

# Optimización de Modelos de Deep Learning para la Detección de Enfermedades en Hojas de Frijol: Un Enfoque Comparativo y Perspectivas para la Agricultura Sostenible

Edgar Quiroz<sup>a</sup>

<sup>a</sup>Universidad Autónoma de Querétaro, Querétaro, Qro., México

## Abstract

El frijol, un alimento esencial en muchas culturas globales, destaca por su valor nutricional y versatilidad culinaria. La detección temprana de enfermedades en las hojas de frijol es crucial para prevenir pérdidas de cosechas y asegurar recursos alimentarios en la lucha contra la hambruna a nivel mundial.

Este estudio utiliza un conjunto de datos de Kaggle que consta de 1034 imágenes para entrenamiento y 133 imágenes de prueba. Las imágenes se dividen en tres categorías: saludable, con manchas angulares en las hojas y con óxido en los frijoles. Se exploraron diversas arquitecturas de modelos, como una red convolucional de tres capas y enfoques de transfer learning utilizando las arquitecturas LeNet5, ResNet50 y DenseNet121.

Los resultados muestran que la arquitectura DenseNet121 con fine tuning alcanza la mayor precisión, con un 96.09%. Este hallazgo destaca la eficacia de la conectividad densa y la adaptabilidad del fine tuning en la detección de enfermedades en hojas de frijol. Estos resultados tienen implicaciones significativas para la aplicación de técnicas de aprendizaje profundo en la agricultura y la seguridad alimentaria, contribuyendo a la prevención de pérdidas de cultivos y garantizando la disponibilidad de recursos alimentarios.

**Keywords:** Frijol, CNN, Redes, Detección de enfermedades, Agricultura, Seguridad alimentaria

## 1. Introducción

Dentro del grupo de las leguminosas con semillas comestibles, el frijol común se destaca como una de las más importantes. Actualmente, se encuentra distribuido en los cinco continentes y es un componente esencial de la dieta, especialmente en Centroamérica y Sudamérica Amaya et al. (2018). México ha sido reconocido como el centro más probable de su origen o, al menos, como el centro primario de diversificación. El cultivo del frijol se considera uno de los más antiguos, con hallazgos arqueológicos en México y Sudamérica que indican su conocimiento hace unos 5000 años antes de Cristo.

## 2. El frijol y su importancia

Ulloa et al. (2011) Las propiedades nutritivas del frijol están vinculadas a su alto contenido proteico y, en menor medida, a su aporte de carbohidratos, vitaminas y minerales. Dependiendo del tipo de frijol, el contenido de proteínas varía del 14 al 33%, siendo rico en aminoácidos como la lisina (6.4 a 7.6 g/100 g de proteína) y la fenilalanina más tirosina (5.3 a 8.2 g/100 g de proteína), pero con deficiencias en los aminoácidos azufrados de metionina y cisteína. Sin embargo, según evaluaciones biológicas, la calidad de la proteína del frijol cocido puede llegar a ser hasta el 70% en comparación con una proteína de origen animal, a la que se le asigna el 100%.

En cuanto a la contribución de carbohidratos, 100 g de frijol crudo aportan de 52 a 76 g, dependiendo de la variedad, siendo el almidón la fracción más importante. Aunque durante

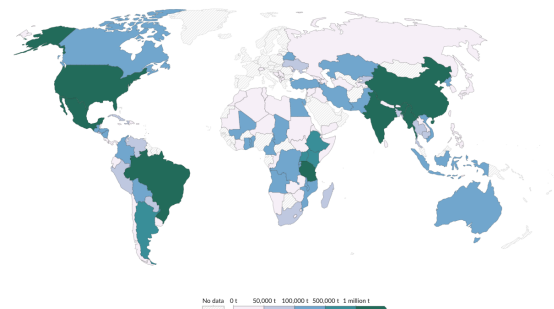


Figure 1: Food and Agriculture Organization of the United Nations (FAO)

la cocción, parte de este se vuelve indisponible al transformarse en el denominado almidón resistente a la digestión.

Dentro de los macronutrientes del frijol, la fracción de lípidos es la más pequeña (1.5 a 6.2 g/100 g), compuesta por una mezcla de acilglicéridos cuyos ácidos grasos predominantes son los mono y poliinsaturados.

## 3. Dataset

El conjunto de datos utilizado en este estudio contiene fotografías de tres estados de lesiones en hojas de leguminosas, con un total de 1034 imágenes en el conjunto de entrenamiento y 133 imágenes en el conjunto de calibración, con un tamaño de datos de 155 MB. Se han incluido las siguientes categorías de estados de lesiones en hojas para su reconocimiento: healthy, angular leaf spot y bean rust.

