IDEAS EN CIENCIAS DE LA INGENIERÍA

Optimización Factorial de Redes Neuronales Convolucionales para la Detección de Pulgones (Aphidinae) en Cultivos de Rosal

Factorial Optimization of Convolutional Neural Networks for Aphid (Aphidinae) Detection in Rose Crops

José Hernández[1], Eréndira Rendón [1], Roberto Alejo [1], Federico del Razo [1], Everardo Granda [2],

Daniel Cervantes [1] *
(*dd26280389@toluca.tecnm.mx)

Recibido: 30 de agosto de 2026 Aceptado: 09 de octubre de 2026

Resumen

La implementación de redes neuronales convolucionales en la agricultura de precisión enfrenta el desafío de la configuración empírica de sus hiperparámetros, lo que deriva en un uso ineficiente de recursos computacionales y en tiempos de entrenamiento prolongados. Para abordar esta limitación, el presente estudio propone una metodología basada en un diseño experimental factorial de tres factores para optimizar el rendimiento de las arquitecturas VGG16 y EfficientNetB0 en la clasificación de pulgones (Aphidinae) en cultivos de rosal. Para ello, se empleó un conjunto de datos conformado por 1438 imágenes, particionado en 1222 instancias para la fase de entrenamiento y 216 para el conjunto de prueba. El entrenamiento de las arquitecturas se ejecutó mediante niveles fijos de épocas, prescindiendo deliberadamente de mecanismos de detención temprana con el propósito de cuantificar de manera aislada el impacto de la convergencia dentro del diseño factorial. Se evaluaron el tamaño de lote, el número de épocas y la función de activación, analizando tanto sus efectos principales como sus interacciones sobre la exactitud del modelo. Los resultados del análisis de varianza mostraron que la función de activación es el factor con mayor influencia estadística en ambas arquitecturas. Como hallazgo principal, EfficientNetB0 superó a VGG16, alcanzando una Accuracy del 94.42 % bajo la configuración óptima de lote de 32, 30 épocas y activación *softmax*. Este enfoque demuestra que la aplicación de

[1]Departamento de Estudios de Posgrado e Investigación, Instituto Tecnológico de Toluca, Metepec, Estado de México, México.
[2]Centro Universitario UAEM Atlacomulco, Universidad Autónoma del Estado de México, Atlacomulco, Estado de México, México.

diseños factoriales elimina la incertidumbre del ajuste por prueba y error, lo que favorece modelos más robustos, reproducibles y escalables para el monitoreo de plagas. En esta investigación se han evaluado dos arquitecturas convolucionales representativas; no obstante, esta metodología es extensible a otras topologías con parámetros análogos y a diversas aplicaciones.

Palabras clave: CNN, clasificación, redes neuronales, diseño factorial, rosas, pulgones

Abstract

The implementation of convolutional neural networks in precision agriculture faces the challenge of empirically configuring hyperparameters, resulting in inefficient use of computational resources and prolonged training times. To address this limitation, the present study proposes a methodology based on a three-factor factorial experimental design to optimize the performance of the VGG16 and EfficientNetB0 architectures for detecting aphids (Aphidinae) in rose crops. To this end, a dataset comprising 1,438 images was used, partitioned into 1,222 instances for the training phase and 216 for the test set. The training of the architectures was executed using fixed epoch levels, deliberately dispensing with early stopping mechanisms to isolate and quantify the impact of convergence within the factorial design. We evaluated batch size, number of epochs, and activation function, analyzing both their main effects and interactions on model accuracy. The analysis of variance showed that the activation function had the greatest statistical influence on both architectures. As the main finding, EfficientNetB0 outperformed VGG16, achieving 94.42 % accuracy under the optimal configuration of a batch size of 32, 30 epochs, and the softmax activation function. This approach shows that the application of factorial designs eliminates the uncertainty of trial-and-error tuning, promoting more robust, reproducible, and scalable models for pest monitoring. Although this research evaluated two representative convolutional architectures, this methodology is extensible to other topologies with analogous parameters and to diverse applications.

Keywords: CNN, clasification, neural networks, factorial design, roses, aphids

Introducción

En la actualidad, diversas aplicaciones del aprendizaje automático, principalmente las redes neuronales convolucionales (CNN, por sus siglas en inglés, *convolutional neural networks*) han generado un amplio campo de propuestas debido a su capacidad para analizar imágenes y extraer información relevante para tareas específicas. En el ámbito agrícola, estas herramientas han sido empleadas como una alternativa para identificar enfermedades y plagas en diferentes cultivos.

Por ejemplo, en [1] se empleó un modelo CNN para apoyar en la problemática causada por la presencia

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

de plagas en plantas de papa. El tizón tardío fue detectado en su etapa temprana mediante análisis de imágenes de las hojas de los cultivos, obteniendo una precisión del 90 % en el diagnóstico de la presencia de la plaga. Asimismo, en el artículo [2] se compararon modelos como VGG16, CNNs de 8 y 12 capas, así como un modelo FCN (*fully connected networks*) de 2 capas para la clasificación de plagas y enfermedades presentes en cultivos de papa y maíz.

Otro ejemplo de aplicación de redes neuronales se expone en el artículo [3], donde a partir de imágenes de rosas, en un experimento controlado, se empleó una red neuronal con el fin de detectar la presencia de moho. Por otra parte, en la investigación realizada por [4] se expone un sistema de clasificación automática de insectos llamado InsectCV, a partir de imágenes de trampas Moericke para analizar plagas como el pulgón, parasitoides y *trips*, obteniendo niveles de confianza del 98.5 % en parasitoides y 97.6 % en pulgón.

Por otra parte, el procesamiento digital de imágenes permite extraer características esenciales para el análisis o la identificación de objetos de interés, través de técnicas (preprocesamiento, transformación, realce y segmentación) con la finalidad mejorar la calidad de la imagen, reduciendo el ruido (información sin importancia) o adaptarla para alguna aplicación.

En el contexto agrícola, los cultivos de rosas representan una importante actividad económica en México. La producción se concentra principalmente en el Estado de México, Puebla y Morelos, siendo el Estado de México el de mayor participación nacional [5]. Entre los principales municipios productores de rosa del Estado de México se encuentran Tenancingo, Villa Guerrero, Coatepec de Harinas, Zumpahuacán, Ixtapan de la Sal y Malinalco.

