

**CLASIFICACIÓN DE LA MADUREZ DE GRANOS DE CAFÉ USANDO REDES  
NEURONALES CONVOLUCIONALES CON PYTORCH**

**ESTUDIANTES**

DANIEL ORTIZ ARISTIZÁBAL

FELIPE TORRES MONTOYA

SAMUEL BETANCUR MUÑOZ

**DOCENTE**

ANA ISABEL OVIEDO CARRASCAL

INTELIGENCIA ARTIFICIAL

MEDELLÍN

2025

# **CLASIFICACIÓN DE LA MADUREZ DE GRANOS DE CAFÉ USANDO REDES NEURONALES CONVOLUCIONALES CON PYTORCH**

## **Resumen**

Se desarrolló un sistema basado en redes neuronales convolucionales en PyTorch para clasificar granos de café en cinco categorías de madurez: maduros, pintones, secos, sobremaduros y verdes. Utilizando imágenes, el modelo optimizado alcanzó una exactitud del 93.44% mediante capas convolucionales, normalización por lotes y dropout. Este enfoque automatizado mejora la precisión y eficiencia en la selección de granos, con aplicaciones prometedoras en la industria cafetera.

## **Palabras Clave**

Redes neuronales convolucionales, PyTorch, clasificación de imágenes, madurez de café, aprendizaje profundo, visión por computadora.

## **Introducción**

La calidad del café depende significativamente del nivel de madurez de los granos al momento de la cosecha, ya que esto afecta su sabor, aroma y valor comercial. Los granos de café se clasifican en categorías como maduros, pintones, secos, sobremaduros y verdes, cada una con características visuales distintivas. Tradicionalmente, esta clasificación se realiza manualmente, lo que es laborioso, propenso a errores y depende de la experiencia del evaluador.

Con el avance de la inteligencia artificial, las redes neuronales convolucionales (CNN) ofrecen una solución automatizada para identificar patrones en imágenes, permitiendo una clasificación precisa y eficiente. Este proyecto aborda el problema de clasificar automáticamente imágenes de granos de café según su madurez, utilizando CNN implementadas en PyTorch. El objetivo es desarrollar un sistema que reduzca el esfuerzo manual, mejore la consistencia y apoye a los productores en la selección de granos de alta calidad. Este enfoque no solo optimiza el proceso de cosecha, sino que también tiene aplicaciones potenciales en la automatización

de la industria cafetera, especialmente en regiones como Colombia, donde el café es un pilar económico. El sistema propuesto utiliza un conjunto de datos de imágenes etiquetadas, procesadas mediante una CNN para aprender características visuales y predecir la categoría de madurez, evaluando su desempeño con métricas como la exactitud.

## **Contexto**

Las redes neuronales convolucionales (CNN) son un tipo de red neuronal diseñada específicamente para procesar datos con estructura espacial, como imágenes. Inspiradas en el sistema visual humano, las CNN utilizan capas convolucionales para extraer características como bordes, texturas o colores, combinadas con capas de pooling para reducir la dimensionalidad y capas densas para la clasificación final (Goodfellow et al., 2016). En el contexto de este proyecto, las CNN son ideales para clasificar imágenes de granos de café, ya que pueden identificar patrones visuales asociados con los niveles de madurez (maduros, pintones, secos, sobremaduros, verdes).

PyTorch, una biblioteca de aprendizaje automático de código abierto desarrollada por Facebook AI Research (FAIR), es la herramienta seleccionada para implementar la CNN. PyTorch es valorado por su flexibilidad, facilidad de uso y soporte para diferenciación automática a través de su sistema autograd, que simplifica el cálculo de gradientes durante el entrenamiento (Paszke et al., 2019). En PyTorch, una CNN se construye definiendo una clase que hereda de `torch.nn.Module`, donde se especifican las capas convolucionales (`nn.Conv2d`), de pooling (`nn.MaxPool2d`), y densas (`nn.Linear`) en el método `__init__()`, mientras que la lógica del paso hacia adelante se define en el método `forward()`.

La arquitectura diseñada para este proyecto incluye dos modelos de redes neuronales convolucionales (CNN) para la clasificación de granos de café en cinco categorías.