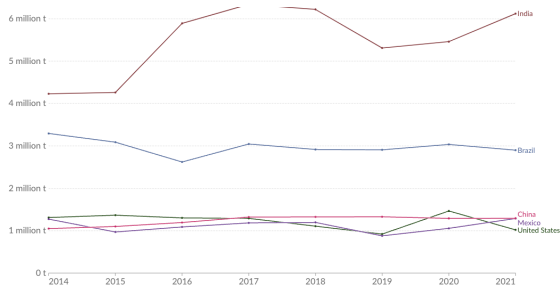


Figure 2: Food and Agriculture Organization of the United Nations (FAO)



Figure 3: Ejemplo de hoja saludable



Figure 4: Ejemplo de angular leaf spot



Figure 5: Ejemplo de bean rust

### 3.1. Data Augmentation

Se establece un flujo de trabajo para cargar y procesar imágenes de un conjunto de datos, aplicar técnicas de aumento de datos durante el entrenamiento y generar lotes de datos para el entrenamiento, la validación y la prueba de un modelo de aprendizaje automático.

#### 3.1.1. Parámetros del Modelo y Data Augmentation:

- `input_shape`: Define el tamaño de las imágenes de entrada al modelo. En este caso, se asume un tamaño de 224x224 píxeles con tres canales de color (RGB).
- `batch_size`: 32.
- `epochs`: 100.

#### 3.1.2. Generador de Datos con Data Augmentation:

Se crea un generador de datos utilizando la clase `ImageDataGenerator` de Keras, que aplica técnicas de aumento de datos durante el entrenamiento. Estas técnicas incluyen la normalización (`rescale=1./255`), rotación, desplazamiento horizontal y vertical, cizallamiento, zoom, volteo horizontal y un modo de llenado para manejar los bordes de la imagen.

## 4. Modelos y Arquitecturas

Se exploraron diversas arquitecturas de redes neuronales para abordar la resolución del problema planteado.

### 4.1. Modelo CNN de 3 Capas

Para este modelo, se adoptó una arquitectura convolucional estándar, donde las capas `Conv2D` se emplean para extraer características cruciales de la imagen. Las capas `MaxPooling2D` reducen las dimensiones, y la capa `Dense` final se encarga de la clasificación. La función de activación `Softmax` en la capa de salida es común en problemas de clasificación multiclase, ya que asigna probabilidades a cada clase. Asimismo, la función de activación `ReLU` se implementa en las capas convolucionales y densas para introducir no linealidades en la red.

- **Capa Conv2D:** Número de Filtros: 32. Tamaño del Kernel: (3, 3). Función de Activación: ReLU. Input Shape: input\_shape (el tamaño de la imagen de entrada).
- **Capa MaxPooling2D:** Tamaño del Pooling: (2, 2). Reduce las dimensiones de la salida de la capa anterior mediante la selección del valor máximo en cada región.
- **Capa Conv2D:** Número de Filtros: 64. Tamaño del Kernel: (3, 3). Función de Activación: ReLU.
- **Capa MaxPooling2D:** Tamaño del Pooling: (2, 2).
- **Capa Conv2D:** Número de Filtros: 128. Tamaño del Kernel: (3, 3). Función de Activación: ReLU.
- **Capa MaxPooling2D:** Tamaño del Pooling: (2, 2).
- **Capa Flatten:** Convierte la salida de las capas convolucionales en un vector unidimensional.
- **Capa Dense:** Número de Neuronas: 128. Función de Activación: ReLU.
- **Capa Dense (Capa de Salida):** Número de Neuronas: 3 (asumiendo un problema de clasificación con tres clases). Función de Activación: Softmax.