Las ganancias y producción son mermadas por plagas comunes en las rosas, además de requerirse una mayor inversión económica por parte de los productores en insumos como pesticidas, personal de inspección y tiempo para la eliminación de cultivos infectados. Cuando estas plantas no son atendidas en una etapa temprana, existe la posibilidad de propagación hacia plantas sanas. Además, puesto que la inspección de los cultivos es realizada por personal operativo en campo, existe la posibilidad de omitir plantas afectadas por pulgones debido al error humano [6].

Entre las plagas comunes de los rosales se encuentra el pulgón (*Aphidinae*), descrito como un insecto de aproximadamente 3 mm de longitud y color verdoso que se presenta principalmente en vástagos jóvenes o yemas florales, provocando manchas descoloridas y hundidas en los pétalos. Su desarrollo se presenta usualmente en ambientes secos y con temperaturas templadas a cálidas [7].

En el contexto de los párrafos precedentes, es importante destacar que que se eligió como caso de estudio esta plaga y cultivo, por dos razones principales: en primer lugar, por el impacto económico de la plaga, y además por el desafío que supone el desarrollo de herramientas para automatizar una inspección

visual sujeta a errores humanos, en donde las imágenes presentan variabilidad debido la iluminación y la composición del fondo, lo cual resulta particularmente idóneo para evaluar la robustez del método. Cabe aclarar que la metodología propuesta no presenta restricciones técncias o teóricas para su aplicación a otros cultivos o incluso en sectores diferentes a la agricultura.

Algunos de los modelos más utilizados para clasificación automática de imágenes en plantas, que se han identificado en la literatura, son VGG (en sus variantes), EfficientNet, ResNet y MobileNet, entre otros [8]. Sin embargo, la configuración de sus parámetros se realiza comúnmente de forma empírica, mediante prueba y error, y son pocos los trabajos que abordan esta problemática mediante un procedimiento sistemático de ajuste que permita incrementar la precisión de los modelos [9].

En el contexto del aprendizaje profundo, el ajuste de hiperparámetros constituye una de las etapas más críticas y costosas en el desarrollo de modelos, ya que comúnmente se realiza mediante enfoques empíricos como la búsqueda exhaustiva (*grid search*) o aleatoria (*random search*). Sin embargo, estos métodos pueden requerir un número combinatorio de experimentos conforme aumenta la cantidad de hiperparámetros y de valores considerados, lo que eleva el consumo de recursos computacionales y prolonga los tiempos de entrenamiento. Además, no aprovechan explícitamente la información obtenida en evaluaciones anteriores, por lo que pueden explorar de forma ineficiente el espacio de búsqueda [10].

Ante esta problemática, se ha propuesto utilizar un diseño factorial para analizar de manera estructurada y simultánea la influencia de múltiples factores y sus interacciones sobre el desempeño del modelo, reduciendo el número de experimentos necesarios. Además, este enfoque proporciona evidencia estadística que respalda la selección de configuraciones adecuadas, incrementando la confiabilidad y reproducibilidad de los resultados obtenidos [11].

En este sentido, el objetivo de esta investigación es evaluar los resultados de un diseño de experimentos factorial, de tres factores, para obtener los parámetros durante el entrenamiento de los modelos VGG16 y EfficientNetB0, con la finalidad de identificar la presencia de pulgón en imágenes de plantas de rosales con una exactitud (*accuracy*) mayor al 90 %. Los factores analizados son: tamaño de lote, número de épocas y función de activación, considerando también sus interacciones, donde la Accuracy obtenida por los modelos es la variable de respuesta.

La integración del diseño factorial en el proceso de ajuste de parámetros contribuye al desarrollo de una metodología para determinar las configuraciones con mayor influencia sobre el desempeño de los modelos VGG16 y EfficientNetB0, además de generar una alternativa para la identificación de pulgones mediante imágenes de plantas de rosales.

Finalmente, la principal aportación de este trabajo consiste en la propuesta de un método estadístico para el ajuste de parámetros en redes neuronales convolucionales, con el propósito de mejorar, en este

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

caso, la clasificación de pulgones en rosales. A diferencia de enfoques tradicionales, como la búsqueda aleatoria, la metodología propuesta permite evaluar tanto el efecto individual de cada parámetro como sus interacciones simultáneas, reduciendo significativamente el número de experimentos requeridos y, al mismo tiempo, proporcionando una base estadística para la selección de la configuración óptima del modelo.

Materiales y métodos

Para establecer el enfoque propuesto y materializar los objetivos planteados, se diseñó una arquitectura experimental en la que convergen la ingeniería de datos, la selección de modelos de aprendizaje profundo y el sustento estadístico. Esta sección detalla los componentes fundamentales del estudio, iniciando con la conformación y el preprocesamiento del conjunto de imágenes para el entrenamiento. Posteriormente, se describen las topologías de las redes VGG16 y EfficientNetB0, destacando sus particularidades estructura-les. El apartado culmina con la formalización del diseño factorial que rige la exploración del espacio de hiperparámetros y la validación de su impacto en la Accuracy de los modelos.

Conjunto de datos

El conjunto de imágenes utilizado en esta investigación fue adquirido de diferentes fuentes: 1) de la base de datos Flowers Dataset [12], disponible en Kaggle, se identificaron imágenes de libre uso, 2) usando el motor de búsqueda Google se seleccionaron imágenes de rosales sin derechos de autor, y 3) imágenes tomadas de rosales en jardines.

Para conformar el conjunto de datos se utilizaron imágenes de rosales sanos y rosales con pulgones. Se consideraron imágenes de diferentes secciones de la planta, como tallos, hojas, botones y pétalos, así como imágenes tomadas desde distintos ángulos. Se eliminaron aquellas correspondientes a arreglos florales, ilustraciones, representaciones de rosas, plagas diferentes a los pulgones y elementos ajenos al objeto de estudio. Los archivos obtenidos mediante el motor de búsqueda Google fueron revisados para eliminar imágenes corruptas antes de ser utilizados en los modelos.

