

## COMPARACIÓN DE DOS MODELOS DE RED NEURONAL APLICADOS EN LA DETECCIÓN DE PATÓGENOS EN PLANTAS DE ORNATO

Ochoa-Sandoval Rafael Alejandro<sup>1</sup>, García-Díaz Noel<sup>1</sup>, Verduzco-Ramírez Jesús Alberto<sup>1</sup>, García-Virgen Juan<sup>1</sup>, Herrera-Morales José Román<sup>2</sup>

<sup>1</sup>Tecnológico Nacional de México / Instituto Tecnológico de Colima  
División de Estudios de Posgrado e Investigación  
Av. Tecnológico No. 1, Col. Liberación, Villa de Álvarez, Colima, C.P. 28976  
Teléfonos: (312) 312 6393, (312) 312 9920

<sup>2</sup>Universidad de Colima / Facultad de Telemática  
Avenida Universidad 333, Las Víboras, 28040 Colima, Col.  
Teléfono: (312) 316 1075  
{G2246015, ngarcia, averduzco, jgarcia}@colima.tecnm.mx, rherrera@ucol.mx

### RESUMEN.

La productividad de los cultivos de plantas ornamentales es afectada por plagas, enfermedades y otros factores. Actualmente, los métodos tradicionales de detección de patógenos son laboriosos, costosos, lentos y propensos a errores. El objetivo es clasificar imágenes para detectar patógenos en el follaje de las plantas ornamentales usando la inteligencia artificial, específicamente, las redes neuronales convolucionales. La comparación de rendimiento que se realizó en este trabajo podría aportar en la selección entre de un modelo pre-entrenado y un modelo de desarrollo propio (*GreenGuardNet*). Los resultados obtenidos fueron que tanto *MobileNetV2* y *GreenGuardNet* mostraron la misma exactitud 95.72%, en términos de precisión el modelo pre-entrenado mostró una ligera ventaja, pero utilizar un modelo pre-entrenado reduce el tiempo de desarrollo de manera significativa. Se concluye que la elección entre desarrollar un modelo o utilizar uno pre-entrenado depende de los objetivos y recursos disponibles, considerando la precisión y el tiempo de desarrollo.

Palabras clave: redes neuronales convolucionales, detección de patógenos, modelos pre-entrenados, *Python*.

### ABSTRACT.

The productivity of ornamental plant crops is affected by pests, diseases and other factors. Currently, traditional pathogen detection methods are laborious, expensive, time-consuming, and error-prone. The objective is to classify images to detect pathogens in the foliage of ornamental plants using artificial intelligence, specifically, convolutional neural networks. The performance comparison carried out in this work could contribute to the selection between a pre-trained model and a self-developed model (*GreenGuardNet*). The results obtained were that both *MobileNetV2* and *GreenGuardNet* showed the same accuracy 95.72%, in terms of precision the pre-trained model showed a slight advantage, but using a pre-trained model significantly reduces development time. It is concluded that the choice between developing a model or using a pre-trained one depends on the objectives and available resources, considering the precision and development time.

Keywords: convolutional neural networks, pathogen detection, pre-trained models, *Python*.

### 1. INTRODUCCIÓN

#### 1.1. Antecedentes

Latinoamérica tiene cualidades geográficas que permiten una alta productividad en cultivos de plantas ornamentales. México ocupa el tercer lugar a nivel mundial en términos de superficie de producción generando grandes oportunidades de empleos al sector primario [1]. Sin embargo, esta productividad, no está exenta de riesgos y pérdidas, ya que las plagas representan una amenaza significativa para la producción. Por lo anterior, es crucial contar con mecanismos eficaces para la detección oportuna de patógenos y tomar las medidas pertinentes que minimicen el impacto negativo. El método tradicional utilizado por los productores, es realizar inspecciones visuales periódicas en sus predios de cultivo. Este proceso es laborioso y puede introducir errores, debido a que se puede pasar por alto algún ejemplar que contenga muy pocos residuos de algún patógeno invasor, lo que incrementa el riesgo de errores en la detección de patógenos. Una vez que se encuentra una discrepancia entre las hojas y tallos de las plantas, el productor determina qué tipo de plaga es, basándose en su juicio experto que está sujeto a errores. Otro método utilizado para la detección de patógenos, es la utilización de pruebas serológicas en laboratorio, como *ELISA* (*Enzyme-Linked Immunosorbent Assay*), que utiliza anticuerpos para buscar una proteína específica que indica la presencia de un virus vegetal [2]. La realización de esta prueba puede tardar hasta 45 días debido a la necesidad de cultivar el virus [3].

