

Clasificación de Especies de Hongos Mediante Arquitecturas Baseline y Transfer Learning (EfficientNet-B0)

Diego Alfaro

27 de noviembre de 2025

Resumen

Este reporte presenta un estudio comparativo exhaustivo entre dos enfoques de aprendizaje profundo (*Deep Learning*) para la clasificación de 169 especies de hongos. Se diseñó e implementó una Red Neuronal Convolutacional (CNN) desde cero (*Baseline*) y se comparó su rendimiento con una arquitectura moderna basada en *Transfer Learning* utilizando EfficientNet-B0. Los experimentos se realizaron utilizando el dataset de clasificación de hongos de Kaggle. Los resultados evidencian la superioridad del *Transfer Learning* en escenarios con alta variabilidad intra-clase, alcanzando una precisión superior al 99 % en comparación con el 15.70 % del modelo Baseline.

Índice

1. Introducción	2
1.1. Objetivos	2
2. Metodología y Dataset	2
2.1. Descripción del Dataset	2
2.2. Preprocesamiento y Aumentación	2
3. Arquitecturas de Modelos	2
3.1. Modelo 1: Baseline CNN (Entrenamiento desde cero)	2
3.1.1. Estructura de la Red	3
3.2. Modelo 2: Transfer Learning (EfficientNet-B0)	3
4. Configuración del Entrenamiento	3
5. Resultados y Discusión	4
5.1. Comparación Cuantitativa	4
5.2. Análisis de Convergencia	4
5.3. Visualización de Predicciones	4
6. Conclusiones	5
7. Referencias	5

1. Introducción

La identificación precisa de especies de hongos es una tarea crítica debido a la similitud visual entre especies comestibles y tóxicas. La visión por computadora ofrece herramientas prometedoras para automatizar este proceso. Este proyecto tiene como objetivo evaluar y comparar la eficacia de entrenar una red neuronal tradicional frente a la adaptación de modelos preentrenados en ImageNet.

1.1. Objetivos

- Implementar una arquitectura CNN personalizada (Baseline) para establecer una línea base de rendimiento.
- Implementar un clasificador basado en EfficientNet-B0 utilizando técnicas de *Transfer Learning*.
- Comparar métricas de precisión, pérdida y tiempo de convergencia entre ambos modelos.

2. Metodología y Dataset

2.1. Descripción del Dataset

Se utilizó el dataset "Mushroom Species Classification" proveniente de Kaggle.

- **Total de clases:** 169 especies distintas.
- **Resolución de entrada:** Las imágenes fueron redimensionadas a 224×224 píxeles.
- **Distribución:** El dataset se dividió en conjuntos de entrenamiento, validación y prueba.

2.2. Preprocesamiento y Aumentación

Para mejorar la generalización del modelo, se aplicaron las siguientes transformaciones utilizando la librería `torchvision`:

- *Resize:* 224×224 .
- *Random Horizontal Flip:* Probabilidad $p = 0,5$.
- *Random Rotation:* $\pm 15^\circ$.
- *Color Jitter:* Brillo, contraste y saturación.
- *Normalización:* Media $[0,485, 0,456, 0,406]$ y Desviación Estándar $[0,229, 0,224, 0,225]$.

3. Arquitecturas de Modelos

3.1. Modelo 1: Baseline CNN (Entrenamiento desde cero)

Se diseñó una arquitectura secuencial clásica con 5 bloques convolucionales para la extracción de características, seguida de un clasificador denso.

3.1.1. Estructura de la Red

La red consta de los siguientes bloques secuenciales:

1. **Bloque 1:** Conv2d ($3 \rightarrow 32$) + BatchNorm + ReLU + MaxPool.
2. **Bloque 2:** Conv2d ($32 \rightarrow 64$) + BatchNorm + ReLU + MaxPool.
3. **Bloque 3:** Conv2d ($64 \rightarrow 128$) + BatchNorm + ReLU + MaxPool.
4. **Bloque 4:** Conv2d ($128 \rightarrow 256$) + BatchNorm + ReLU + MaxPool.
5. **Bloque 5:** Conv2d ($256 \rightarrow 512$) + BatchNorm + ReLU + MaxPool.