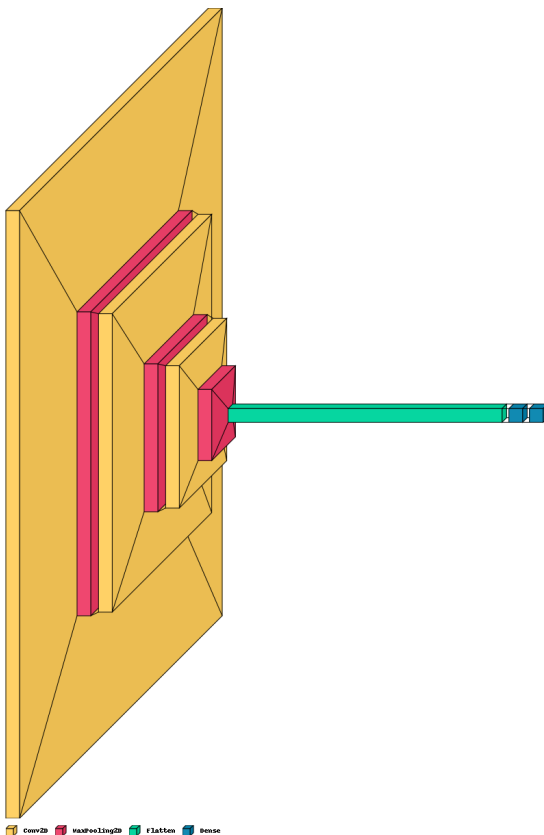


Figure 6: Arquitectura de La red convolucional de 3 capas

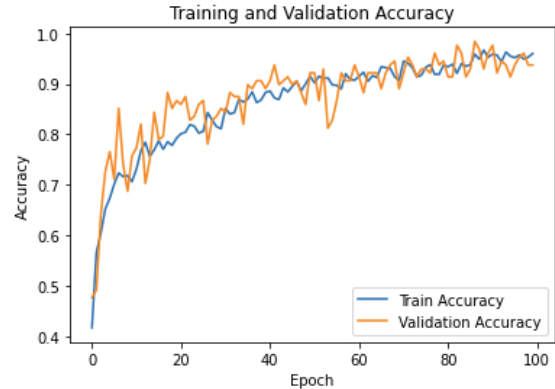


Figure 7: Gráfico de Precisión en Entrenamiento y Validación



Figure 8: Gráfico de Pérdida en Entrenamiento y Validación

#### 4.1.1. Desempeño

El modelo presentó un accuracy del 93.75%.

Los resultados en la base de prueba fueron los siguientes:

#### 4.2. Modelo LeNet-5

LeNet-5 O'Shea and Nash (2015) es una arquitectura de red neuronal convolucional (CNN) diseñada por Yann LeCun y su equipo a principios de la década de 1990. Fue una de las primeras CNNs exitosas y se utilizó para tareas de reconocimiento de dígitos escritos a mano, específicamente en el conjunto de datos MNIST. La arquitectura LeNet-5 fue un hito importante en el desarrollo de las CNN y sentó las bases para muchas arquitecturas posteriores.

La arquitectura LeNet-5 LeCun et al. (2015) consta de varias capas, incluyendo capas convolucionales, capas de submuestreo (pooling), y capas totalmente conectadas. Aquí hay una descripción general de sus componentes:

- **Capas convolucionales:** LeNet-5 utiliza capas convolucionales para extraer características de las imágenes de entrada. Estas capas aplican filtros convolucionales a la imagen para detectar características como bordes y patrones en diferentes regiones.
- **Capas de submuestreo (pooling):** Después de cada capa convolucional, LeNet-5 utiliza capas de submuestreo

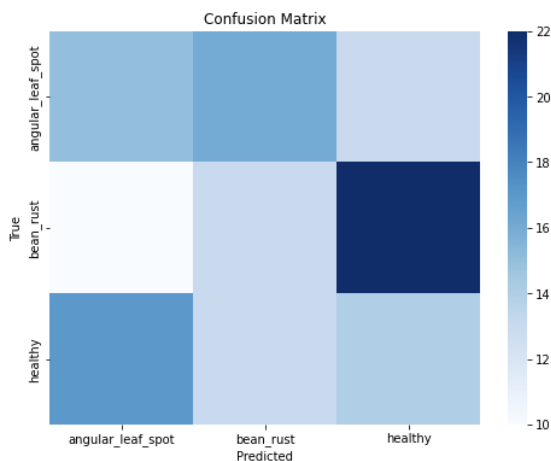


Figure 9: Matriz de Confusión

	precision	recall	f1-score	support
angular_leaf_spot	0.36	0.34	0.35	44
bean_rust	0.31	0.29	0.30	45
healthy	0.29	0.32	0.30	44
accuracy			0.32	133
macro avg	0.32	0.32	0.32	133
weighted avg	0.32	0.32	0.32	133

Figure 10: Reporte de Clasificación

para reducir la resolución espacial y el tamaño de las características extraídas. Esto ayuda a reducir la cantidad de parámetros y a mejorar la eficiencia computacional.

