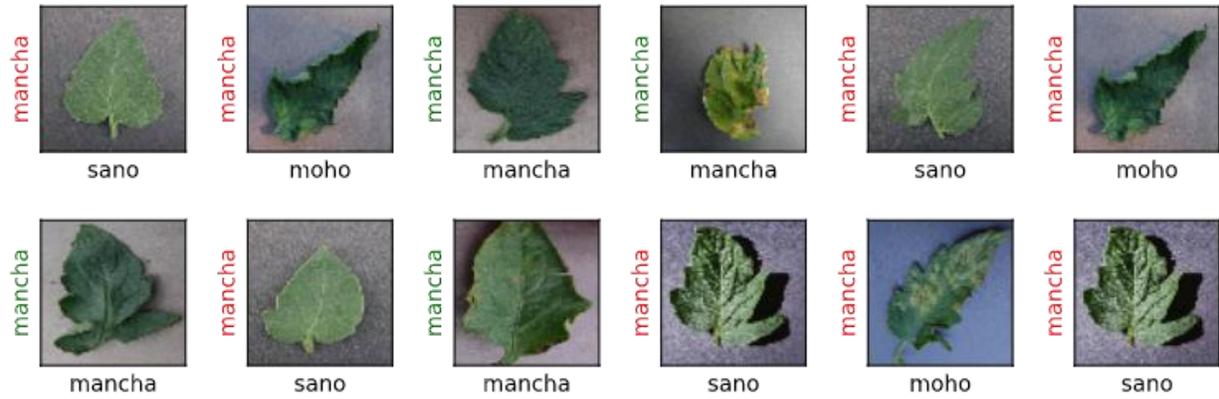


Figura 5. Comparación del desempeño de una red entrenada y otra sin entrenar.

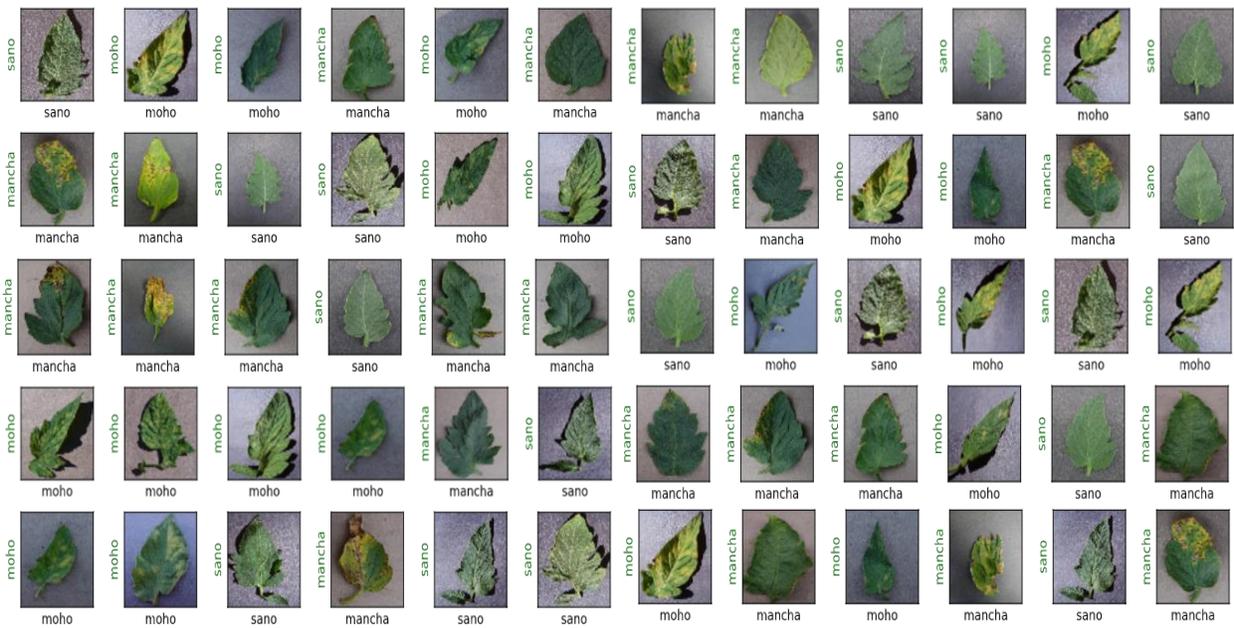


Figura 6. Prueba final de diagnóstico después de los entrenamientos y comprobada la efectividad del método que se utilizó.