El clasificador final (cabezal") está compuesto por:

- AdaptiveAvgPool2d(1, 1)
- Linear ($512 \rightarrow 256$)
- Dropout (0,5)
- Linear ($256 \rightarrow \text{num_classes}$)

Total de parámetros entrenables: **3,803,798**.

3.2. Modelo 2: Transfer Learning (EfficientNet-B0)

Se utilizó la arquitectura EfficientNet-B0, creada mediante la librería `timm`. Esta red utiliza coeficientes compuestos para escalar profundidad, ancho y resolución de manera uniforme.

- **Backbone:** EfficientNet-B0 preentrenado en ImageNet.
- **Estrategia:** Fine-tuning completo de los pesos.
- **Modificación:** Se reemplazó la capa completamente conectada final para adaptarse a las 169 clases del dataset de hongos.

Total de parámetros entrenables: **4,153,614**.

4. Configuración del Entrenamiento

Ambos modelos fueron entrenados utilizando PyTorch en un entorno con aceleración por GPU (NVIDIA Tesla T4).

Hiperparámetro	Configuración
Optimizador	AdamW
Learning Rate	0.001 (Baseline) / 0.0003 (EfficientNet)
Weight Decay	0.01
Función de Pérdida	CrossEntropyLoss (con label smoothing 0.1)
Scheduler	ReduceLROnPlateau (Factor 0.5, Patience 2)
Épocas	15 - 18

Cuadro 1: Configuración de hiperparámetros de entrenamiento.

5. Resultados y Discusión

5.1. Comparación Cuantitativa

Los resultados obtenidos muestran una diferencia drástica entre ambos enfoques.

Modelo	Acc. Entrenamiento	Acc. Validación	Acc. Test
Baseline CNN	23.45 %	15.93 %	15.70 %
EfficientNet-B0	99.87 %	99.23 %	99.07 %

Cuadro 2: Comparación de precisión (Accuracy) final entre modelos.

5.2. Análisis de Convergencia

- **Baseline CNN:** El modelo mostró dificultades para converger. A pesar de tener una capacidad teórica suficiente (3.8M de parámetros), la falta de *priors* visuales y la complejidad del dataset (169 clases) resultaron en un sobreajuste rápido y una generalización pobre. La pérdida de validación se estancó cerca de 4.28.
- **EfficientNet-B0:** Gracias a los pesos preentrenados en ImageNet, el modelo comenzó el entrenamiento con la capacidad de detectar bordes, texturas y formas complejas. Esto permitió una convergencia extremadamente rápida, alcanzando $> 90\%$ de precisión en las primeras épocas y finalizando con una pérdida cercana a 0.04.

5.3. Visualización de Predicciones

Al realizar inferencia en imágenes individuales (como se muestra en el notebook `model_usage.ipynb`):

- El modelo **Baseline** tiende a adivinar entre clases comunes o fallar con baja confianza, distribuyendo la probabilidad entre varias especies incorrectas.
- El modelo **EfficientNet** muestra una confianza alta ($> 95\%$) en la clase correcta, demostrando robustez ante variaciones de iluminación y perspectiva.

6. Conclusiones

El estudio confirma que para tareas de clasificación de grano fino (*fine-grained classification*) con un alto número de clases como la identificación de hongos, el entrenamiento desde cero (*scratch*) es ineficiente sin un dataset masivo.

El uso de **Transfer Learning** con EfficientNet-B0 resultó ser la estrategia óptima, logrando una mejora relativa de más del **83 %** sobre el modelo base. Esto valida la importancia de reutilizar representaciones visuales aprendidas en dominios grandes (ImageNet) para resolver problemas específicos con alta precisión.

7. Referencias

1. Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML.
2. Dataset: Mushroom Species Classification - Kaggle.
3. Documentación de PyTorch y librería `timm`.