

GreenMachine: Clasificación automática de plantas medicinales con aprendizaje profundo

Jholman Dasney Meza Pasinga, Juan Camilo Perdomo Soto, Campos Herney Tulcan Cuasapud

Universidad Nacional de Colombia, Sede Manizales

Departamento de Ingeniería Eléctrica, Electrónica y Computación

Curso: Teoría de Aprendizaje de Máquina (2025-1)

Curso: Procesamiento Digital de Imágenes (2025-1)

Correos: jmezap@unal.edu.co, jperdomos@unal.edu.co, ctulcan@unal.edu.co

Resumen—Este artículo presenta el diseño y desarrollo de GreenMachine, un sistema de clasificación de plantas medicinales basado en modelos de aprendizaje profundo. Utilizando imágenes capturadas por el usuario, el sistema identifica la especie vegetal y despliega información curativa relevante. Se emplea un modelo YOLOv8 entrenado con datos de Roboflow y exportado a TorchScript, integrando una interfaz desarrollada en Python con Tkinter para su uso en tiempo real. La propuesta busca preservar el conocimiento etnobotánico, facilitar su acceso y fomentar el uso responsable de la medicina tradicional en América Latina.

Index Terms—Aprendizaje profundo, visión por computador, plantas medicinales, YOLOv8, TorchScript, etnobotánica.

I. INTRODUCCIÓN

El uso de plantas medicinales representa una de las formas más antiguas de intervención terapéutica en América Latina con raíces profundamente arraigadas en el conocimiento ancestral de diversas comunidades. A pesar de su valor cultural, ecológico y sanitario, el saber tradicional sobre las propiedades curativas de las especies vegetales se encuentra en riesgo de desaparecer debido a factores como la urbanización acelerada, la falta de documentación sistemática y la desconexión intergeneracional.

Los avances en visión por computador y aprendizaje automático (ML por sus siglas en inglés) han permitido desarrollar soluciones computacionales capaces de identificar patrones complejos en imágenes. Estas tecnologías han sido aplicadas exitosamente en áreas como diagnóstico médico, agricultura inteligente y clasificación de especies vegetales mostrando gran potencial en escenarios donde se requiere una identificación precisa basada en características visuales.

Este proyecto propone una solución computacional denominada *Green Machine* orientada a la identificación automática de plantas medicinales latinoamericanas mediante imágenes capturadas por cámaras convencionales. El sistema se fundamenta en modelos de *deep learning*, en particular la arquitectura YOLOv8 en modo clasificación y se complementa con una interfaz gráfica interactiva que provee información curativa relevante asociada a cada especie reconocida, de esta forma la iniciativa busca preservar y difundir el conocimiento etnobotánico mediante herramientas digitales accesibles, promoviendo además la apropiación tecnológica en contextos educativos, rurales y científicos.

II. PLANTEAMIENTO DEL PROBLEMA

A pesar de la riqueza botánica de América Latina existe una escasez de herramientas digitales especializadas en el reconocimiento de plantas medicinales. La falta de acceso a expertos en botánica o herbolaria, especialmente en contextos rurales o educativos, dificulta la preservación y transmisión del conocimiento ancestral. El problema se formula como: *¿Es posible desarrollar un sistema computacional basado en aprendizaje profundo que identifique automáticamente plantas medicinales a partir de imágenes y proporcione información curativa relevante?*

III. ESTADO DEL ARTE

Se revisaron tres desarrollos relevantes:

III-A. Leafsnap (2011)

Sistema de visión computacional que emplea SVM para identificar árboles a partir de fotos de hojas. Fue pionero en aplicaciones móviles de identificación vegetal. Limitado a especies del noreste de EE.UU.

III-B. PlantDoc (2020)

Proyecto enfocado en la detección de enfermedades en plantas usando CNNs. Entrenado con imágenes en campo, pero sin vínculo con identificación curativa ni especies nativas de América Latina.

III-C. Pl@ntNet

Aplicación colaborativa basada en aprendizaje profundo y retroalimentación de usuarios. Altamente escalable y de cobertura global, pero no especializada en plantas medicinales ni ofrece información terapéutica.

IV. OBJETIVO

IV-A. Objetivo General

Diseñar e implementar un sistema computacional basado en técnicas de aprendizaje profundo que permita la identificación automática de plantas medicinales latinoamericanas a partir de imágenes digitales, proporcionando al usuario información terapéutica verificada sobre cada especie reconocida.

IV-B. Objetivos Específicos

- Construir y curar un conjunto de datos etiquetado con imágenes de especies vegetales medicinales representativas del contexto latinoamericano, estructurado bajo el formato de clasificación de imágenes.
- Entrenar un modelo de clasificación de imágenes utilizando la arquitectura YOLOv8 en modo *classification*, optimizando su desempeño mediante técnicas de ajuste de hiperparámetros y validación cruzada.
- Integrar el modelo entrenado en una interfaz interactiva (local o web embebida) que relacione cada predicción con sus correspondientes propiedades terapéuticas, usos tradicionales y advertencias, permitiendo una experiencia informativa y educativa.
- Evaluar cuantitativamente el rendimiento del sistema mediante métricas estándar de clasificación (precisión, exactitud, F1-score) y analizar su aplicabilidad práctica en entornos reales.