4. CONCLUSIONES

- La base de datos cuenta con imágenes de ocho enfermedades que expresan síntomas en el follaje de la planta de tomate, de las cuales dos fueron programadas en la red neuronal para ser diagnosticadas.
- La red neuronal construida momentáneamente tiene la capacidad de poder realizar el diagnóstico con un alto grado de eficiencia de dos enfermedades, siempre y cuando la expresión de los síntomas sea clara y no se vea afectada por otro factor determinante.

5. RECOMENDACIONES

- Con las imágenes que posee la base de datos, realizar la programación en la red neuronal de las seis enfermedades restantes.
- Completar la base de datos con imágenes de las enfermedades más comunes que afectan al cultivo de tomate.
- Trabajar en la programación de la red neuronal para que esta tenga la capacidad de trabajar con síntomas más complejos, o que se expresen en otras áreas de la planta.
- Incorporar la red neuronal codificada en el prototipo de aplicación móvil “Zolycop” para que esta tenga una mayor capacidad de diagnóstico de enfermedades en el tomate, de la cual puedan usar los productores.

6. LITERATURA CITADA

- Bernal R. 2010. Enfermedades de tomate (*Lycopersicum esculentum* Mill.) en invernadero en las zonas de Salto y Bella Unión. Montevideo: Unidad de comunicación y transferencia de tecnología de INIA. ISBN: 9789974382831.
- Challenger I, Díaz Y, Becerra R. 2014. El lenguaje de programación Python, Ciencias Holguín. Centro de información y gestión tecnológica de Santiago de Cuba Holguín, Cuba. 20(2): 1-13.
- [FHIA] Fundación Hondureña de Investigación Agrícola. 2017. Informe técnico 2017, programa de hortalizas. [consultado el 24 de sep. de 2020] http://www.fhia.org.hn/downloads/informes_tecnicos/Informe_Tecnico_2017_Programa_de_Hortalizas.pdf
- Fornais G. 2007. Conjunto tecnológico para la producción de tomate, variedades y su selección. Puerto Rico: Universidad de Puerto Rico, recinto universitario de Mayagüez. 6 p; [consultado el 25 de jul. de 2020] <https://www.upr.edu/eea/wp-content/uploads/sites/17/2016/03/TOMATE-Variedades-y-su-Selecci%C3%B3n-v2007.pdf>
- Gómez M. 2013. Notas del cursos, bases de datos. 1ª ed. Editado por la Universidad Autónoma Metropolitana. México: Publicista Mexicana S.A. de C.V. 197 p. ISBN: 9786074778809
- Gutiérrez Díaz A. 2009. Bases de datos: Clav-MIS 308. Centro Cultural Ítaca S.C. México. [consultado el 2 de oct. de 2020] <https://www.aiu.edu/cursos/base%20de%20datos/pdf%20leccion%201/lecci%C3%B3n%201.pdf>
- [IFAS/UF] University of Florida, Institute of Food and Agricultural Sciences. 2013. Black mold. [consultado el 6 de nov. de 2020] <https://plantpath.ifas.ufl.edu/u-scout/tomato/black-mold.html>
- [INIA] Instituto Nacional de Investigación y Agropecuaria. 2012. Resultados experimentales en sanidad de tomate y morrón. Uruguay. [consultado el 14 de sep. de 2020] <http://www.inia.uy/Publicaciones/Documentos%20compartidos/18429250912110347.pdf#page=31>
- [INTAGRI] Instituto para la Innovación tecnológica en la Agricultura. 2015. Enfermedades foliares en Tomate. [consultado el 12 de sep. de 2020] <https://www.intagri.com/articulos/fitosanidad/enfermedades-foliares-en-tomate>
- Krogh A. 2008. What are artificial neural networks? Nature biotechnology. Nature Publishing Group (NPG) [consultado el 6 de nov. de 2020] 26(2): 195-197. Inglés. <http://people.binf.ku.dk/krogh/publications/pdf/Krogh08.pdf>
- López L. 2017. Manual técnico del cultivo de tomate (*Solanum lycopersicum*). Instituto Nacional de Innovación y Transferencia en Tecnología Agropecuaria (INTA). San José, Costa Rica. 126 p. ISBN: 9789968586276