Posteriormente, las imágenes fueron clasificadas en dos grupos: rosales sanos y rosales afectados por pulgones, y la distribución del conjunto de datos se especifica en la Tabla 1. Con el propósito de garantizar la validez estadística de las métricas de evaluación empleadas, ambas clases se mantuvieron en una distribución estrictamente equiprobable. Esta decisión de diseño obedece a que, en conjuntos de datos balanceados, la exactitud global constituye un estimador fiable del rendimiento del clasificador, al no existir sesgo hacia la clase mayoritaria. Las imágenes fueron unificadas en formato JPG con dimensiones de $224 \times 224 \times 3$ píxeles. Las características consideradas para la clasificación de rosales sanos correspondieron principalmente a tallos, hojas, pétalos, flores y botones; mientras que para los rosales afectados se consideraron tallos, hojas, botones y la presencia de pulgones [7]. La tarea de clasificación consistió en

determinar, a partir de nuevas imágenes analizadas por los modelos de redes neuronales convolucionales, la presencia de pulgones o la ausencia de esta plaga.

Tabla 1. Distribución de imágenes por clase y subconjunto de datos.

Subconjunto	Rosas Sanas	Rosas Plagadas	Total
Entrenamiento	611	611	1222
Prueba	108	108	216
Total general	**719**	**719**	**1438**

Modelos propuestos de Deep Learning

El aprendizaje profundo (DL por las siglas en inglés de *Deep Learning*) es una rama de la inteligencia artificial (IA) que permite a los sistemas, a través de algoritmos, aprender de conjuntos de datos con la finalidad de mejorar su desempeño en tareas específicas. Es un enfoque basado a partir de modelos matemáticos (algoritmos) que identifican patrones en los datos, útiles para tareas de clasificación de nuevos datos [13, 14].

VGG16

El modelo VGG16 fue diseñado en 2015 por Simonyan y Zisserman, con una arquitectura simple [15]. Las imágenes de entrada tienen un tamaño fijo (224x224x3 píxeles) y pasan a través de una serie de capas convolucionales y de agrupación (*pooling*) para extraer características relevantes.

Cada capa convolucional aplica filtros (*kernels*) para detectar patrones como bordes, texturas y formas; las primeras capas extraen bordes y contornos básicos, las capas intermedias detectan texturas y partes del objeto, las capas finales identifican características específicas de la clase y asignan probabilidades a cada una.

En las capas de convolución, los filtros poseen tamaño fijo de 3x3 (filtros pasa bajo), con paso de 1 en los pixeles de la imagen. Por otro lado, la capa de agrupamiento emplea una función *max-pooling* de 2x2 que reduce la dimensionalidad de la matriz.

VGG16 consta de 16 capas con pesos entrenables: 13 capas convolucionales y 3 capas completamente conectadas [16,17]. La arquitectura se organiza en cinco bloques, cada uno con 2 o 3 capas convolucionales seguidas de una capa de agrupación (*max-pooling*). El número de filtros en las capas convolucionales comienza en 64 y se duplica en cada bloque hasta alcanzar 512. Las tres capas completamente conectadas

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

al final de la red tienen 4,096, 4,096 y 1,000 nodos, respectivamente. En la capa de salida utiliza una función *softmax* para la clasificación de los datos entre 0 y 1. Esta red neuronal se ilustra en la Figura 1.

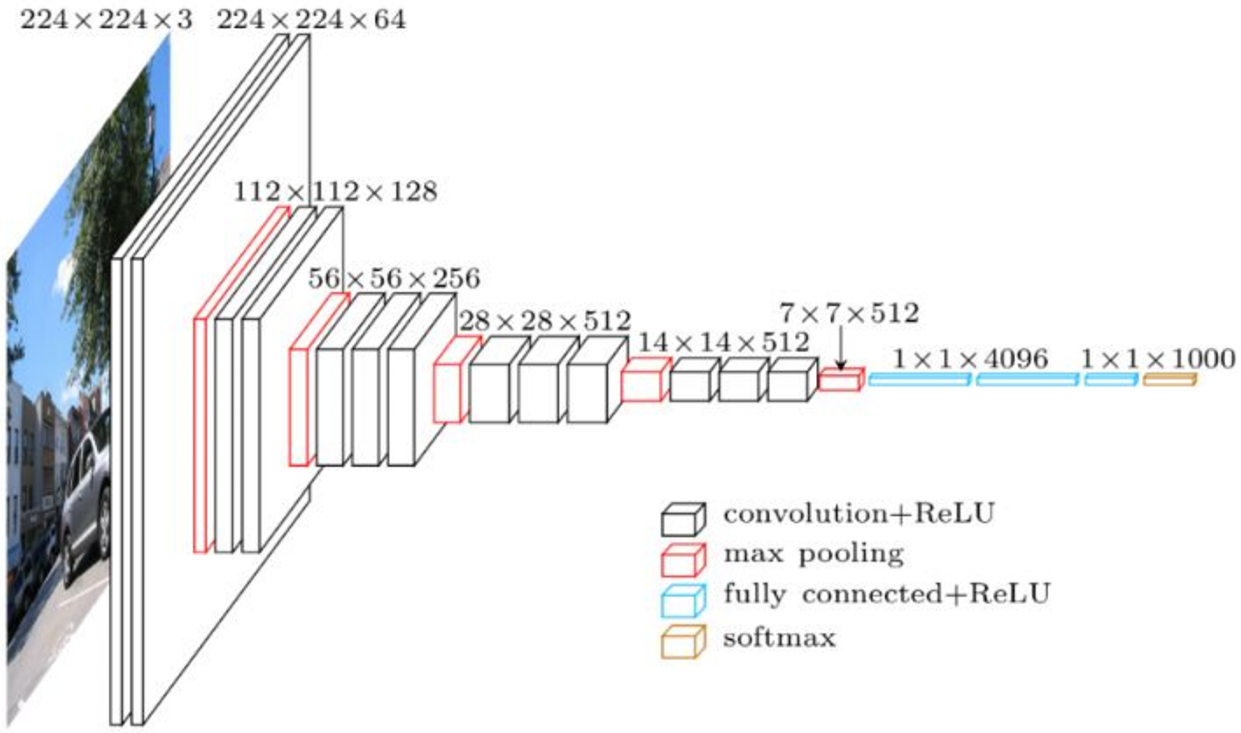


FIGURA 1. Arquitectura de una red VGG16. Imagen de la tesis doctoral [18].