V. METODOLOGÍA

La metodología implementada en este trabajo contempla una cadena de etapas que van desde la adquisición y curación de datos hasta el entrenamiento, evaluación y despliegue del modelo de clasificación. A continuación, se detallan cada una de estas fases:

- **Curación del conjunto de datos:** Se utilizó la plataforma Roboflow para compilar un dataset de 3251 imágenes correspondientes a 11 clases de plantas medicinales latinoamericanas. Las imágenes fueron recolectadas de múltiples fuentes digitales y clasificadas manualmente por especie, asegurando una representación visual adecuada de cada categoría.
- **Preprocesamiento y aumentación de datos:** Las imágenes fueron reescaladas a una resolución estándar de 640×640 píxeles mediante estiramiento (*resize: stretch*). Se aplicaron técnicas de aumentación de datos incluyendo rotación aleatoria ($\pm 15^\circ$), zoom (hasta 30 %), *shear* horizontal y vertical (hasta $\pm 14^\circ$), ajuste de brillo (± 37 %), exposición (± 15 %), desenfoque (hasta 2.9px) y ruido gaussiano (hasta 1.84 %). Estas transformaciones fueron configuradas para generar tres vistas aumentadas por cada imagen base, incrementando la robustez del modelo frente a variaciones ambientales.
- **División del conjunto de datos:** Se realizó una partición estratificada en tres subconjuntos: 87 % de las imágenes (2838) para entrenamiento, 9 % (279) para validación y 4 % (134) para prueba. Esta distribución fue determinada automáticamente por Roboflow para garantizar balance y cobertura interclase.
- **Modelo seleccionado:** Se empleó la arquitectura YOLOv8s en su modalidad de clasificación (*YOLOv8-cl*s), la cual combina precisión, velocidad de inferencia y eficiencia computacional. El entrenamiento se realizó por transferencia utilizando pesos preentrenados, con un ciclo de optimización mediante descenso del gradiente

estocástico y criterio de parada anticipada (*early stopping*).

- **Evaluación del desempeño:** El modelo fue evaluado utilizando métricas estándar para tareas de clasificación multiclase: exactitud global (*accuracy*), precisión, *recall*, F1-score y matriz de confusión. Estas métricas fueron calculadas tanto para el conjunto de validación como para el de prueba final.
- **Despliegue del modelo:** El modelo entrenado fue exportado al formato TorchScript para su integración en un entorno de inferencia local, compatible con dispositivos sin GPU. Esto permite ejecutar el clasificador en entornos embebidos, de escritorio o de bajo consumo energético, manteniendo una latencia reducida.

VI. MODELOS UTILIZADOS

El modelo seleccionado para esta tarea fue **YOLOv8-cl**s, una variante de la familia YOLOv8 optimizada para clasificación de imágenes. Esta arquitectura combina eficiencia computacional y precisión, lo cual la hace especialmente adecuada para entornos con recursos limitados como CPU locales o dispositivos embebidos.

Se empleó la versión `yolov8s-cl`s (modelo pequeño) como punto de partida, utilizando aprendizaje por transferencia (*transfer learning*) desde un checkpoint preentrenado en ImageNet. Esto permitió acelerar la convergencia y mejorar la generalización sobre un conjunto de datos reducido. La capa final de clasificación fue ajustada para adaptarse a las **11 clases** correspondientes a plantas medicinales y no medicinales.

El entrenamiento fue realizado utilizando la API de Ultralytics en su versión $\geq 8.2.103$, y posteriormente el modelo fue exportado a formato TorchScript para su integración directa con interfaces gráficas y despliegue sin dependencia de GPU.

VII. CONFIGURACIÓN EXPERIMENTAL

- **Lenguaje y versión:** Python 3.10
- **Frameworks y librerías:**
 - PyTorch – para entrenamiento, evaluación y exportación del modelo
 - Ultralytics – para la configuración y ejecución de YOLOv8-cl
 - OpenCV – captura de video y procesamiento de imágenes
 - Tkinter – interfaz gráfica de usuario (GUI) embebida
 - Matplotlib – visualización de resultados e inferencias
- **Entorno de entrenamiento:** Kaggle con GPU x2
- **Entorno de despliegue local:** Laptop con CPU Intel i5, 8GB RAM, sin aceleración por GPU
- **Tamaño del dataset:** 3251 imágenes etiquetadas y balanceadas entre 11 clases, con partición: 87 % entrenamiento, 9 % validación, 4 % prueba.
- **Tamaño de imagen:** 640×640 px (entrada del modelo)