- Luna Espinoza EA. 2019. Elaboración de un prototipo de aplicación electrónica para la identificación de las enfermedades más comunes del cultivo de tomate (*Solanum lycopersicum* L.) en Honduras. Honduras: Escuela Agrícola Panamericana, Zamorano. 51 p; [consultado el 2 de sep. de 2020]. <https://bdigital.zamorano.edu/bitstream/11036/6701/1/CPA-2019-T046.pdf>
- Marzal A, Gracia I. 2014. Introducción a la programación con Phyton. Colección Sapientia, 23. Universitat Jaume. Castellón, España. 427 p. ISBN: 9788469258699.
- Plantix. 2020. Manchas foliares por Septoria. [consultado el 15 de sep. de 2020] <https://plantix.net/es/library/plant-diseases/100152/septoria-leaf-spot>
- Plantvillage. 2020. Tomato. [consultado el 12 de sep. de 2020] https://plantvillage.psu.edu/topics/tomato/infos/diseases_and_pests_description_use_propagation
- Rivas W, Mazón B. 2018. Redes neuronales artificiales aplicadas al reconocimiento de patrones. 1ª ed. Machala: editorial UTMACH. 35 p. ISBN: 978-9942-24-100-9.
- Ricardo CM. 2009. Bases de datos. 1er ed. México: McGraw-Hill Interamericana Editores. 642 p. ISBN: 9789701072752
- Rybak MA, Rybak MR. 2018. Principales patógenos diagnosticados en misiones según la especie y breve análisis de la complejidad del Sistema fitosanitario hortícola. Estación experimental agropecuaria Cerro Azul, Misiones Argentina. [consultado el 22 de ago. de 2020] https://inta.gob.ar/sites/default/files/informe_tecnico_104.pdf
- Seminis. 2020. Mancha de la hoja (Septoria). México. [consultado el 6 de nov. de 2020] <https://www.seminis.mx/recursos/guia-de-enfermedades/tomates/cercospora-leaf-spot/>
- [VISAR] Viceministerio de Sanidad Agropecuaria y Regulaciones. 2015. Manual de plan de manejo integrado de enfermedades del tomate en Guatemala (*Solanum lycopersicum* L.). [consultado el 5 de sep. de 2020] <http://proyectoadaintegracion.minex.gob.gt/ada/docs/MINISTERIO%20DE%20AGRICULTURA%20GANADERIA%20Y%20ALIMENTACION/MANUAL%20TOMATE.pdf>

7. ANEXOS

Anexo 1. Código de implementación.

```
import os
import shutil
import random
import torch
import torchvision
import numpy as np
from PIL import Image
from matplotlib import pyplot as plt
torch.manual_seed(0)
enfermedades = ['sano', 'moho', 'mancha']
directorio_ppal = 'Tomato'
directorio_datos = ['Tomato__healthy', 'Tomato__Leaf_Mold', 'Tomato__Bacterial_spot']
if os.path.isdir(os.path.join(directorio_ppal, directorio_datos[1])):
    os.mkdir(os.path.join(directorio_ppal, 'test'))
    for i, d in enumerate(directorio_datos):
        os.rename(os.path.join(directorio_ppal, d), os.path.join(directorio_ppal, enfermedades[i]))
        for c in enfermedades:
            os.mkdir(os.path.join(directorio_ppal, 'test', c))
            for c in enfermedades:
                figuras = [x for x in os.listdir(os.path.join(directorio_ppal, c))
                    if x.endswith('JPG')]
                selected_figuras = random.sample(figuras, 30)
                for image in selected_figuras:
                    source_path = os.path.join(directorio_ppal, c, image)
                    target_path = os.path.join(directorio_ppal, 'test', c, image)
                    shutil.move(source_path, target_path)
class DatosTomate(torch.utils.data.Dataset):
    def __init__(self, directorios_imagenes, transformar_imagen):
        def obtener_imagenes(clase_enfermedad):
            figuras = [x for x in os.listdir(directorios_imagenes[clase_enfermedad])
                if x.endswith('JPG')]
            print(f'Fueron encontrados {len(figuras)} ejemplos de la clase "{clase_enfermedad}".')
            return figuras
```