Los productores tienen a su disposición varios métodos para detectar e identificar patógenos en sus cultivos. No obstante, enfrentan desafíos importantes, como el tiempo requerido para realizar pruebas de laboratorio. Además, estas no son adecuadas para la detección temprana de patógenos, sino que confirman la presencia de plagas una vez que la infestación ya ha ocurrido. Lo oportuno de la detección de las plagas es crucial, ya que el retraso en el tratamiento de los cultivos puede resultar en pérdidas significativas. La limitación económica también es un factor importante, ya que no todos los productores pueden permitirse invertir en laboratorios. Esto los lleva a depender del método tradicional de observación periódica.

Una alternativa tecnológica e innovadora a las técnicas mencionadas, es la utilización de metodología basadas en inteligencia artificial (IA), tales como las redes neuronales (NN, por sus siglas en inglés: *Neural Network*). Las NN es un método de IA que permite a las computadoras procesar datos de manera similar al cerebro humano [4]. Una NN es un procesador distribuido masivamente paralelo compuesto de unidades de procesamiento simples que tiene una propensión natural para almacenar el conocimiento experiencial y ponerlo a disposición para su uso [5]. Específicamente para este tipo de problemática, se requiere analizar y clasificar imágenes, se hace uso de redes neuronales tipo convolucionales (por sus siglas en inglés, CNN), que se utilizan generalmente para tareas de clasificación y visión por computadora. En la actualidad, las CNN ofrecen un enfoque altamente adaptable para la clasificación de imágenes y el reconocimiento de objetos [6].

En una NN, se debe definir un modelo de arquitectura, debido a que influye en la capacidad de la red para representar y aprender de los datos, en su eficiencia computacional, en su adaptabilidad al problema y en la interpretabilidad del modelo resultante. Una elección adecuada de la arquitectura puede mejorar significativamente el rendimiento y la utilidad de la NN en la resolución de problemas. Una vez definida la arquitectura, el modelo debe someterse a un entrenamiento que le permitirá aprender a partir de los datos y que mejore su capacidad para realizar predicciones precisas. Generar un modelo completamente desarrollado puede conllevar tiempo de desarrollo si no se tienen los conocimientos necesarios ya que es preciso ajustar la arquitectura, reunir el conjunto de datos, depurarlo y entrenarlo.

Actualmente, se dispone de numerosos modelos pre-entrenados que han sido capacitados con grandes conjuntos de datos para aprender representaciones generales de los mismos [7]. Estos modelos se entrenan en tareas de aprendizaje automático a gran escala, como el reconocimiento de imágenes o el procesamiento del lenguaje natural, capturan patrones y características relevantes de los datos. Utilizar un modelo pre-entrenado ofrece la ventaja de ahorrar tiempo al evitar pasar por completo este proceso, y que, además, suelen tener una precisión superior al 90% en la clasificación, como se ha demostrado en trabajos previos. Por otro lado, es posible mejorar los resultados obtenidos de un modelo pre-entrenando realizando una técnica llamada “*Transfer Learning*”, que consiste en añadir capas adicionales, cambiar el número de clases, congelar o descongelar capas, modificar hiperparámetros, como fue realizado en [8].

Este trabajo se orienta en ofrecer una comparativa de rendimiento entre el modelo de desarrollo propuesto *GreenGuardNet* y el pre-entrenado *MobileNetV2*, con el fin de obtener una conclusión concisa que aporte en la selección de un modelo pre-entrenado o dedicar tiempo al desarrollo de uno. Ambos modelos serán utilizados en la detección de patógenos en plantas ornamentales.