EfficientNetB0

EfficientNetB0 es un modelo de red neuronal convolucional desarrollado por investigadores de Google en el 2019. Pertenece a la familia de EfficientNet, y su arquitectura se basa en un conjunto de capas iniciales (convolución y normalización de lotes con una función activación *Swish*), seguida de bloques *MBConv* que tiene diferentes tamaños de *kernel* y numero de filtros que dan paso a las capas finales donde hay una capa de convolución después una capa de agrupamiento y una capa completamente conectada con la función de activación *softmax* [19]. Su arquitectura se observa en la Figura 2.

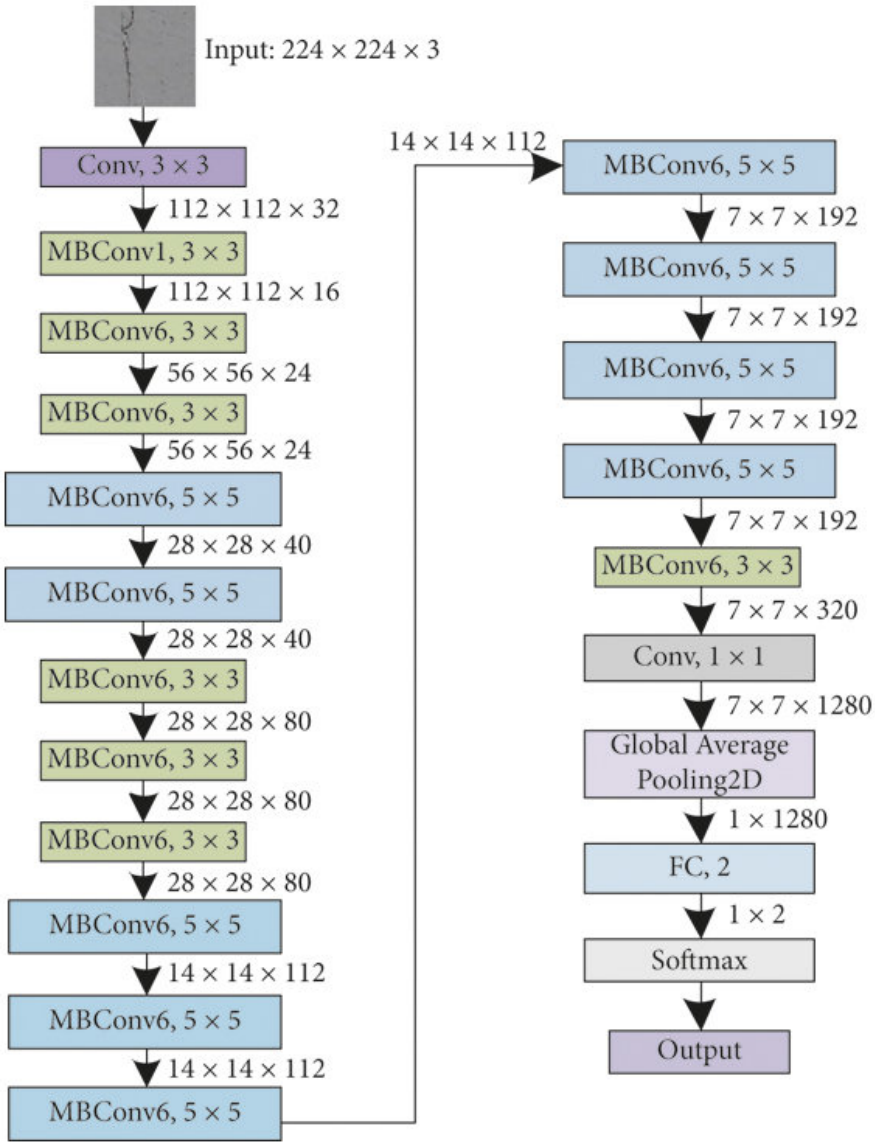


FIGURA 2. Arquitectura de una red EfficientNetB0. Imagen del articulo [20].

Posee tres puntos importantes enfocados en la maximización de la Accuracy: profundidad, ancho y resolución de la imagen de entrada, a través de un coeficiente ϕ. La Ec. (1) describe el escalado compuesto (*Compound Scaling*) de EfficientNet. Su propósito es determinar cómo aumentar el tamaño de la red neuronal de manera equilibrada, escalando simultáneamente la profundidad, el ancho y la resolución de entrada.

$$d = \alpha^{\phi}, \quad w = \beta^{\phi}, \quad r = \gamma^{\phi} \tag{1}$$

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

Donde, ϕ es el coeficiente de escalado compuesto (determina cuántos recursos adicionales están disponibles), α, β, γ son los factores de balance para escalar profundidad d, ancho w y resolución r, respectivamente.

La función de pérdida para este modelo se expresa en la Ec. (2) [19].

$$\zeta = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{2}$$

Donde, C es el número de clases, N es el número de entradas del conjunto de datos, y_{ij} es un valor binario que indica si la muestra i pertenece a la clase j, e $\hat{y}_{ij}$ es la probabilidad predicha de la clase j para la muestra i (salida de la función *softmax*).

Matriz de confusión

Una matriz de confusión es una herramienta utilizada para evaluar el desempeño de un modelo de clasificación mediante la comparación entre las clases predichas por el modelo y las etiquetas reales de los datos. Esta se representa mediante una tabla que permite identificar y cuantificar los aciertos y errores de clasificación.