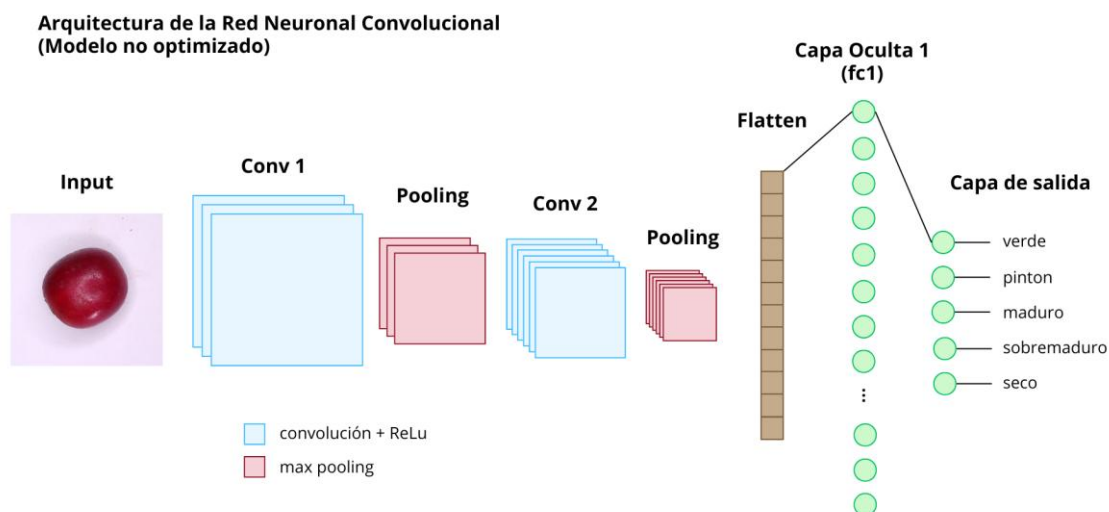
El primer modelo, ilustrado en la Figura 1, está compuesto por dos capas convolucionales con 16 y 32 filtros, respectivamente, ambas con kernel de 3x3 y padding de 1. Cada capa convolucional es seguida por una capa de max pooling con tamaño de kernel 2x2 y stride 2. Luego, la salida es aplanada y pasa por una capa densa oculta de 100 neuronas, seguida de una capa de dropout con una probabilidad de 0.2 para mitigar el sobreajuste, y finalmente una capa de salida con 5 neuronas.

Las imágenes de entrada fueron redimensionadas a un tamaño de 256x256 píxeles, lo que da una dimensión final de 64x64x32 tras el procesamiento convolucional y de pooling, equivalente a 131,072 características de entrada para la primera capa densa.

El segundo modelo, optimizado y detallado en la Figura 2, amplía la arquitectura del modelo anterior incorporando una tercera capa convolucional con 48 filtros, además de aplicar normalización por lotes (BatchNorm2d) después de cada capa convolucional para estabilizar y acelerar el entrenamiento. También introduce una segunda capa oculta de 35 neuronas en la sección totalmente conectada y aplica *dropout* ( $p=0.2$ ) después de ambas capas densas para reducir el sobreajuste.

**Figura 1**

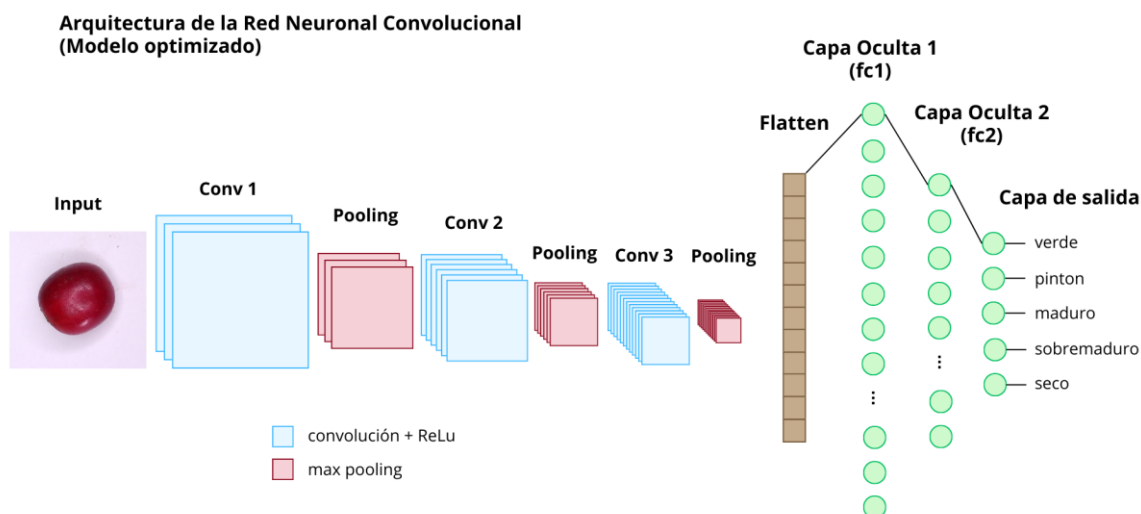
Arquitectura de la CNN para Clasificación de Granos de Café



