**Redes neuronales en la solución de problemas de clasificación  
de especies de flora**



**Analitica para la toma de decisiones**

David Díaz Rodríguez

Nicolas Niño Valderrama

Valentina Jiménez Torres

Daniel Perea Mercado

Estudiantes de Ingeniería Industrial

Manuela Londoño Ocampo

Docente

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Industrial

2024

**Descripción de la problemática de negocio**

Los visitantes de parques naturales como el Arví y El Jardín Botánico de Medellín encuentran dificultades para identificar especies de flores comunes como margaritas, dientes de león, rosas, girasoles y tulipanes.

Esta falta de conocimiento sobre la flora local, sumada a una insuficiente atención por parte del personal, limita la experiencia de los visitantes. La alta demanda de estos espacios verdes contrasta con la escasez de guías especializados en botánica, lo que dificulta que los visitantes puedan disfrutar de recorridos informativos y enriquecedores. En consecuencia, la calidad de la experiencia de los visitantes se ve comprometida, ya que no pueden aprovechar al máximo la diversidad de especies presentes en estos entornos naturales.

# 

# Diseño de solución propuesto

Para solucionar la problemática de identificación de flores en parques naturales y jardines botánicos, se propone desarrollar una aplicación móvil que incorpore una red neuronal artificial. Esta herramienta permitirá a los usuarios identificar de manera rápida y precisa especies comunes como margaritas, dientes de león, rosas, girasoles y tulipanes simplemente tomando una fotografía de la flor.

La red neuronal será entrenada con un conjunto extenso de imágenes etiquetadas para reconocer características visuales distintivas de cada especie. Al utilizar técnicas de aprendizaje profundo, se busca optimizar el rendimiento del modelo y minimizar errores en la clasificación. De esta manera, se espera mejorar significativamente la experiencia del visitante al proporcionar información detallada sobre las flores identificadas, reducir la carga de trabajo del personal y optimizar la gestión de los espacios naturales.

# 

# Desarrollo del proyecto

Para el desarrollo de este proyecto se usa un dataset tipo imagen, encontrado e importado desde kaggle el cual nos proporciona datos a cerca de 5 tipos de flores Margarita, Diente de León, Rosa, Girasol y Tulipán.

### **Imagen 1**. Muestra del dataset

### 

### **Análisis exploratorio**

Durante el análisis exploratorio hallamos que el dataset está conformado por 3670 imágenes que perteneces a 5 categorías o especies y están distribuidas así:

* Diente de león (dandelion) con 898 datos
* Tulipanes (tulips) con 799 registros
* Girasoles (sunflowers) con 699 datos
* Rosas (roses) con 641 registros
* Margaritas (daisy) con 633 datos

Al calcular el peso de cada categoría dentro del dataset tenemos que:

### **Imagen 2.** Pesos de las categorías

Debido a esto, consideramos necesario realizar un balance de clases, ya que tenemos como objetivo la métrica de Accuracy y está necesita de que las clases tengan un balance.

## Balance de clases

Para realizar el balance se seleccionan 630 registros de forma aleatoria de cada especie con lo que cada una de las clases tendrá un peso de 20%.

## Separación en entrenamiento, validación y test

Los datos se dividen en entrenamiento (70%), validación (20%) y prueba (10%) para desarrollar una red neuronal robusta y confiable. El conjunto de entrenamiento ajusta los pesos y sesgos del modelo durante el aprendizaje. El conjunto de validación evalúa el rendimiento en datos no vistos durante el entrenamiento, ayudando a ajustar hiperparámetros y prevenir el sobreajuste. El conjunto de prueba se reserva para una evaluación final, proporcionando una medida imparcial del rendimiento del modelo en datos nuevos. Esta separación estructurada asegura resultados representativos y aplicables a casos del mundo real, facilitando la creación de modelos efectivos y evitando interpretaciones sesgadas de su capacidad predictiva.

## 

## Estandarización y data aumentada

Utilizando la clase *ImageDataGenerator* de ***Keras*** normalizamos las imágenes, escalando los valores de los píxeles de su rango original [0, 255] a [0, 1], dividiendo cada valor por 255. Esto facilita el entrenamiento del modelo al mantener los valores en un rango más manejable y homogéneo. Además, las imágenes se redimensionan a un tamaño uniforme de 128x128 píxeles y se convierten a formato RGB (3 canales) si no lo están.

Adicionalmente, para enriquecer el conjunto de datos de entrenamiento y mejorar la generalización del modelo, se aplicaron técnicas de data augmentation:

## 

## Transformaciones aplicadas

* **Rotación:** hasta 30 grados.
* **Desplazamientos horizontales y verticales:** hasta 20% de la imagen.
* **Zoom:** hasta un 20%.
* **Volteo horizontal:** imágenes reflejadas.
* **Corte y relleno:** mediante interpolación para manejar transformaciones fuera de los límites.

Estas técnicas aumentan la diversidad de las imágenes de entrenamiento, ayudando al modelo a aprender características más generales y robustas.

**Visualización de imágenes aumentadas:**Se generaron muestras de imágenes para cada clase, aplicando las transformaciones mencionadas. Los resultados muestran que las imágenes transformadas mantienen sus características esenciales, asegurando que el modelo pueda aprender patrones relevantes.

# 

# Arquitectura: Red Neuronal Convencional (CNN)

La arquitectura definida para este proyecto emplea una red neuronal convolucional (CNN) estructurada de manera secuencial, diseñada específicamente para clasificar imágenes de cinco categorías florales (margaritas, dientes de león, rosas, girasoles y tulipanes). La CNN combina capas convolucionales, de agrupamiento y densas, que permiten extraer características jerárquicas y complejas de las imágenes, optimizando el desempeño del modelo.