En el caso de una clasificación binaria, se describe bajo la Ec. (3) [15], y se compone de cuatro categorías: Verdaderos Positivos (VP), que corresponden a los casos en los que el modelo predice correctamente la clase positiva; Falsos Positivos (FP), que representan los casos en los que el modelo predice la clase positiva cuando la clase real es negativa; Falsos Negativos (FN), que corresponden a los casos en los que el modelo predice la clase negativa cuando la clase real es positiva; y Verdaderos Negativos (VN), que representan los casos correctamente clasificados como pertenecientes a la clase negativa.

$$\text{Matriz de confusión} = \begin{bmatrix} VP & FP \\ FN & VN \end{bmatrix} \tag{3}$$

A partir de la matriz de confusión, se calcula la precisión del modelo mediante la Ec. (4), que define la Accuracy, y usualmente se representa mediante un porcentaje de clasificación correcta de los datos del conjunto de prueba [15]:

$$\text{Accuracy} = \frac{(VP + VN)}{(VP + FP + FN + VN)} \tag{4}$$

Dado que las topologías originales de VGG16 y EfficientNetB0 están concebidas para la clasificación de múltiples clases, fue necesario realizar una adaptación estructural en sus capas finales para atender la naturaleza binaria de la presente tarea. Específicamente, las capas completamente conectadas de salida

fueron reemplazadas por una nueva capa densa conformada por dos neuronas, correspondientes a las clases de interés. Esta topología de salida, en conjunto con la función de pérdida de entropía cruzada categórica, dicta el comportamiento matemático de las funciones de activación evaluadas. Mientras que la función *softmax* normaliza las salidas en una distribución de probabilidad conjunta idónea para clases mutuamente excluyentes, la función *sigmoide* trata cada neurona de manera independiente. Esta distinción arquitectónica es fundamental para comprender las divergencias en el rendimiento observadas durante el diseño factorial.

Diseño Factorial

Un diseño de experimentos factorial se enfoca en la influencia de cada uno de los factores principales y sus niveles en la respuesta resultante de un experimento. Para ello, cada experimento contempla la combinación de cada factor con sus respectivos niveles. Por ejemplo, si se tiene el factor A con a niveles, el factor B con b niveles y el factor C con c niveles, cada experimento contiene todas las combinaciones de abc. Los efectos que se producen de un factor son reflejados en la variable de respuesta como resultado del cambio en el nivel del factor. Si A,B y C son los efectos principales en el análisis del diseño factorial, la tabla de análisis de varianza del modelo de tres factores con efectos se representa en la Tabla 2 [11].

Para el análisis de los efectos es necesario un mínimo de dos replicas $(n \geq 2)$ con el fin de determinar la suma de cuadrados del error, si todas las interacciones posibles están incluidas en el diseño. El modelo del análisis de varianza de tres factores se representa mediante la Ec. (5).

$$y_{ijkl} = \mu + \tau_i + \beta_j + \gamma_k + (\tau\beta)_{ij} + (\tau\gamma)_{ik} + (\beta\gamma)_{jk} + (\tau\beta\gamma)_{ijk} + \epsilon_{ijkl} \quad \begin{cases} i = 1, 2, \ldots, a \\ j = 1, 2, \ldots, b \\ k = 1, 2, \ldots, c \\ l = 1, 2, \ldots, n \end{cases} \tag{5}$$

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

Tabla 2. Análisis de varianza del modelo de tres factores [11].

Fuente de variación	Suma de cuadrados	Grados de libertad	Cuadrado medio	Cuadrado medio esperado	F_0
A	SS_A	$a-1$	MS_A	$\sigma + \frac{bcn\sum \tau_i^2}{a-1}$	$\frac{MS_A}{MS_E}$
B	SS_B	$b-1$	MS_B	$\sigma^2 + \frac{acn\sum \beta_i^2}{b-1}$	$\frac{MS_B}{MS_E}$
C	SS_C	$c-1$	MS_C	$\sigma^2 + \frac{abn\sum \gamma_i^2}{c-1}$	$\frac{MS_C}{MS_E}$
AB	SS_{AB}	$(a-1)(b-1)$	MS_{AB}	$\sigma^2 + \frac{cn\sum\sum(\tau\beta)_{ij}^2}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_E}$
AC	SS_{AC}	$(a-1)(c-1)$	MS_{AC}	$\sigma^2 + \frac{bn\sum\sum(\tau\gamma)_{ik}^2}{(a-1)(c-1)}$	$\frac{MS_{AC}}{MS_E}$
BC	SS_{BC}	$(b-1)(c-1)$	MS_{BC}	$\sigma^2 + \frac{an\sum\sum(\beta\gamma)_{jk}^2}{(b-1)(c-1)}$	$\frac{MS_{BC}}{MS_E}$
ABC	SS_{ABC}	$(a-1)(b-1)(c-1)$	MS_{ABC}	$\sigma^2 + \frac{n\sum\sum(\tau\beta\gamma)_{ijk}^2}{(a-1)(b-1)(c-1)}$	$\frac{MS_{ABC}}{MS_E}$
Error	SS_E	$abc(n-1)$	MS_E	σ^2	
Total	SS_T	$abcn-1$			

La suma de cuadrados total se calcula mediante la Ec. (6).

$$SS_T = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}\sum_{l=1}^{n} y_{ijkl}^2 - \frac{y^2}{abc...n} \tag{6}$$

Mientras que las sumas de cuadrados de los efectos principales se encuentran a partir de los totales de los factores $A(y_{i...})$, $B(y_{.j.})$ y $C(y_{..k.})$. En las Ec. (7, 8 y 9) se muestra cómo efectuar el cálculo.

$$SS_A = \frac{1}{bcn}\sum_{i=1}^{a} y_{i...}^2 - \frac{y_{...}^2}{abcn} \tag{7}$$

$$SS_B = \frac{1}{acn}\sum_{j=1}^{b} y_{.j.}^2 \frac{y_{...}^2}{abcn} \tag{8}$$

$$SS_C = \frac{1}{abn}\sum_{k=1}^{c} y_{..k.}^2 - \frac{y_{...}^2}{abcn} \tag{9}$$

Para calcular las sumas de cuadrados de las interacciones entre dos factores, se utilizan los totales de las celdas $A \times B$, $A \times C$ y $B \times C$. Las Ec. (10, 11 y 12) muestran cómo realizar este cálculo.

$$SS_{AB} = \frac{1}{cn}\sum_{i=1}^{a}\sum_{j=1}^{b} y_{ij..}^2 - \frac{y_{...}^2}{abcn} - SS_A - SS_B = SS_{\text{subtotales}(AB)} - SS_A - SS_B \tag{10}$$

$$SS_{AC} = \frac{1}{bn}\sum_{i=1}^{a}\sum_{k=1}^{c} y_{i.k.}^2 - \frac{y_{...}^2}{abcn} - SS_A - SS_C = SS_{\text{subtotales}(AC)} - SS_A - SS_C \tag{11}$$

$$SS_{BC} = \frac{1}{an}\sum_{j=1}^{b}\sum_{k=1}^{c} y_{.jk.}^2 - \frac{y_{...}^2}{abcn} - SS_B - SS_C = SS_{\text{subtotales}(BC)} - SS_B - SS_C \tag{12}$$