- **Capas totalmente conectadas:** Después de las capas convolucionales y de submuestreo, LeNet-5 utiliza capas totalmente conectadas para realizar la clasificación final. Estas capas están diseñadas para tomar las características extraídas y producir una salida que representa las clases posibles.
- **Funciones de activación:** En cada capa, se utilizan funciones de activación, típicamente la función sigmoide en la versión original de LeNet-5, para introducir no linealidad en la red.

LeNet-5 fue especialmente relevante en su época debido a su capacidad para clasificar dígitos escritos a mano de manera eficiente y precisa. Desde entonces, ha habido muchas variaciones y mejoras en las arquitecturas de CNN, pero LeNet-5 sigue siendo un hito importante en la historia del aprendizaje profundo y las redes neuronales convolucionales.

#### 4.2.1. Desempeño

El modelo presentó un accuracy del 75.0%.

Los resultados en la base de prueba fueron los siguientes:

#### 4.3. Modelo ResNet

ResNet, o Redes Residuales, es una arquitectura de red neuronal profunda que aborda el desafío del entrenamiento de redes cada vez más profundas. Introducida por Kaiming He, et al., en

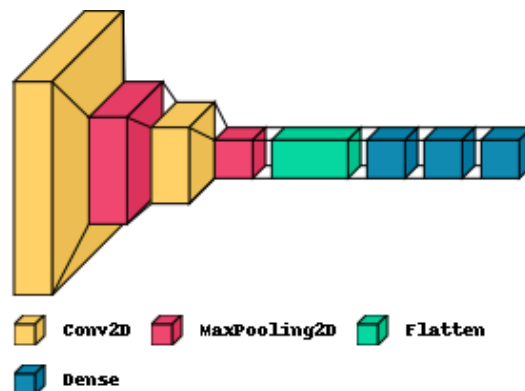


Figure 11: Arquitectura de La red LeNet5

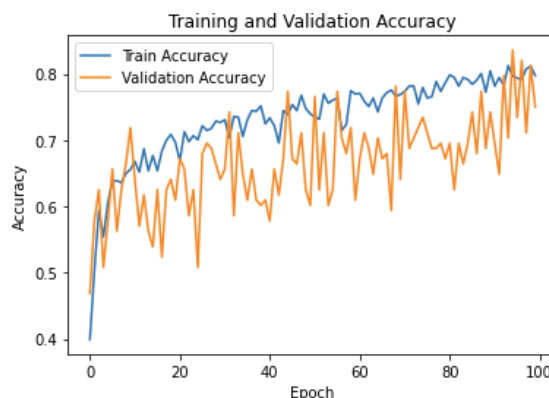


Figure 12: Gráfico de Precisión en Entrenamiento y Validación

2015 Mascarenhas and Agarwal (2021), ResNet ha demostrado ser fundamental para superar las limitaciones de las redes neuronales muy profundas.

La característica distintiva de ResNet es el concepto de "residual learning" o aprendizaje residual. En lugar de intentar aprender directamente la función subyacente, ResNet utiliza conexiones residuales para aprender la diferencia entre la entrada y la salida. Esto se logra mediante la incorporación de "skip connections" o conexiones de salto, que saltan una o más capas.

A continuación, se proporciona una descripción general de los componentes clave de ResNet:

- **Bloques Residuales:** La unidad básica de ResNet es el bloque residual, que consta de dos capas convolucionales y una conexión de salto. La entrada se agrega directamente a la salida, permitiendo que el gradiente fluya fácilmente a través de la red, incluso en entrenamientos profundos.
- **Skip Connections:** Las conexiones de salto, también conocidas como conexiones shortcut, permiten que la información fluya directamente de una capa a otra, evitando así la pérdida de información durante el proceso de aprendizaje.
- **Capas de Batch Normalization:** ResNet utiliza la normalización por lotes para estabilizar y acelerar el entre-