Continuación Anexo 1

```
self.figuras = {}
self.enfermedades = ['sano', 'moho', 'mancha']
for clase_enfermedad in self.enfermedades:
    self.figuras[clase_enfermedad] = obtener_imagenes(clase_enfermedad)

    self.directorios_imagenes = directorios_imagenes
self.transformar_imagen = transformar_imagen
def __len__(self):
    return sum([len(self.figuras[clase_enfermedad]) for clase_enfermedad in self.enfermedades])
def __getitem__(self, index):
    clase_enfermedad = random.choice(self.enfermedades)
    index = index % len(self.figuras[clase_enfermedad])
    image_name = self.figuras[clase_enfermedad][index]
    image_path = os.path.join(self.directorios_imagenes[clase_enfermedad], image_name)
    image = Image.open(image_path).convert('RGB')
    return self.transformar_imagen(image), self.enfermedades.index(clase_enfermedad)
transformacion_entrenamiento = torchvision.transforms.Compose([
    torchvision.transforms.Resize(size=(224, 224)),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])])
transformacion_test = torchvision.transforms.Compose([
    torchvision.transforms.Resize(size=(224, 224)),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])])
entrenamiento_dir = {
    'sano': 'Tomato/sano',
    'moho': 'Tomato/moho',
    'mancha': 'Tomato/mancha'}
entrenamiento_datos = DatosTomate(entrenamiento_dir, transformacion_entrenamiento)
prueba_dir = {
    'sano': 'Tomato/test/sano',
    'moho': 'Tomato/test/moho',
```

Continuación Anexo 1

```
'mancha': 'Tomato/test/mancha'}
prueba_datos = DatosTomate(prueba_dir, transformacion_test)
batch_size = 6
dl_train = torch.utils.data.DataLoader(entrenamiento_datos, batch_size=batch_size, shuffle=True)
dl_test = torch.utils.data.DataLoader(prueba_datos, batch_size=batch_size, shuffle=True)
print('Número de lotes de entrenamiento', len(dl_train))
print('Número de lotes de test', len(dl_test))
enfermedades = entrenamiento_datos.enfermedades
def mostrar_figuras(figuras, labels, preds):
    plt.figure(figsize=(8, 4))
    for i, image in enumerate(figuras):
        plt.subplot(1, 6, i + 1, xticks=[], yticks=[])
        image = image.numpy().transpose((1, 2, 0))
        mean = np.array([0.485, 0.456, 0.406])
        std = np.array([0.229, 0.224, 0.225])
        image = image * std + mean
        image = np.clip(image, 0., 1.)
        plt.imshow(image)
        col = 'green'
        if preds[i] != labels[i]:
            col = 'red'
        plt.xlabel(f'{enfermedades[int(labels[i].numpy())]}')
        plt.ylabel(f'{enfermedades[int(preds[i].numpy())]}', color=col)
    plt.tight_layout()
    plt.show()
figuras, labels = next(iter(dl_train))
mostrar_figuras(figuras, labels, labels)
figuras, labels = next(iter(dl_test))
mostrar_figuras(figuras, labels, labels)
resnet18 = torchvision.models.resnet18(pretrained=False, progress=True)
print(resnet18)
resnet18.fc = torch.nn.Linear(in_features=512, out_features=3)
loss_fn = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(resnet18.parameters(), lr=3e-5)
def show_preds():
    resnet18.eval()
    figuras, labels = next(iter(dl_test))
```