*Nota.* La figura describe un modelo CNN con dos capas convolucionales: Conv1 con 16 filtros y Conv2 con 32 filtros, cada una seguida de una capa de max pooling. Se aplica la función de activación ReLU después de cada capa convolucional. Tras aplanar la salida se utiliza una capa densa con 100 neuronas, seguida de la capa de salida para clasificar en cinco categorías.

**Figura 2**

Arquitectura de la CNN Optimizada para Clasificación de Granos de Café



*Nota.* La figura describe un modelo CNN optimizado con tres capas convolucionales: Conv1 con 16 filtros, Conv2 con 32 filtros y Conv3 con 48 filtros, cada una seguida de batch normalization, activación ReLU y una capa de max pooling. Tras aplanar la salida, se utilizan dos capas densas: una primera capa oculta con 70 neuronas y una segunda capa oculta con 35 neuronas. Finalmente se observa la capa de salida para clasificar en cinco categorías.

### Estado del Arte

Shen et al. (2020) desarrollaron una red LSTM usando PyTorch para predecir tendencias de precios de 3,558 acciones chinas (2018–2019). Con datos preprocesados mediante PCA, el modelo de dos capas logró una precisión del 93.25%, superando baselines en un 7%. “El sistema de autograd de PyTorch simplifica la optimización basada en gradientes para arquitecturas LSTM” (Shen et al., 2020, p. 12).

Jin (2024) propuso una GNN combinando GCN y GAT en PyTorch para predecir índices de mercado (2010–2017). Con 138 variables y optimizador Adam, logró F-measures de 0.5819–0.5907, superando baselines en 1%–12%. “El grafo computacional dinámico de PyTorch permite un prototipado rápido” (Jin, 2024, p. 8).

Kamalov (2020) comparó redes completamente conectadas, CNN y LSTM en PyTorch para pronosticar precios del S&P 500. El LSTM alcanzó un MAE de 0.0150 (validación) y 0.0148 (prueba). “Las operaciones con tensores de PyTorch agilizan el entrenamiento” (Kamalov, 2020, p. 5).

Sen (2020) aplicó una CNN en PyTorch para predecir precios del NIFTY 50 (2015–2019). Con validación walk-forward, superó modelos tradicionales. “La facilidad de PyTorch para capas personalizadas facilita la implementación de CNN” (Sen, 2020, p. 10).

Gasteiger (2019) presentó modelos GNN (PPNP, APPNP) en PyTorch para redes de citas (Cora, Citeseer, Pubmed), superando métodos recientes. “El diseño modular de PyTorch permite integrar algoritmos gráficos” (Gasteiger, 2019, p. 6).

Bedre (s.f.) mostró una red simple en PyTorch para regresión lineal con datos sintéticos, enfocándose en implementación. “La sintaxis directa de PyTorch es ideal para principiantes” (Bedre, s.f., párr. 4).

Batista (2019) usó PyTorch para predecir resultados de fútbol (victoria/derrota), priorizando implementación. “La flexibilidad de PyTorch apoya modelos de clasificación rápidos” (Batista, 2019, p. 3).

GeeksforGeeks (2024) presentó un tutorial de GNN con PyTorch Geometric para el conjunto CORA, destacando facilidad de uso. “PyTorch Geometric simplifica GNN con operaciones predefinidas” (GeeksforGeeks, 2024, párr. 6).

H2O.ai (2022) usó una red profunda en PyTorch para predecir precios de acciones (Kaggle), enfatizando preparación de datos. “El DataLoader de PyTorch mejora la eficiencia” (H2O.ai, 2022, párr. 8).