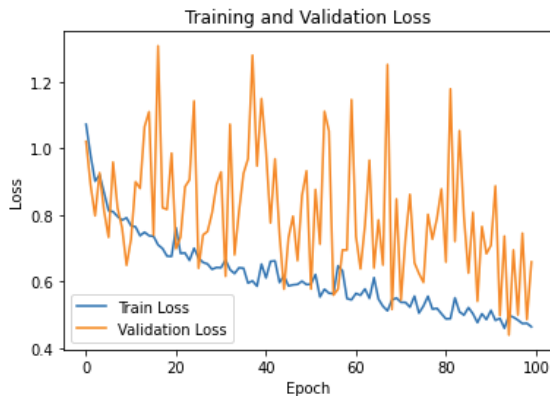


Figure 13: Gráfico de Pérdida en Entrenamiento y Validación

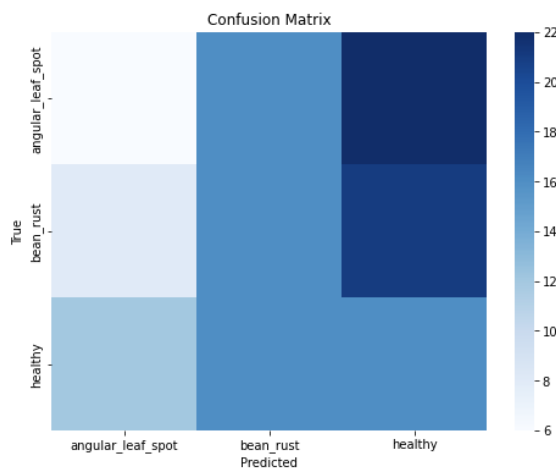


Figure 14: Matriz de Confusión

namiento. Cada bloque residual suele incluir la normalización por lotes después de las capas convolucionales.

- **Función de Activación ReLU:** ResNet utiliza la función de activación ReLU (Rectified Linear Unit) en sus capas, introduciendo no linealidad en la red.
- **Estrategia de Downsampling:** Para ajustar las dimensiones, ResNet emplea capas de pooling y convoluciones con pasos mayores para reducir el tamaño de las representaciones espaciales.
- **Arquitecturas Variantes:** ResNet está disponible en diferentes variantes, como ResNet-50, ResNet-101 y ResNet-152, que varían en la profundidad de la red. Estas variantes han demostrado ser altamente efectivas en una variedad de tareas, incluyendo la clasificación de imágenes y la detección de objetos.

La introducción de ResNet ha sido crucial para permitir el entrenamiento de redes neuronales excepcionalmente profundas, estableciendo un nuevo estándar en el campo del aprendizaje profundo y mejorando significativamente el rendimiento en diversas tareas de visión por computadora.

Classification Report:

	precision	recall	f1-score	support
angular_leaf_spot	0.23	0.14	0.17	44
bean_rust	0.33	0.36	0.34	45
healthy	0.27	0.36	0.31	44
accuracy			0.29	133
macro avg	0.28	0.29	0.28	133
weighted avg	0.28	0.29	0.28	133

Figure 15: Reporte de Clasificación



Figure 16: Arquitectura de La red ResNet

#### 4.3.1. Desempeño

El modelo presentó un accuracy del 57.81%.

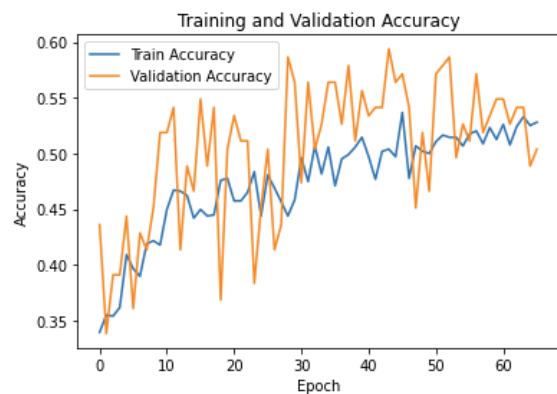


Figure 17: Gráfico de Precisión en Entrenamiento y Validación

Los resultados en la base de prueba fueron los siguientes:

#### 4.4. Modelo DenseNet

DenseNet, o Redes Densamente Conectadas, es una arquitectura de red neuronal profunda que se destaca por su innovador enfoque de conectividad entre capas. Desarrollada por Gao Huang, et al., en 2017 Huang et al. (2016), DenseNet propone una estructura densa en la que cada capa tiene conexiones directas con todas las capas subsiguientes.

A continuación, se proporciona una descripción general de los componentes clave de DenseNet:

- **Bloques Densos:** La unidad básica de DenseNet es el bloque denso, que consta de capas de convolución, normalización por lotes y activación. La característica distintiva es la conexión densa, donde cada capa recibe entradas no solo de su predecesora, sino también de todas las capas anteriores en la secuencia.
- **Conexiones Directas:** DenseNet aprovecha al máximo las conexiones directas mediante la concatenación de las salidas de las capas anteriores como entradas para las capas subsiguientes. Esto facilita el flujo de información a través de la red y mejora el gradiente durante el entrenamiento.