Continuación Anexo 1

```
    outputs = resnet18(figuras)
    _, preds = torch.max(outputs, 1)
    mostrar_figuras(figuras, labels, preds)
show_preds()
def train(epochs):
    print('Comenzando entrenamiento...')
    for e in range(0, epochs):
        print('='*20)
        print(f'Comenzando epoch {e + 1}/{epochs}')
        print('='*20)
        train_loss = 0.
        val_loss = 0.
        resnet18.train() # set model to training phase
        for train_step, (figuras, labels) in enumerate(dl_train):
            optimizer.zero_grad()
            outputs = resnet18(figuras)
            loss = loss_fn(outputs, labels)
            loss.backward()
            optimizer.step()
            train_loss += loss.item()
            if train_step % 20 == 0:
                print('Evaluando paso', train_step)
                accuracy = 0
                resnet18.eval() # set model to eval phase
                for val_step, (figuras, labels) in enumerate(dl_test):
                    outputs = resnet18(figuras)
                    loss = loss_fn(outputs, labels)
                    val_loss += loss.item()
                    _, preds = torch.max(outputs, 1)
                    accuracy += sum((preds == labels).numpy())
                val_loss /= (val_step + 1)
                accuracy = accuracy/len(prueba_datos)
                print(f'Validación de pérdida: {val_loss:.4f}, Exactitud:
{accuracy:.4f}')
            show_preds()
            resnet18.train()
            if accuracy >= 0.95:
                print('Condición de desempeño satisfecha, deteniendo..
.'
```

Continuación Anexo 1

```
        return
    train_loss /= (train_step + 1)
    print(f'Pérdida en entrenamiento: {train_loss:.4f}')
    print('Entrenamiento completo..')
train(epochs=1)
show_preds()
```

Anexo 2. Ejemplo de uso

Validando muestras... =====

Fueron encontrados 1560 ejemplos de la clase "sano".
Fueron encontrados 922 ejemplos de la clase "moho".
Fueron encontrados 2097 ejemplos de la clase "mancha".

Fueron encontrados 30 ejemplos de la clase "sano".
Fueron encontrados 30 ejemplos de la clase "moho".
Fueron encontrados 30 ejemplos de la clase "mancha".

Número de lotes de entrenamiento 764

Número de lotes de test 15

Comenzando entrenamiento... =====

Comenzando epoch 1/1 =====

Evaluando paso 0 Validación de pérdida: 1.2596, Exactitud: 0.3000

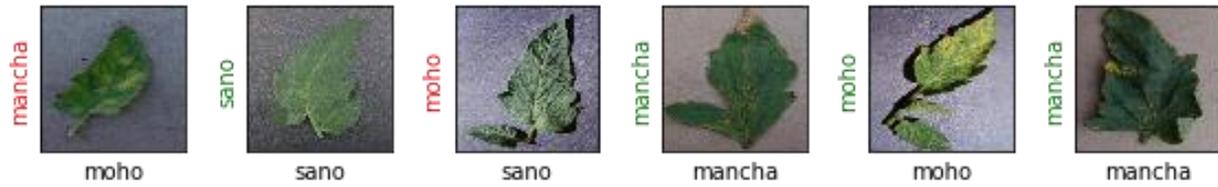


Evaluando paso 20 Validación de pérdida: 1.3358, Exactitud: 0.3444



Evaluando paso 40 Validación de pérdida: 0.7159, Exactitud: 0.8111

Continuación Anexo 2



Evaluando paso 60 Validación de pérdida: 0.3408, Exactitud: 0.8444

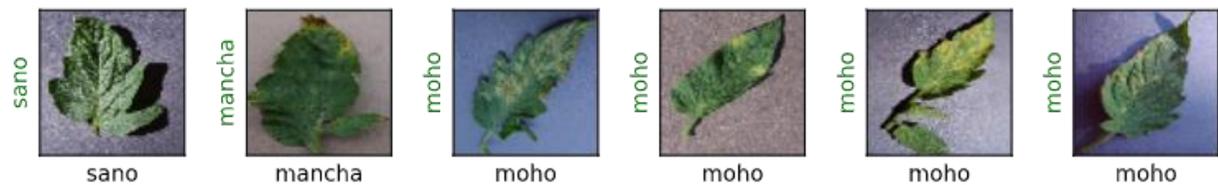


Evaluando paso 80 Validación de pérdida: 0.1782, Exactitud: 0.9667



Condición de desempeño satisfecha, deteniendo...

Identificando y evaluando en imagen específica...



Saliendo del programa.