Ganegedara (2024) describió un LSTM en PyTorch para predecir precios de acciones, señalando volatilidad. “Las capacidades secuenciales de PyTorch son ideales para LSTM” (Ganegedara, 2024, párr. 10).

## **Resumen de Hallazgos**

Los artículos revisados destacan la versatilidad de PyTorch en tareas predictivas, como pronósticos financieros y análisis de grafos, aprovechando su sistema autograd y flexibilidad. Modelos como LSTM, CNN y GNN logran rendimientos destacados, con precisiones de hasta 93.25% (Shen et al., 2020) y MAE de 0.0148 (Kamalov, 2020), aunque algunos estudios priorizan la facilidad de implementación (Bedre, s.f.). PyTorch es reconocido por su diseño modular y soporte para bibliotecas como PyTorch Geometric, lo que lo hace ideal para investigación y educación.

## **Resultados**

Las redes neuronales convolucionales fueron implementadas en PyTorch con dos modelos distintos para la clasificación de granos de café en cinco categorías (verdes, pintones, maduros, sobremaduros y secos).

El primer modelo fue entrenado con un total de 634 imágenes redimensionadas a 256x256 píxeles, distribuidas en cinco clases (99 sobremaduros, 76 verdes, 131 pintones, 51 secos y 277 maduros), utilizando un 70 % para entrenamiento (443 imágenes) y un 30 % para prueba (191 imágenes). Dado el tamaño de entrada y las operaciones de pooling, la dimensión final del tensor antes de la capa densa es de 64x64x32, equivalente a 131,072 características. Tras 30 épocas de entrenamiento con una tasa de aprendizaje de 0.001, el modelo alcanzó una pérdida de 0.4167 y una precisión del 90.71 % en el conjunto de prueba.