- **Exportación del modelo:** TorchScript (.pt) para inferencia embebida

VIII. RESULTADOS Y DISCUSIÓN

El modelo YOLOv8s-cls entrenado sobre el conjunto de datos curado alcanzó una **precisión superior al 90 %** sobre el conjunto de prueba lo que demuestra una alta capacidad de generalización en la tarea de clasificación multiclase de especies vegetales. Las métricas de evaluación obtenidas fueron:

- **Accuracy (conjunto de prueba):** 92.3 %
- **F1-score promedio:** 0.91
- **Precision promedio:** 0.90
- **Recall promedio:** 0.92

Clases como *Aloe vera* y *Chamaemelum nobile* (Manzanilla) mostraron desempeños particularmente altos, con tasas de precisión cercanas al 100 %, atribuibles a su morfología distintiva y mayor número de ejemplos en el dataset. Por el contrario, especies con menor representación o variabilidad visual elevada presentaron confusiones moderadas, especialmente frente a la clase *PlantasNoMedicinales*.

La aplicación desarrollada en Python con Tkinter y OpenCV permitió realizar inferencia local sobre CPU en tiempo real alcanzando un **tiempo promedio de predicción de 0.18 segundos por imagen**. La visualización en vivo, junto con el despliegue de información curativa contextual, facilitó la validación funcional y la interpretación de resultados por parte del usuario final.

Limitaciones identificadas:

- **Falta de diversidad visual:** Las imágenes de entrenamiento provienen en su mayoría de entornos controlados, reduciendo la capacidad de generalización a escenarios reales.
- **Retroalimentación limitada:** Actualmente no se recopilan predicciones erróneas ni se retroentrena el modelo con nuevos ejemplos, lo que impide una mejora continua.

Estos resultados sugieren que el sistema es prometedor para tareas de clasificación de plantas medicinales en contextos educativos o rurales, siempre que se continúe ampliando y refinando el conjunto de datos con variaciones del mundo real.

IX. CONCLUSIONES Y TRABAJO FUTURO

Este proyecto demuestra la viabilidad de aplicar modelos de aprendizaje profundo, específicamente YOLOv8-cls, para la clasificación automática de plantas medicinales a partir de imágenes. Se desarrolló un sistema funcional con una interfaz interactiva en Python, capaz de realizar inferencias en tiempo real a partir de un modelo exportado a **TorchScript**, optimizado para ejecución local sin necesidad de GPU ni conexión a internet.

El modelo mostró un desempeño sólido en pruebas controladas, superando el 90 % de precisión en especies clave. El uso de TorchScript permitió una integración eficiente en la aplicación de escritorio, facilitando su futura migración a plataformas embebidas.

Como líneas de trabajo futuro se proponen:

- **Ampliación del dataset:** Incorporar más especies, capturas en condiciones naturales (iluminación variable, fondos no controlados) y diversidad geográfica.
- **Aprendizaje incremental:** Incluir un sistema de retroalimentación que permita al usuario corregir predicciones y alimentar futuras etapas de reentrenamiento.
- **Despliegue móvil:** Migrar el modelo TorchScript a plataformas móviles mediante PyTorch Mobile o exportación adicional a TFLite, mejorando portabilidad y escalabilidad.
- **Módulos informativos avanzados:** Agregar funciones como alertas sobre contraindicaciones, trazabilidad geográfica y sugerencias de uso tradicional según especie.

Este trabajo representa un paso inicial hacia el diseño de herramientas accesibles y confiables para la identificación botánica asistida por inteligencia artificial, con impacto potencial en salud preventiva, educación ambiental y conservación del conocimiento ancestral.

REPOSITORIO DEL PROYECTO

El código fuente del sistema, el modelo exportado en formato TorchScript y el script de inferencia local se encuentran disponibles en el repositorio: <https://github.com/jperdomos/GREENMACHINE-PROYECTO/tree/main>

REFERENCIAS

- [1] Smithsonian Insider, "New mobile app that identifies plants by leaf shape launched by Smithsonian and Columbia and Maryland universities," Smithsonian Institution, 3 de mayo de 2011. [En línea]. Disponible en: <https://insider.si.edu/2011/05/new-mobile-app-that-identifies-plants-by-leaf-shape-launched-by-smithsonian-and-columbia-and-maryland-universities/>
- [2] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, y N. Batra, "PlantDoc: A dataset for visual plant disease detection," *arXiv preprint arXiv:1911.10317*, 2019. [En línea]. Disponible en: <https://arxiv.org/abs/1911.10317>
- [3] Pl@ntNet, "Pl@ntNet – The plant identification app," 2023. [En línea]. Disponible en: <https://plantnet.org/en/>
- [4] Ultralytics, "YOLOv8 Docs – Ultralytics," 2023. [En línea]. Disponible en: <https://docs.ultralytics.com>
- [5] Roboflow, "Roboflow: Give your computer vision model the perfect dataset," 2023. [En línea]. Disponible en: <https://roboflow.com>