## 1.2. Trabajos relacionados

Numerosos trabajos de investigación y proyectos han abordado el uso de NN para la detección de patógenos en diferentes áreas, como la medicina, la seguridad alimentaria y el control de enfermedades infecciosas. En [9], se utilizó un modelo de CNN combinado con la red *MobileNetV2* para detectar enfermedades en las hojas de la Yuca (*Manihot esculenta*). El modelo logró una precisión del 85%, aunque este resultado es considerado bueno, se espera que la precisión de una red neuronal sea superior al 90% para considerarlo aceptable. Es importante mencionar que el conjunto de datos de entrenamiento utilizado fue descargado de internet, lo cual no es recomendable, ya que las imágenes suelen tener baja calidad de píxeles, tamaño no estandarizado y mucho ruido que afecta el entrenamiento de la NN. En [10], se emplearon cinco arquitecturas de NN pre-entrenadas (*LeNet*, *ShuffleNet*, *AlexNet*, *EffNet* y *MobileNet*) para detectar enfermedades en las hojas de las plantas. Cada una de estas redes mostró resultados distintos, destacando *MobileNet* con una precisión del 95.77%. En [11], se empleó el modelo pre-entrenado *InceptionV3 model* y se aplicó Transferencia de Aprendizaje a una red más pequeña desarrollada por los investigadores para detectar enfermedades en las hojas del café. El objetivo era utilizar el conocimiento adquirido por el modelo pre-entrenado y aplicarlo a la tarea específica de detección de enfermedades en las hojas de café. Sin embargo, el enfoque de ambos estudios [10-11], se limitan a utilizar solo redes pre-entrenadas, sin explorar la posibilidad de desarrollar una arquitectura propia que potencialmente pudiera mejorar la precisión del modelo, tal como será desarrollado en el presente trabajo.

## 2. DESARROLLO

### 2.1. Dataset

El *dataset* (conjunto de datos) utilizado en este estudio para el entrenamiento de ambos modelos consta de una variedad de imágenes que representan el daño causado por dos patógenos en particular: la araña roja (*Tetranychus urticae*) [12] y el pulgón (*Aulacorthum solani*) [13]. El conjunto de datos se particionó empleando el método *Hold-out* [14], con 168 imágenes en cada una de ellas. Para determinar la partición se implementó una batería de pruebas con diferentes configuraciones, con el fin de determinar cual es la que mejor se adapta al modelo (ver tabla 1)

Tabla 1 Pruebas para la partición de los datos (Fuente: Propia).

Porcentaje de partición	Precisión
50-50	83.56%
60-40	85.16%
70-30	87.29%
80-20	93.45%
90-10	95.72%

Se concluye que la mejor partición que se adapta al conjunto de datos es 90-10, debido a que se cuenta con un conjunto de datos relativamente pequeño de 336 imágenes. Esto quiere decir que con un conjunto de entrenamiento más grande (90% de los datos), es posible que el modelo tenga una mejor capacidad para

aprender los patrones en los datos de entrenamiento, lo que puede llevar a una mayor precisión en los datos de prueba. Una partición del 70-30 tiene una menor cantidad de datos de entrenamiento, lo que resulta en un modelo que no ha capturado completamente los patrones subyacentes en los datos.

Además, el conjunto de datos se compone de una combinación de imágenes cuidadosamente seleccionadas de fuentes en línea y fotografías capturadas con una cámara de alta resolución. Esta combinación se realizó con el propósito de asegurar la diversidad en términos de calidad de las imágenes utilizadas en el proceso de entrenamiento. La figura 1 presenta algunos ejemplos de los datos de entrenamiento que se utilizaron para ambos modelos.

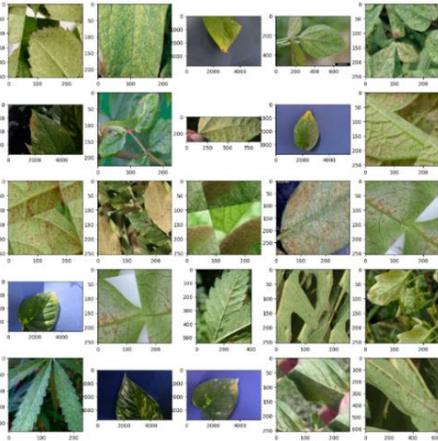


Figura 1 Dataset de Imágenes (Fuente: Propia).

Para mejorar el rendimiento del *dataset*, fue sometido a una técnica llamada “aumento de datos”, que selecciona al azar imágenes y realiza una serie de transformaciones tales como rotación, desplazamiento, ampliación o reducción, volteo horizontal o vertical, recorte aleatorio y modificación del brillo y contraste de la imagen. Esto ayuda al modelo evitar un sobre ajuste (*overfitting*). La figura 3 representa algunas imágenes sometidas al aumento de datos.



Figura 2 Aumento de Datos (Fuente: Propia).