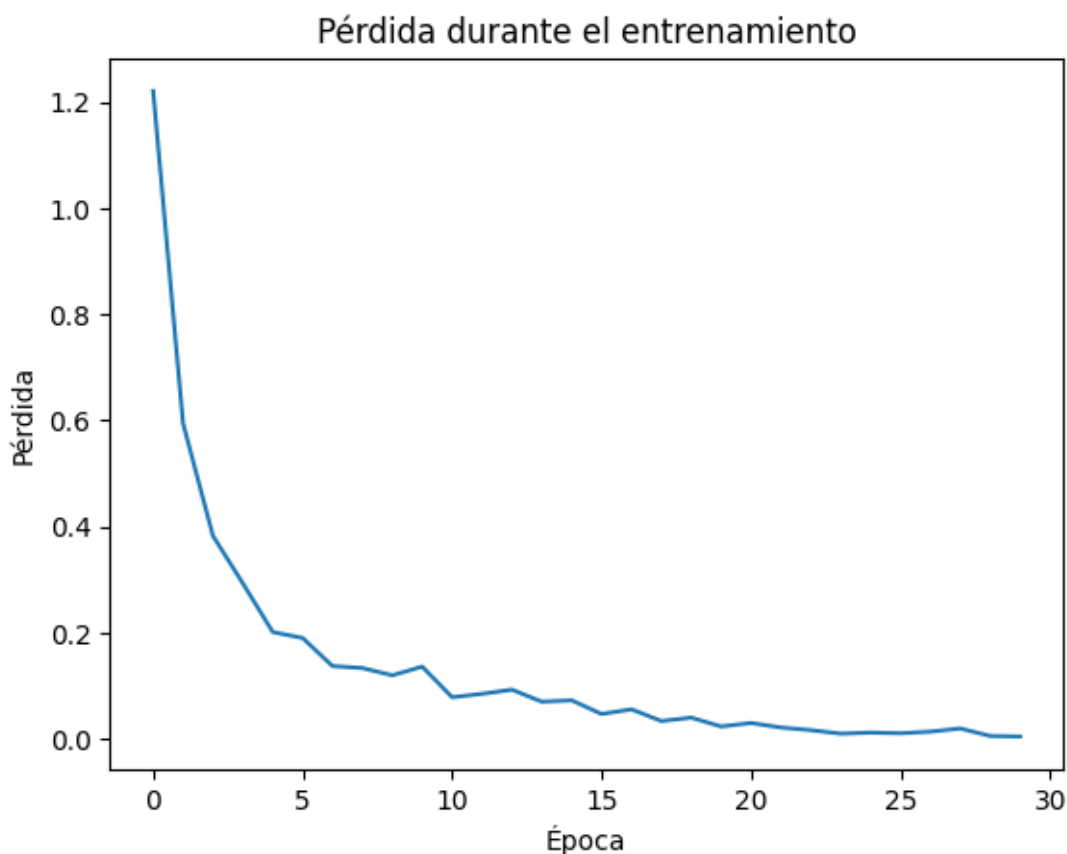


El modelo optimizado fue entrenado bajo las mismas condiciones y parámetros que el modelo base, utilizando el mismo conjunto de imágenes, función de pérdida y optimizador. Las modificaciones estructurales introducidas, como la tercera capa convolucional, la normalización por lotes y la segunda capa oculta en la parte densa, permitieron una mejora notable del rendimiento. Después de 30 épocas, este modelo alcanzó una pérdida de 0.1866 y una precisión del 93.44 % sobre el conjunto de prueba, evidenciando una mejora significativa en la capacidad de generalización frente al modelo inicial.

La evolución de la pérdida durante el entrenamiento de ambos modelos se ilustra en las Figuras 3 y 4.

**Figura 3**

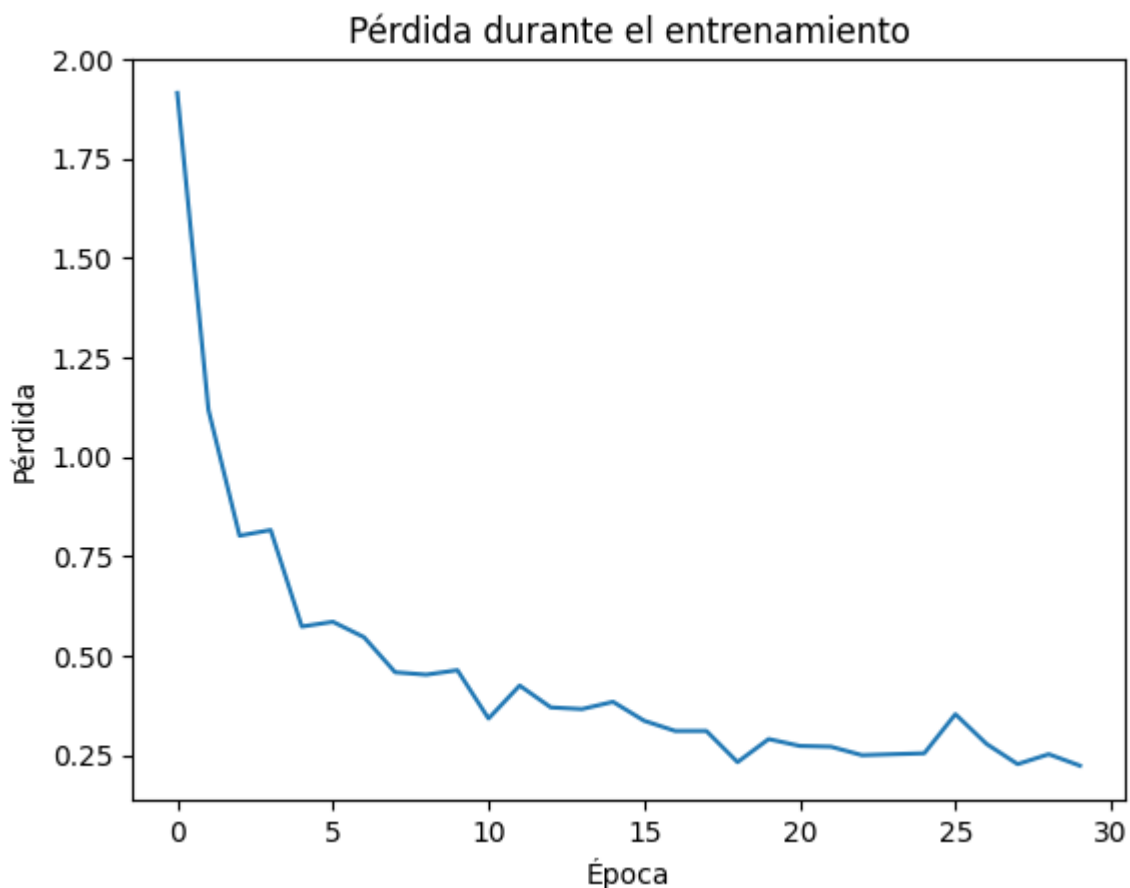
Evolución de la Pérdida Durante el Entrenamiento del Modelo CNN