En un diseño factorial de tres factores, la suma de cuadrados de la interacción triple se calcula a partir de los totales de las celdas $\{y_{ijk.}\}$. Las Ec. (13 y 14) describen el proceso de cálculo.

$$SS_{ABC} = \frac{1}{n}\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c} y_{ijk.}^2 - \frac{y_{...}^2}{abcn} - SS_A - SS_B - SS_C - SS_{AB} - SS_{AC} - SS_{BC} \tag{13}$$

$$SS_{ABC} = SS_{\text{subtotales}(ABC)} - SS_A - SS_B - SS_C - SS_{AB} - SS_{AC} - SS_{BC} \tag{14}$$

Por último, la suma de cuadrados del error puede calcularse con la Ec. (15).

$$SS_E = SS_T - SS_{\text{subtotales}(ABC)} \tag{15}$$

RESULTADOS Y DISCUSIÓN

Para el proceso de entrenamiento, se estableció un conjunto de hiperparámetros base que se mantuvieron constantes a lo largo de las pruebas, según la configuración por defecto de cada modelo, salvo aquellos factores que formaron parte del diseño experimental.

No obstante, se debe destacar la estrategia de entrenamiento respecto al número de épocas. A diferencia de las prácticas convencionales que recurren a mecanismos de detención temprana (*early stopping*)

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

para prevenir el sobreajuste, en esta investigación se optó por entrenar las redes mediante niveles fijos de iteración. Esta decisión responde a la necesidad de cuantificar de manera aislada el impacto de la convergencia como un factor independiente dentro del diseño factorial. Al fijar el número de épocas, se evita que la detención temprana enmascare la verdadera influencia de esta variable sobre la varianza del modelo, garantizando que el análisis de varianza refleje el efecto real de cada nivel evaluado.

De este modo, y como resultado de partida, se presenta la Tabla 3, que muestra la configuración inicial y la Accuracy obtenida antes del ajuste estadístico.

Tabla 3. Configuración inicial de hiperparámetros y Accuracy obtenida antes del ajuste estadístico.

Modelo	**Tamaño de Lote**	**Épocas**	**Activación**	**Accuracy Inicial (%)**
VGG16	32	150	sigmoid	81.32
EfficientNetB0	16	150	softmax	92.09

Tras el análisis de varianza de los modelos EfficientNetB0 y VGG16, se obtuvieron los valores de p correspondientes a los factores utilizados en el diseño de experimentos factorial, correspondientes al tamaño de lote (factor A), número de épocas (factor B), función de activación (factor C) y sus interacciones.

Como se muestra en la Tabla 4, para EfficientNetB0 únicamente la función de activación C presentó un efecto significativo sobre la *accuracy*, con $p = 1{,}11 \times 10^{-16}$, mientras que el tamaño de lote A, el número de épocas B y las interacciones analizadas presentaron valores de p mayores a 0.025, considerándose que no existe una contribución estadísticamente significativa en el resultado.

En contraste, para el modelo VGG16, el tamaño de lote A y la función de activación C presentaron efectos significativos, con valores de $p = 0{,}00000006$ y $p = 1 \times 10^{-16}$, respectivamente. Asimismo, las interacciones $A \times B$ y $A \times C$ resultaron significativas, mientras que el número de épocas B, y las interacciones $B \times C$ y $A \times B \times C$ no mostraron evidencia estadística suficiente de influencia sobre la *accuracy*.

De acuerdo con estos resultados, se determina que la función de activación C fue el factor de mayor influencia en ambos modelos, siendo *softmax* la de mejor desempeño para EfficientNetB0 y *sigmoid* para VGG16.

Tabla 4. Síntesis de los resultados del diseño factorial para los modelos EfficientNetB0 y VGG16.

Modelo	Factor o interacción	Valor de p	Resultado estadístico	Interpretación
EfficientNetB0	Tamaño de lote (A)	0.171	No significativo	No se encontró evidencia suficiente de influencia sobre la *accuracy*.
	Número de épocas (B)	0.340	No significativo	No presentó un efecto significativo sobre la *accuracy*.
	Función de activación (C)	1.11×10^{-16}	Significativo	Fue el factor de mayor influencia; *softmax* presentó los mejores resultados.
	$A \times B$	0.733	No significativo	No se observó una interacción significativa.
	$A \times C$	0.222	No significativo	No se observó una interacción significativa.
	$B \times C$	0.385	No significativo	No se observó una interacción significativa.
	$A \times B \times C$	0.249	No significativo	No se observó una interacción significativa.
VGG16	Tamaño de lote (A)	0.00000006	Significativo	El tamaño de lote influyó significativamente en la *accuracy*.
	Número de épocas (B)	0.6355	No significativo	No presentó un efecto significativo sobre la *accuracy*.
	Función de activación (C)	1×10^{-16}	Significativo	Fue uno de los factores más relevantes; *sigmoid* presentó el mejor desempeño.

Continúa en la siguiente página

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

Continuación de la Tabla 4.

Modelo	Factor o interacción	Valor de p	Resultado estadístico	Interpretación
	$A \times B$	0.0186	Significativo	Se observó una interacción significativa entre ambos factores.
	$A \times C$	0.000000147	Significativo	Se observó una interacción significativa entre ambos factores.
	$B \times C$	0.121	No significativo	No se observó una interacción significativa.
	$A \times B \times C$	0.566	No significativo	No se observó una interacción significativa.

De manera sintetizada, las configuraciones óptimas que materializaron estos hallazgos se detallan en la Tabla 5, donde se resalta la sinergia entre los niveles de los factores evaluados y la respuesta predictiva de cada red en la variable de salida (*Accuracy*).

Tabla 5. Mejores resultados obtenidos para los modelos EfficientNetB0 y VGG16, antes y después de la optimización con el diseño de experimentos factorial.