Figure 18: Gráfico de Pérdida en Entrenamiento y Validación

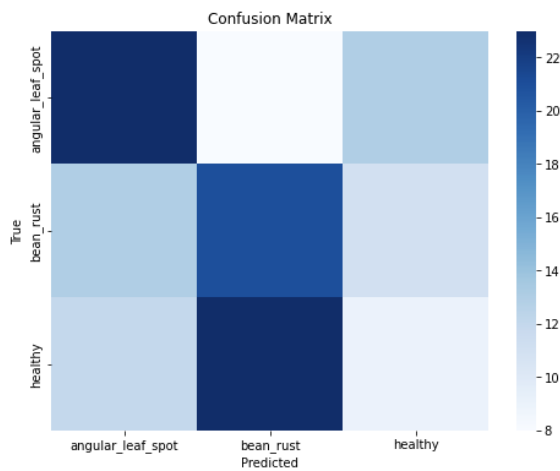


Figure 19: Matriz de Confusión

- **Estructura en Escala:** DenseNet se organiza en bloques densos, y la profundidad de la red se controla mediante la variación del número de bloques y el crecimiento de las capas convolucionales dentro de cada bloque. Esto proporciona una mayor flexibilidad y adaptabilidad a diferentes tareas.
- **Transiciones:** Para reducir las dimensiones y controlar la complejidad computacional, DenseNet incorpora capas de transición que incluyen convoluciones y submuestreo (pooling).
- **Función de Activación ReLU:** Similar a otras arquitecturas de redes neuronales, DenseNet utiliza la función de activación ReLU para introducir no linealidad.
- **Global Average Pooling:** En lugar de capas totalmente conectadas al final de la red, DenseNet a menudo utiliza global average pooling para reducir la dimensionalidad y generar salidas finales.
- **Regularización:** DenseNet incorpora técnicas de regularización, como dropout, para evitar el sobreajuste durante el entrenamiento.

Classification Report:				
	precision	recall	f1-score	support
angular_leaf_spot	0.48	0.52	0.50	44
bean_rust	0.40	0.47	0.43	45
healthy	0.27	0.20	0.23	44
accuracy			0.40	133
macro avg	0.39	0.40	0.39	133
weighted avg	0.39	0.40	0.39	133

Figure 20: Reporte de Clasificación

DenseNet ha demostrado ser eficaz en diversas tareas de visión por computadora, logrando un equilibrio entre eficiencia y rendimiento al aprovechar las conexiones densas para mejorar la propagación de información y reducir la cantidad de parámetros entrenables. Su estructura innovadora ha influido en el diseño de arquitecturas subsiguientes en el campo del aprendizaje profundo. Zhu and Newsam (2017)

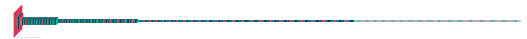


Figure 21: Arquitectura de La red Densenet

#### 4.4.1. Desempeño

El modelo presentó un accuracy del 94.53%.

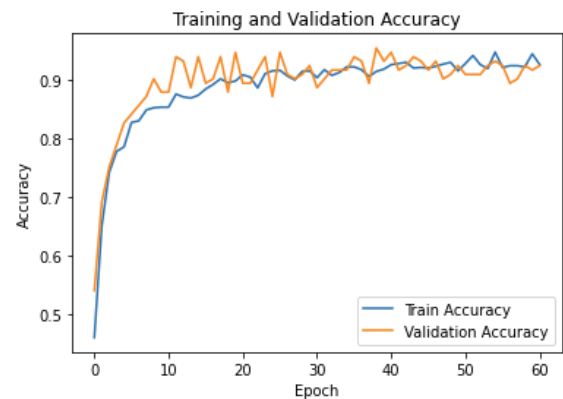


Figure 22: Gráfico de Precisión en Entrenamiento y Validación

Los resultados en la base de prueba fueron los siguientes:

#### 4.5. Modelo DenseNet con Fine Tuning

Se llevó a cabo el reentrenamiento de las 10 capas superiores del modelo DenseNet 121 con el objetivo de mejorar el rendimiento del modelo.

##### 4.5.1. Desempeño

El modelo presentó un accuracy del 96.09%.