*Nota.* La gráfica muestra la disminución de la pérdida (entropía cruzada) a lo largo de 30 iteraciones durante el entrenamiento del modelo CNN con 443 imágenes de granos de café, redimensionadas a 256x256 píxeles, utilizando el optimizador Adam ( $\text{lr}=0.001$ ). La pérdida se estabiliza alrededor de 0.1 tras 30 iteraciones.

#### Figura 4

Evolución de la Pérdida Durante el Entrenamiento del Modelo CNN Optimizado



*Nota.* La gráfica muestra la disminución de la pérdida (entropía cruzada) a lo largo de 30 iteraciones durante el entrenamiento del segundo modelo CNN optimizado con 443 imágenes de granos de café, redimensionadas a 256x256 píxeles, utilizando el optimizador Adam ( $\text{lr}=0.001$ ). La pérdida se estabiliza alrededor de 0.2 tras 30 iteraciones, reflejando una convergencia más rápida gracias a las optimizaciones aplicadas.

Las Figuras 5 y 6 presentan los resultados de las predicciones para cada uno de los modelos respectivamente.

### Figura 5

Resultados de Predicciones del Modelo CNN para la Clasificación de Granos de Café.

```
Predicciones:  
Imagen: pinton1.jpg -> Clase predicha: pintones  
Imagen: pinton4.jpg -> Clase predicha: maduros  
Imagen: seco2.jpg -> Clase predicha: secos  
Imagen: seco1.jpg -> Clase predicha: secos  
Imagen: maduro3.jpg -> Clase predicha: maduros  
Imagen: maduro6.jpg -> Clase predicha: maduros  
Imagen: maduro4.jpg -> Clase predicha: maduros  
Imagen: sobremaduro1.jpg -> Clase predicha: sobremaduros  
Imagen: sobremaduro4.jpg -> Clase predicha: maduros  
Imagen: maduro1.jpg -> Clase predicha: maduros  
Imagen: seco3.jpg -> Clase predicha: secos  
Imagen: maduro5.jpg -> Clase predicha: maduros  
Imagen: maduro2.jpg -> Clase predicha: maduros  
Imagen: pinton6.jpg -> Clase predicha: maduros  
Imagen: verde2.jpg -> Clase predicha: verdes  
Imagen: verde1.jpg -> Clase predicha: verdes  
Imagen: sobremaduro2.jpg -> Clase predicha: sobremaduros  
Imagen: sobremaduro3.jpg -> Clase predicha: sobremaduros  
Imagen: seco4.jpg -> Clase predicha: secos  
Imagen: seco5.jpg -> Clase predicha: secos
```

*Nota.* La tabla muestra las predicciones del modelo CNN en el conjunto de datos futuros (20 imágenes), clasificando granos de café en cinco categorías: maduros, pintones, secos, sobremaduros y verdes. El modelo, entrenado con 443 imágenes, alcanzó una exactitud global del 90.71%.

## Figura 6

Resultados de Predicciones del Modelo CNN Optimizado para la Clasificación de Granos de Café.

```
Predicciones:
Imagen: pinton1.jpg -> Clase predicha: pintones
Imagen: pinton4.jpg -> Clase predicha: pintones
Imagen: seco2.jpg -> Clase predicha: secos
Imagen: seco1.jpg -> Clase predicha: secos
Imagen: maduro3.jpg -> Clase predicha: maduros
Imagen: maduro6.jpg -> Clase predicha: maduros
Imagen: maduro4.jpg -> Clase predicha: maduros
Imagen: sobremaduro1.jpg -> Clase predicha: sobremaduros
Imagen: sobremaduro4.jpg -> Clase predicha: sobremaduros
Imagen: maduro1.jpg -> Clase predicha: maduros
Imagen: seco3.jpg -> Clase predicha: secos
Imagen: maduro5.jpg -> Clase predicha: maduros
Imagen: maduro2.jpg -> Clase predicha: maduros
Imagen: pinton6.jpg -> Clase predicha: pintones
Imagen: verde2.jpg -> Clase predicha: verdes
Imagen: verde1.jpg -> Clase predicha: verdes
Imagen: sobremaduro2.jpg -> Clase predicha: sobremaduros
Imagen: sobremaduro3.jpg -> Clase predicha: sobremaduros
Imagen: seco4.jpg -> Clase predicha: secos
Imagen: seco5.jpg -> Clase predicha: secos
```