Modelo	Accuracy inicial	Accuracy final	Mejor configuración	Factores e interacciones relevantes
EfficientNetB0	92.09 %	94.42 %	Tamaño de lote: 32 Número de épocas: 30 Función de activación: *softmax*	La función de activación fue la variable más relevante;*softmax* presentó el mejor desempeño.
VGG16	81.32 %	90.70 %	Tamaño de lote: 16 Número de épocas: 30 Función de activación: *sigmoid*	Tamaño de lote (A), función de activación (C), interacción $A \times B$ e interacción $A \times C$.

El cumplimiento de la hipótesis inicial, que establecía un umbral de *accuracy* igual o superior a 90 %, no constituye un mero resultado empírico, sino la consecuencia del diseño experimental. Al superar

las barreras del ajuste heurístico, la exploración del espacio de hiperparámetros permitió identificar configuraciones que maximizan la capacidad de generalización de las redes neuronales convolucionales estudiadas. La validación estadística confirma que la selección de variables, particularmente la función de activación y el tamaño de lote, resulta determinante para obtener modelos robustos en tareas de visión artificial en aplicaciones agrícolas.

Si bien ambas arquitecturas fueron entrenadas con el mismo conjunto de datos, la divergencia en sus topologías determinó su desempeño final. EfficientNetB0 demostró una superioridad intrínseca, superando a VGG16 por un margen del 3.72 %. Esta ventaja puede asociarse a su mecanismo de escalado compuesto y a la implementación de bloques *MBConv*, los cuales optimizan el flujo de información y la extracción de características jerárquicas con una mayor eficiencia paramétrica [19].

En contraste, VGG16 se basa en una secuencia relativamente rígida de capas convolucionales de 3 × 3 y contiene un número considerablemente mayor de parámetros que EfficientNet-B0. Por ello, su desempeño puede depender de manera importante de la configuración de entrenamiento y de la capacidad de regularización disponible. Esta interpretación es compatible con estudios de optimización de hiperpará-metros que han demostrado que el efecto del tamaño de lote, la tasa de aprendizaje, el optimizador y otras variables dependen de la arquitectura empleada [21]. Esto queda evidenciado por la significancia estadística de las interacciones entre el tamaño de lote y las demás variables, lo que explica tanto su exactitud basal más reducida como su notable margen de mejora tras la optimización factorial.

El contraste entre ambos modelos estudiados permite determinar cuál es la arquitectura más idónea para la clasificación de pulgones en rosales, y establece un precedente metodológico que demuestra que la adopción de diseños factoriales en el entrenamiento de redes convolucionales mitiga la incertidumbre operativa, garantizando que los futuros despliegues en entornos de agricultura de precisión se sustenten en evidencias cuantitativas y no en la intuición del desarrollador.

También es importante señalar que a diferencia de la búsqueda exhaustiva o aleatoria, este método no se limita a arrojar el valor más alto de Accuracy. Esta diferencia es muy importante, ya que se garantiza que la configuración de parámetros obtenida corresponde a un patrón real y no a un golpe de suerte. Siendo como resultado, un modelo más fácil de reproducir y no se desperdicia tiempo de cómputo en variables que no impactan sobre el rendimiento del modelo.

Con el propósito de mayor claridad a los resultados de los mejores modelos obtenidos con base en la metodología propuesta, la Tabla 6 presenta las matrices de confusión de las configuraciones óptimas de VGG16 y EfficientNetB0. Esta representación complementa las métricas globales de desempeño al mostrar la distribución de las predicciones correctas y los errores de clasificación para cada clase, permitiendo observar con mayor detalle el comportamiento de ambas arquitecturas.

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

Tabla 6. Matrices de confusión obtenidas para los mejores modelos EfficientNetB0 y VGG16.

Clase real	EfficientNetB0 Clase predicha		VGG16 Clase predicha	
	Rosa Plagada	**Rosa Sana**	**Rosa Plagada**	**Rosa Sana**
Rosa Pulgón	102	6	94	14
Rosa Sana	6	101	6	101

La decisión de emplear la *accuracy* global como variable de respuesta en el diseño factorial se sustenta en la distribución equiprobable del conjunto de datos, condición bajo la cual esta métrica refleja la capacidad de generalización de los modelos sin verse distorsionada por la prevalencia de una clase sobre otra. Las matrices de confusión presentadas en la Tabla 6 corroboran esta interpretación, al evidenciar que los errores de clasificación se distribuyen de manera simétrica entre falsos positivos y falsos negativos en ambas arquitecturas.

Conclusiones y futuras lineas de investigación

Con el uso de modelos de redes neuronales convolucionales es factible atender las problemáticas relacionadas con la clasificación de plagas en cultivos de rosales, gracias a las bases datos (imágenes) con los que se entrenaron los modelos EfficientNetB0 y VGG16, logrando una Accuracy de 94.42 % y 90.70 %, respectivamente.

No obstante, las arquitecturas de los modelos deben ser ajustadas a las necesidades específicas de la tarea que se busca atender. Por lo tanto, en este trabajo se utilizó el diseño de experimentos factorial que permitió obtener buenos resultados sin recurrir a la experimentación empírica tradicional de prueba y error, utilizando en su lugar ensayos controlados donde se intercambiaron las uniones de parámetros, tamaño de lote, número de épocas y función de activación. Además, un diseño factorial con base en los niveles de significancia permite tomar elecciones fundamentadas sobre cómo influyen algunos parámetros más que otros y determinar los mejores arreglos de estos para cada modelo.

El proyecto de investigación se enfocó en determinar el mejor arreglo de parámetros a través de un diseño factorial de tres factores eliminando métodos empíricos para las pruebas, de tal modo cada combinación fue evaluada para determinar si esta influía o no en la respuesta de la variable de salida (*accuracy*); para que dichos modelos entrenados fueran capaces de realizar la clasificación de pulgones en rosales.

Los modelos de redes neuronales convolucionales ajustados en sus parámetros para la clasificación de pulgones en rosales muestran una alternativa diferente a los métodos tradicionales que son usados en los cultivos de los municipios del sur del Estado de México, además de ser potencialmente aplicables en otros tipos de cultivo de flores, como lo son: gerbera, margarita, clavel, lirio, entre otras de importancia

Estatal.