## 2.2. Modelo propuesto *GreenGuardNet*

*GreenGuardNet* es una *CNN* secuencial implementada utilizando *TensorFlow/Keras*. Comienza con una capa convolucional de entrada 2D que utiliza 32 filtros de tamaño 3x3 y se aplica la función de activación *ReLU*. A continuación, se agrega una capa de *MaxPooling* que realiza un muestreo máximo de 2x2 en las características extraídas por la capa anterior, lo que ayuda a reducir la dimensionalidad y extraer las características

más relevantes. Las siguientes cinco capas siguen la misma estructura mencionada anteriormente, pero con un aumento en el número de filtros utilizados. Esto permite que la red aprenda características cada vez más complejas a medida que profundiza en las imágenes. Posteriormente, se incorpora una capa de *Dropout* que desactiva de forma aleatoria un número de neuronas en la capa anterior durante el entrenamiento. Esto ayuda a regularizar el modelo al reducir la dependencia de una única combinación de características. Se agrega una capa *Flatten* que transforma los mapas de características en un vector unidimensional, lo que permite conectarlos con capas totalmente conectadas. Se incluye una capa *Dense* con 250 neuronas totalmente conectadas utilizando la función de activación *ReLU*. Finalmente, se agrega una capa de salida con dos neuronas conectadas y se utiliza la función de activación *softmax* para clasificar las imágenes en dos clases distintas (ver figura 4).

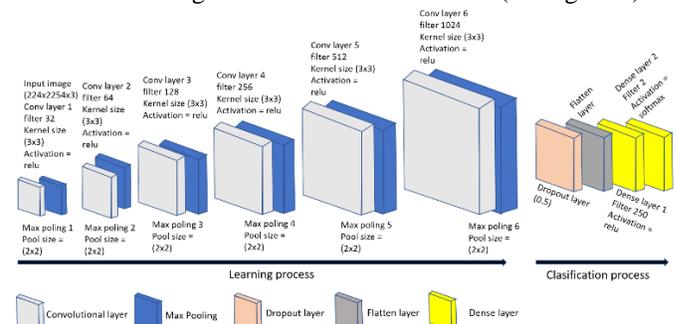


Figura 3 Modelo *GreenGuardNet* (Fuente: Propia).

Este modelo *CNN* es bastante profundo, con varias capas convolucionales y capas de reducción de dimensionalidad. Está diseñado para procesar imágenes de entrada de tamaño 224x224x3 y clasificarlas en dos categorías (araña roja y pulgón).

Es importante mencionar que la arquitectura propuesta en *GreenGuardNet* fue resultado de una combinación de pruebas exhaustivas en la variación de los parámetros en la configuración de la *NN*, tales como el número de capas, el número de neuronas en cada capa, funciones de activación, el tamaño del *batch* de entrenamiento, así como la cantidad de épocas en el entrenamiento.

## 2.3. Modelo pre-entrenado *MobileNetV2*

*MobileNetV2* se caracteriza por su estructura de residual invertido, donde las capas de entrada y salida son capas delgadas de tipo "cuello de botella". Esto contrasta con los modelos residuales tradicionales. Además, utiliza convoluciones de baja profundidad para filtrar características en la capa de expansión intermedia.

La arquitectura del modelo pre-entrenado *MobileNetV2* se compone de una capa convolucional inicial con 32 filtros,

seguida de 19 capas *Bottleneck*. Se utiliza la función de activación ReLU6 en cada capa, y el tamaño del *kernel* es de 3x3. Durante el entrenamiento, se emplean las capas *Dropout* y la normalización por lotes [15].

Tabla 2 Arquitectura de MobileNetV2 (Fuente: [13]).

Input	Operator	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
224 <sup>2</sup> × 3	conv2d	-	32	1	2
112 <sup>2</sup> × 32	bottleneck	1	16	1	1
112 <sup>2</sup> × 16	bottleneck	6	24	2	2
56 <sup>2</sup> × 24	bottleneck	6	32	3	2
28 <sup>2</sup> × 32	bottleneck	6	64	4	2
14 <sup>2</sup> × 64	bottleneck	6	96	3	1
14 <sup>2</sup> × 96	bottleneck	6	160	3	2
7 <sup>2</sup> × 160	bottleneck	6	320	1	1
7 <sup>2</sup> × 320	conv2d 1x1	-	1280	1	1
7 <sup>2</sup> × 1280	avgpool 7x7	-	-	1	-
1 × 1 × 1280	conv2d 1x1	-	k	-	-

#### 2.4. Criterios de comparación

Para conocer el rendimiento de una red neuronal, existen varias métricas utilizadas para evaluarlas. Las métricas seleccionadas dependen del tipo de problema tratado y de los objetivos propuestos. Para este estudio, se seleccionaron las siguientes métricas para evaluar el modelo *GreenGuardNet* y el pre-entrenado.