Los resultados en la base de prueba fueron los siguientes:

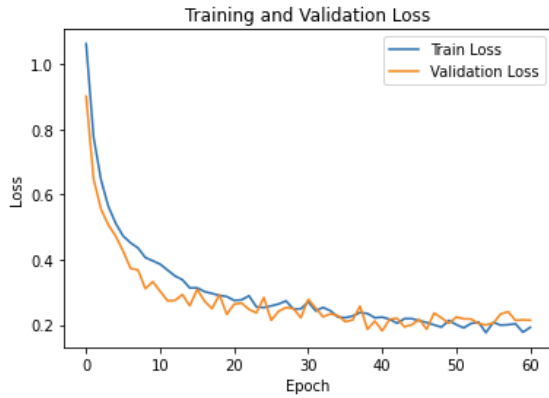


Figure 23: Gráfico de Pérdida en Entrenamiento y Validación

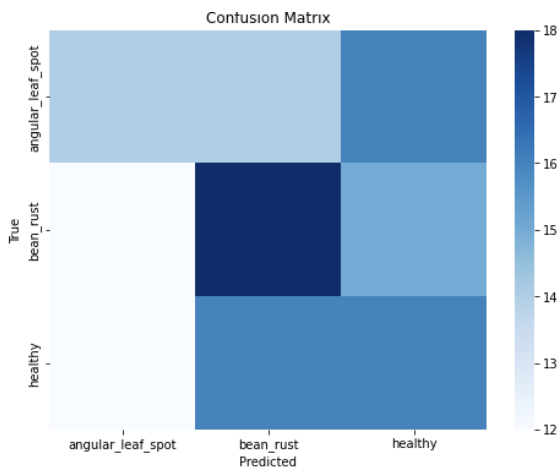


Figure 24: Matriz de Confusión

## 5. Comparación de Modelos

Los resultados de los modelos evaluados se presentan a continuación:

Modelo	Accuracy
CNN 3 capas	93.75%
LeNet 5	75.00%
ResNet 50	57.81%
DenseNet 121	94.53%
DenseNet 121 Fine Tuning	96.09%

Table 1: Comparación de modelos según su precisión (accuracy).

La Tabla 1 resume los resultados obtenidos por cada modelo en términos de precisión. Se observa que el modelo DenseNet 121 Fine Tuning alcanza la precisión más alta, con un 96.09%, lo que indica una mejora significativa después del proceso de fine tuning. Le sigue de cerca el modelo DenseNet 121 con un accuracy del 94.53%. Por otro lado, el modelo ResNet 50 muestra un rendimiento inferior con un accuracy del 57.81%.

Estos resultados sugieren que la arquitectura DenseNet, tanto en su forma original como con fine tuning, supera a las otras arquitecturas consideradas en este estudio. Este hallazgo resalta

Classification Report:				
	precision	recall	f1-score	support
angular_leaf_spot	0.37	0.32	0.34	44
bean_rust	0.38	0.40	0.39	45
healthy	0.34	0.36	0.35	44
accuracy			0.36	133
macro avg	0.36	0.36	0.36	133
weighted avg	0.36	0.36	0.36	133

Figure 25: Reporte de Clasificación

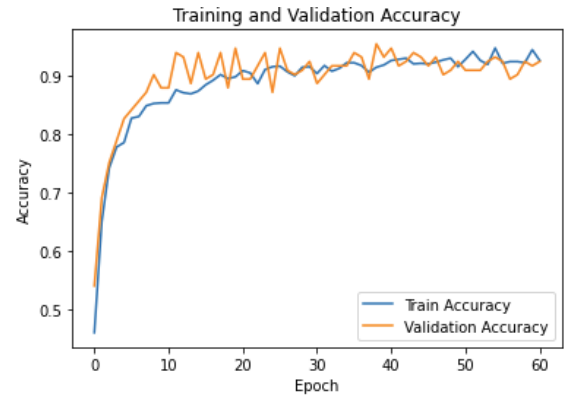


Figure 26: Gráfico de Precisión en Entrenamiento y Validación

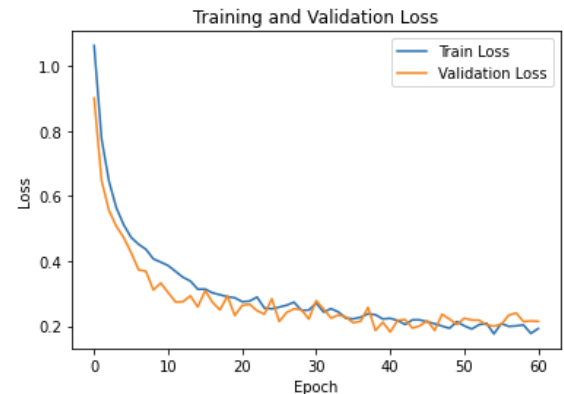


Figure 27: Gráfico de Pérdida en Entrenamiento y Validación

la importancia de la conectividad densa y la capacidad de fine tuning para mejorar la capacidad predictiva del modelo.