Como línea futura de investigación, se plantea la evaluación de las configuraciones óptimas obtenidas en este estudio sobre conjuntos de datos con distribuciones asimétricas que simulen las condiciones reales, donde la prevalencia de plantas sanas suele ser significativamente mayor que la de plantas afectadas. En tales escenarios, será necesario incorporar técnicas de balanceo, como el remuestreo sintético, así como métricas de evaluación más robustas ante el desbalance, con el fin de validar la transferibilidad de los hallazgos factoriales a entornos operativos.

Una vez ajustado los modelos de redes neuronales, la tarea de inspección por el agricultor o ingeniero agrónomo puede ser migrada a sistemas computacionales que pueden reducir tiempos, mano de obra, errores en la inspección de los cultivos o abuso de pesticidas.

Declaración de contribución de autores y colaboradores

José Hernández: Conceptualización, Metodología, Software y Preparación del borrador original. **Daniel Cervantes**: Redacción, Edición e Investigación. **Roberto Alejo**: Supervisión e Investigación. **Federico Del Razo**: Revisión y Edición. **Everardo Granda**: Análisis de resultados y Corrección de estilo.

Referencias

[1] W. A. Lozada-Portilla, M. J. Suarez-Barón, and E. Avendaño-Fernández, "Aplicación de redes neuronales convolucionales para la detección del tizón tardío phytophthora infestans en papa solanum tuberosum," *Revista UDCA Actualidad & Divulgación Científica*, vol. 24, no. 2, pp. 1–9, 2021. DOI: 10.31910/rudca.v24.n2.2021.1917

[2] S. Hemalatha and M. Sangeetha, "Identifying pests in precision agriculture using low-cost image data acquisition," *Brazilian Journal of Biology*, vol. 84, pp. 1–13, 2024. DOI: 10.1590/1519-6984.281671

[3] L. Torres, L. Romero, E. Aguirre, and R. Ferro, "Iot detection system for mildew disease in roses using neural networks and image analysis," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 4, pp. 105–116, 2023. DOI: 10.9781/ijimai.2023.07.001

[4] T. De Cesaro Júnior, R. Rieder, J. R. Di Domênico, and D. Lau, "Insectcv: A system for insect detection in the lab from trap images," *Ecological Informatics*, vol. 67, pp. 1–12, 2022. DOI: https://doi.org/10.1016/j.ecoinf.2021.101516

https://doi.org/10.36677/rici.v4i3.29396 ISSN: 2992-7447 Noviembre 2026 - Febrero 2027 págs. 6- 25

[5] A. L. Velázquez-Torres, N. Callejas-Juárez, and F. E. Martínez-Castañeda, "Análisis de competitividad del sistema de rosa para corte (rosa sp) en el entorno mexiquense," *Terra Latinoamericana*, vol. 40, pp. 1–9, 2022. DOI: 10.28940/terra.v40i0.948

[6] T. Zhang, K. Li, X. Chen, C. Zhong, B. Luo, I. Grijalva, B. McCornack, D. Flippo, A. Sharda, and G. Wang, "Aphid cluster recognition and detection in the wild using deep learning models," *Scientific Reports*, vol. 13, no. 1, pp. 1–12, 2023. DOI: 10.1038/s41598-023-38633-5

[7] R. R. Reyes Recalde, "Evaluación de inductores externos de la activación del sistema inmunológico en el cultivo de rosas (rosa sp.)," Trabajo de maestría, Universidad Técnica de Ambato, Facultad de Ciencias Agropecuarias, Ambato, Ecuador, 2016. [Online]. Available: https://repositorio.uta.edu.ec/jspui/handle/123456789/22666

[8] B. Duman, "Mobile Device-Based Detection System of Diseases and Pests in Rose Plants Using Deep Convolutional Neural Networks and Quantization," *Tarım Bilimleri Dergisi*, vol. 31, no. 2, pp. 302–318, Mar. 2025. DOI: 10.15832/ankutbd.1514972

[9] M. Subramanian, K. Shanmugavadivel, and P. S. Nandhini, "On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13 951–13 968, Aug. 2022. DOI: 10.1007/s00521-022-07246-w

[10] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, D. Deng, and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 1–43, 2023. DOI: https://doi.org/10.1002/widm.1484

[11] D. C. Montgomery, *Design and Analysis of Experiments*, 10th ed. Wiley, 2021, ch. 5–6, pp. 191–315.

[12] Flowers Dataset, "Flowers dataset," Kaggle dataset, Feb. 2021, consultado en la tesis como repositorio de imágenes de libre uso. [Online]. Available: https://www.kaggle.com/datasets/imsparsh/flowers-dataset/data

[13] E. Raff, *Inside Deep Learning: Math, Algorithms, Models*. Manning Publications, 2022, ch. 1, pp. 1–15.

[14] L. Antiga, E. Stevens, H. Huang, and T. Viehmann, *Deep Learning with Pytorch, Second Edition: Training and Applying Deep Learning and Generative AI Models*, 2nd ed. Manning, 2026.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations*, 2015, pp. 1–14. DOI: 10.48550/arXiv.1409.1556

[16] A. B. Naeem, B. Senapati, and A. Zaidi, "Enhancing brain tumor detection from mri-based images through deep transfer learning models," *AI*, vol. 6, no. 12, pp. 1–25, 2025.

[17] D. Pérez-Aguilar, R. Risco-Ramos, and L. Casaverde-Pacherrez, "Transfer learning en la clasificación binaria de imágenes térmicas," *Ingenius. Revista de Ciencia y Tecnología*, vol. 26, pp. 71–86, 2021. DOI: 10.17163/ings.n26.2021.07

[18] D. G. Alonso, "Enfoque liviano para reconocimiento de gestos manuales híbridos con cámaras de profundidad," Ph.D. dissertation, Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), 2020.

[19] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2020, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[20] C. Su and W. Wang, "Concrete cracks detection using convolutional neuralnetwork based on transfer learning," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 10 2020. DOI: 10.1155/2020/7240129

[21] H. Xu, "Comparative Analysis of VGG16 and EfficientNet for Image-Based Cat Breed Classification:," in *Proceedings of the 1st International Conference on Modern Logistics and Supply Chain Management*. Singapore, Singapore: SCITEPRESS - Science and Technology Publications, 2024, pp. 238–242. DOI: 10.5220/0013297100004558