**Tiempo de entrenamiento:** se considera el tiempo del entrenamiento de cada modelo.

**Accuracy:** nos indica la proporción de aciertos que obtuvo el modelo (ver ecuación 1).

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (1)$$

**Precision:** nos indica el porcentaje de los aciertos como verdaderos respecto a los predichos como verdaderos (ver ecuación 2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

**Recall:** nos dice el porcentaje de los aciertos verdaderos del modelo respecto a los que realmente son verdaderos (ver ecuación 3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

**Loss:** esta métrica es utilizada durante el entrenamiento y nos mide que tan bien se ejecutaron las predicciones a los valores reales. La función de pérdida seleccionada fue entropía cruzada binaria (*Binary Cross Entropy*) [16].

### 3. RESULTADOS

A continuación, en la Tabla 2 se exponen las evidencias del entrenamiento y ejecución de ambos modelos, destacando los criterios de comparación.

Tabla 3 Criterios de Comparación (Fuente: Propia).

Modelo	Tiempo de entrena	Accuracy	Precision	Recall	Loss
<b>GreenGuardNet</b>	48 min.	95.72%	0.75	0.8	0.1225
<b>MobileNetV2</b>	32 min.	95.72%	0.8	0.6	0.1162

Las figuras 5 y 6 muestran dos gráficas en donde se hace una comparativa de la exactitud en el entrenamiento y en pruebas, así como la pérdida en entrenamiento y en pruebas de *GreenGuardNet* y *MobileNetV2*.

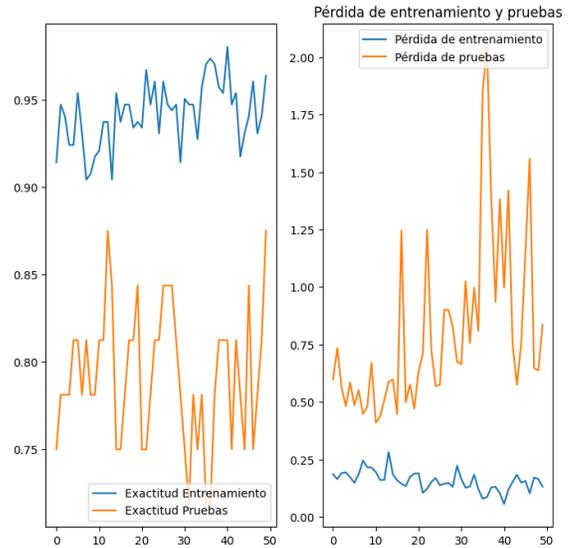


Figura 4 Exactitud y Pérdida en Entrenamiento y Pruebas de *GreenGuardNet* (Fuente: Propia).

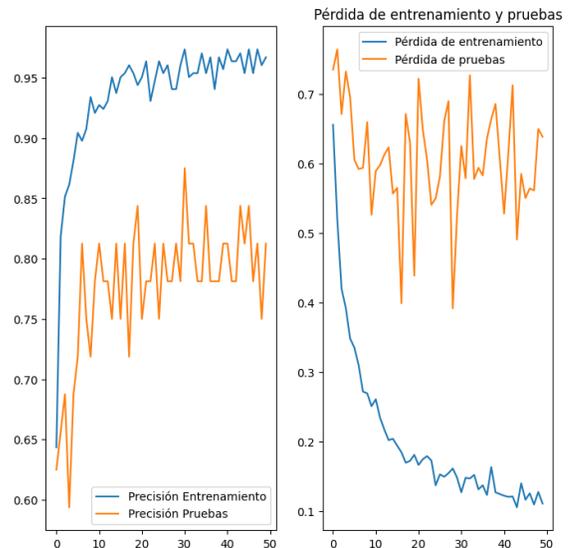


Figura 5 Exactitud y Pérdida en Entrenamiento y Pruebas de *MobileNetV2* (Fuente: Propia).

#### 4. TRABAJO FUTURO

Los resultados obtenidos en este trabajo son una fuerte motivación para su continuación. Con el objetivo de evolucionar el modelo, se plantean las siguientes pistas.