## 6. Conclusiones

En este estudio, se abordó el problema de la detección temprana de enfermedades en las hojas de frijol utilizando diversas arquitecturas de redes neuronales convolucionales (CNN). Se exploraron modelos como CNN de 3 capas, LeNet-5, ResNet-50, DenseNet-121 y DenseNet-121 con fine tuning.

Los resultados obtenidos revelan que la arquitectura DenseNet-121 con fine tuning exhibe el rendimiento más destacado, alcanzando un accuracy del 96.09%. Esta mejora sugiere que el fine tuning de las capas superiores de una red preentrenada puede adaptar mejor el modelo a los datos específicos del problema.

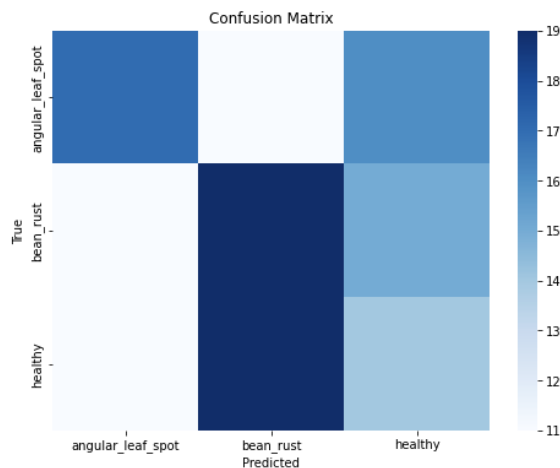


Figure 28: Matriz de Confusión

	precision	recall	f1-score	support
angular_leaf_spot	0.44	0.39	0.41	44
bean_rust	0.39	0.42	0.40	45
healthy	0.31	0.32	0.31	44
accuracy			0.38	133
macro avg	0.38	0.38	0.38	133
weighted avg	0.38	0.38	0.38	133

Figure 29: Reporte de Clasificación

En comparación, modelos como ResNet-50 mostraron un rendimiento inferior, subrayando la importancia de la arquitectura específica en la resolución de problemas particulares. La conectividad densa de DenseNet y la capacidad de adaptación del fine tuning parecen ser beneficiosas para la tarea de detección de enfermedades en hojas de frijol.

Estos resultados tienen implicaciones significativas para la aplicación de técnicas de aprendizaje profundo en la agricultura y la seguridad alimentaria. La capacidad de detectar enfermedades en las etapas tempranas puede contribuir a la prevención de pérdidas de cultivos y garantizar la disponibilidad de recursos alimentarios, siendo crucial en la lucha contra la hambruna a nivel mundial.

En conclusión, este estudio destaca la eficacia de las redes neuronales convolucionales, especialmente DenseNet-121 con fine tuning, en la detección de enfermedades en hojas de frijol. Además, señala la importancia de seleccionar y adaptar la arquitectura del modelo según las características específicas del conjunto de datos y la tarea en cuestión.

## 7. Repositorio de Código Fuente

El código fuente utilizado en este estudio está disponible en GitHub para referencia y reproducción. Puede acceder al repositorio a través del siguiente enlace:

[https://github.com/Eddqzc/Bean\\_leaf\\_deep\\_learning](https://github.com/Eddqzc/Bean_leaf_deep_learning)

Agradecemos cualquier contribución y retroalimentación que pueda mejorar y enriquecer este trabajo.

## References

- Amaya, N., Meldrum, G., Padulosi, S., Cifuentes, R., 2018. Identificación de oportunidades para la comercialización de frijol de la cadena de valor del frijol común .
- Huang, G., Liu, Z., Weinberger, K.Q., 2016. Densely connected convolutional networks. CoRR abs/1608.06993. URL: <http://arxiv.org/abs/1608.06993>, arXiv:1608.06993.
- LeCun, Y., et al., 2015. Lenet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet> 20, 14.
- Mascarenhas, S., Agarwal, M., 2021. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification 1, 96–99.
- O'Shea, K., Nash, R., 2015. An introduction to convolutional neural networks. CoRR abs/1511.08458. URL: <http://arxiv.org/abs/1511.08458>, arXiv:1511.08458.
- Ulloa, J.A., Rosas Ulloa, P., Ramírez Ramírez, J.C., Ulloa Rangel, B.E., et al., 2011. El frijol (phaseolus vulgaris): su importancia nutricional y como fuente de fitoquímicos. CONACYT .
- Zhu, Y., Newsam, S., 2017. Densenet for dense flow , 790–794.