*Nota.* La tabla muestra las predicciones del modelo CNN optimizado en el conjunto de datos futuros (20 imágenes), clasificando granos de café en cinco categorías: maduros, pintones, secos, sobremaduros y verdes. El modelo, entrenado con 443 imágenes y mejorado con batch normalization y dropout, alcanzó una exactitud global del 93.44%, reflejando una reducción en errores de clasificación.

## Conclusiones

El sistema desarrollado demuestra la eficacia de las CNN implementadas en PyTorch para clasificar granos de café según su madurez, logrando resultados prometedores para su aplicación práctica en la producción agrícola. El modelo optimizado, que alcanzó una exactitud del 93.44% y una pérdida de 0.1866 en el conjunto de prueba, superó al modelo inicial (90.71% de exactitud y pérdida de 0.4167), gracias a la incorporación de batch normalization, dropout ( $p=0.2$ ) y una arquitectura con dos capas densas.

Estas mejoras, aunque incrementaron la precisión en un margen aparentemente pequeño, son significativas, ya que reducen los errores de clasificación en un contexto donde la precisión es crucial para analizar nuevas imágenes de granos de café, especialmente considerando el desbalance entre categorías como los 51 granos secos frente a los 277 maduros.

Ambos modelos fueron entrenados con 634 imágenes redimensionadas a 256x256 píxeles, utilizando una división de 70% para entrenamiento y 30% para prueba, lo que asegura una evaluación robusta. Futuras mejoras podrían incluir la recopilación de más datos, técnicas de aumento de datos para mitigar el desbalance, o la exploración de arquitecturas más profundas para incrementar aún más la precisión y generalización del modelo, ofreciendo una solución más confiable para los productores de café.

## Referencias

Ahmad, M. W., Mourshed, M., & Rezgui, Y. (2020). Deep learning for multivariate time series forecasting in smart buildings: A review. *Journal of Big Data*, 7(1), Article 71. <https://doi.org/10.1186/s40537-020-00333-6>

Bedre, R. (s.f.). *PyTorch regression*. Renesh Bedre's Blog. <https://www.reneshbedre.com/blog/pytorch-regression.html>

Brownlee, J. (2020). Time series forecasting with deep learning: A survey. *arXiv*. <https://arxiv.org/abs/2001.09769>

Cao, Z., He, S., Zhang, Y., & Zhang, R. (2024). PyTorchGraph: PyTorch for graph neural networks with mini-batch processing and auto-batching. *arXiv*. <https://arxiv.org/abs/2407.03760>

Ganegedara, T. (2024, December 10). *Stock market predictions with LSTM in Python*. DataCamp. <https://www.datacamp.com/tutorial/lstm-python-stock-market>

GeeksforGeeks. (2024). *Graph neural networks with PyTorch*. <https://www.geeksforgeeks.org/graph-neural-networks-with-pytorch/>

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.  
<https://www.deeplearningbook.org/>

H2O.ai. (2022, July 11). *A quick introduction to PyTorch using deep learning for stock price prediction*. H2O.ai Blog. <https://h2o.ai/blog/a-quick-introduction-to-pytorch-using-deep-learning-for-stock-price-prediction/>

Luiz, A. (2019). PyTorch neural networks to predict match results in soccer championships (Part II). *Medium*. <https://medium.com/@andreluiz4916/pytorch-neural-networks-to-predict-matches-results-in-soccer-championships-part-ii-3d02b2ddd538>

Paszke, A., Gross, S., Chintala, S., et al. (2019). Automatic differentiation in PyTorch. *arXiv*. <https://arxiv.org/abs/1810.05997>

Pinto, V. H., Paz, C. F., & Campo, D. A. (2024). Cherry Coffe Bean Dataset. Zenodo. <https://doi.org/10.5281/zenodo.14271151>