- Mejorar la precisión y el rendimiento para la detección de los patógenos.
- Para beneficiar a los productores se portaría este modelo a una aplicación móvil, para incrementar la disponibilidad de este servicio.

#### 5. CONCLUSIONES

Este estudio reporta la comparación de rendimiento entre dos modelos *CNN*: un modelo desarrollado *GreenGuardNet* y uno pre-entrenado *MobileNetV2*. Ambos mostraron la misma exactitud 95.72%, en términos de precisión el modelo pre-entrenado mostró una ligera ventaja.

El tiempo de desarrollo es otro factor relevante, el uso de un modelo pre-entrenado reduce el tiempo de entrenamiento al congelar ciertos parámetros. Si el objetivo principal es incrementar la precisión del modelo, desarrollar uno desde cero y ajustarlo gradualmente puede ser la opción más adecuada. En situaciones en las que no se disponga de las competencias suficientes, pocos datos de entrenamiento o el tiempo necesario, utilizar un modelo pre-entrenado puede ser una oportunidad viable.

#### 6. REFERENCIAS

- [1] “Impulsar la industria ornamental de México con la experiencia y la innovación holandesas – OPF NEWS”, el 13 de septiembre de 2021. <https://opf.news/impulsar-la-industria-ornamental-de-mexico-con-la-experiencia-y-la-innovacion-holandesas/> (consultado el 5 de junio de 2023).
- [2] “Diagnóstico de enfermedades de las plantas.”, Normec Groen Agro Control. <https://agrocontrol.nl/es/servicios/diagnostico-de-enfermedades-de-las-plantas/> (consultado el 17 de mayo de 2023).
- [3] V. Valenzuela-Herrera y E. Redondo-Juárez, “Detección de Virus por Serología y Plantas Indicadoras en el Tubérculo-Semilla y Plantas de Cultivo de Meristemas en Papa (*Solanum tuberosum* L.) var. Alfa”.
- [4] “¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS”, Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/neural-network/> (consultado el 18 de mayo de 2023).
- [5] S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.
- [6] “¿Qué son las redes neuronales convolucionales? | IBM”. <https://www.ibm.com/mx-es/topics/convolutional-neural-networks> (consultado el 18 de mayo de 2023).
- [7] “TensorFlow Hub”. <https://tfhub.dev/> (consultado el 6 de septiembre de 2023).
- [8] T. Wolf, V. Sanh, J. Chaumond, y C. Delangue, “TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents”. arXiv, el 4 de febrero de 2019. Consultado: el 6 de septiembre de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1901.08149>
- [9] R. Surya y E. Gautama, “Cassava Leaf Disease Detection Using Convolutional Neural Networks”, en 2020 6th International Conference on Science in Information Technology (ICSITech), Palu, Indonesia: IEEE, oct. 2020, pp. 97–102. doi: 10.1109/ICSITech49800.2020.9392051.
- [10] A. K. Singh y B. Chaurasia, “Plant Leaf Disease Detection using Convolutional Neural Network and Random Forest Classifier”, International Journal of Innovations in Engineering and Science, vol. 6, núm. 10, p. 204, sep. 2021, doi: 10.46335/IJIES.2021.6.10.42.
- [11] M. Kumar, P. Gupta, P. Madhav, y Sachin, “Disease Detection in Coffee Plants Using Convolutional Neural Network”, en 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India: IEEE, jun. 2020, pp. 755–760. doi: 10.1109/ICCES48766.2020.9138000.
- [12] “Araña roja | Koppert México”. <https://www.koppert.mx/retos/control-de-plagas/aranas-rojas-y-otras-aranas/arana-roja/> (consultado el 19 de junio de 2023).
- [13] “Pulgones | Koppert México”. <https://www.koppert.mx/retos/control-de-plagas/pulgones/> (consultado el 19 de junio de 2023).
- [14] A. Kumar, “Hold-out Method for Training Machine Learning Models”, Data Analytics, el 22 de mayo de 2023. <https://vitalflux.com/hold-out-method-for-training-machine-learning-model/> (consultado el 19 de junio de 2023).
- [15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, y L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. arXiv, el 21 de marzo de 2019. Consultado: el 19 de mayo de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1801.04381>.
- [16] “7. Entrenamiento de redes neuronales c) Métricas [logongas]”. <https://logongas.es/doku.php?id=clase:iabd:pia:2eval>.