En total, la CNN tiene **3,308,933 parámetros**, lo que evidencia su capacidad para aprender patrones complejos. Sin embargo, este número de parámetros puede requerir técnicas avanzadas de regularización para evitar el sobreajuste.

## 

## Optimización de hiper parámetros

Mediante el uso de *HyperModel* y *RandomSearch* de *Keras Tuner* se busca optimizar los hiper parámetros de la arquitectura de la CNN. En este proceso, se define un modelo base donde múltiples parámetros son seleccionados dinámicamente durante la búsqueda, incluyendo el número de filtros, el tamaño de los kernels en las capas convolucionales, las unidades en las capas densas y las funciones de activación. Además, se optimiza la tasa de aprendizaje (learning rate) para el optimizador Adam.

**Resultados de la Optimización**

Al revisar los resultados, notamos que el modelo no presenta *overfitting* ni *underfitting*, y que está logrando una buena generalización de los datos pues el gap entre el accuracy de Train y Val no es muy grande, además consideramos que es un modelo bueno ya que está presentando una métrica de cerca del 80%, aunque se le aplicará regularización con la intención de buscar mejorar dicha métrica de desempeño.

Para las capas convolucionales:

* filters\_1: 64
* filters\_2: 208
* filters\_3: 128
* kernel\_size\_1: 3
* kernel\_size\_2: 5
* kernel\_size\_3: 5

Para las capas densas:

* units\_1: 128
* units\_2: 192
* activation\_d1: tanh
* activation\_d2: tanh

Para el optimizador:

* learning\_rate: 0.0001

### 

### **Imagen 3.** Resultado del modelo optimizado

# Regularización

En esta etapa del proyecto decidimos utilizar tres métodos de regularización:

* Dropout
* L2
* Combinación de Dropout y L2

## 

## Dropout

El método Dropout es una técnica de regularización utilizada para reducir el sobreajuste en redes neuronales profundas. Funciona "apagando" aleatoriamente un porcentaje de conexiones neuronales durante el entrenamiento, evitando que la red dependa demasiado de características específicas y fomentando la generalización.

Se usa Keras Tuner para optimizar los hiper parámetros, encontrando la mejor tasa de Dropout para cada capa dentro de un rango de 0.1 a 0.5, con incrementos de 0.1. Dropout se aplica inmediatamente después de capas convolucionales o densas clave con los siguientes propósitos:

* **Dropout en capas convolucionales**: Después de cada operación de convolución y max pooling, se agrega Dropout para interrumpir conexiones y prevenir que las características extraídas sean demasiado específicas del conjunto de entrenamiento.
* **Dropout en capas densas**: Al final de la red, se aplica Dropout en capas totalmente conectadas para reducir el riesgo de que estas capas memoricen patrones específicos del conjunto de entrenamiento.

Optimizar las tasas de Dropout para cada capa permite al modelo equilibrar la complejidad de la red y su capacidad de generalización. El proceso de búsqueda de hiper parámetros incluye entrenar varios modelos con diferentes combinaciones de tasas de Dropout y seleccionar el conjunto que maximice la precisión de validación.

**Resultados del Dropout**

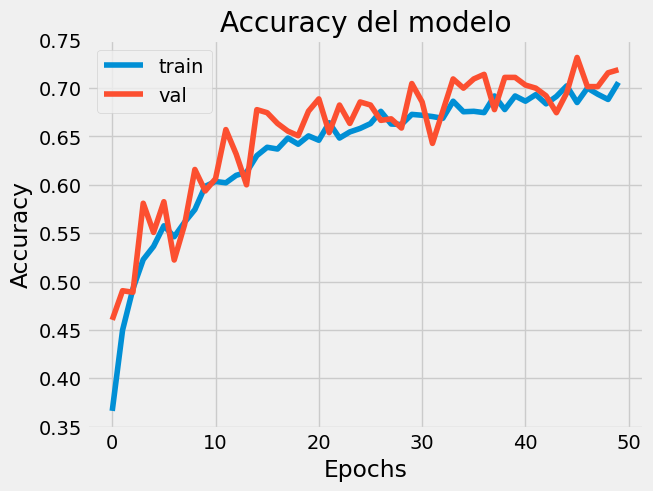
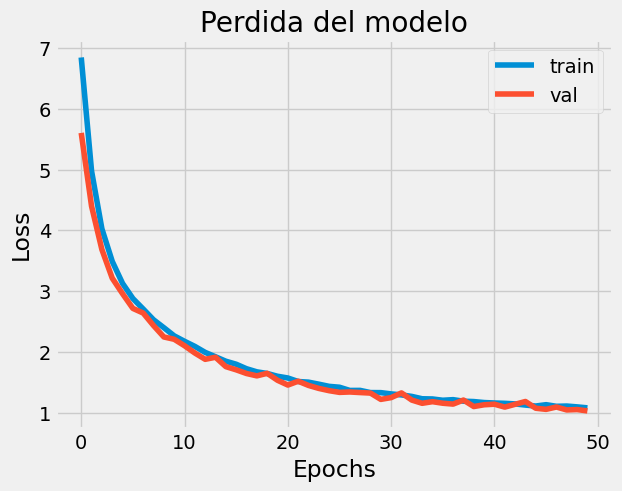
Finalmente, los gráficos de pérdida y precisión permiten evaluar si la inclusión de DropOut logra una mejora en la generalización del modelo, mitigando el sobreajuste observado previamente.

* dropout\_rate\_1: 0.2
* dropout\_rate\_2: 0.1
* dropout\_rate\_3: 0.2
* dropout\_rate\_4: 0.1

### **Imagen 4.** Resultado del modelo optimizado con regularización Dropout

## L2

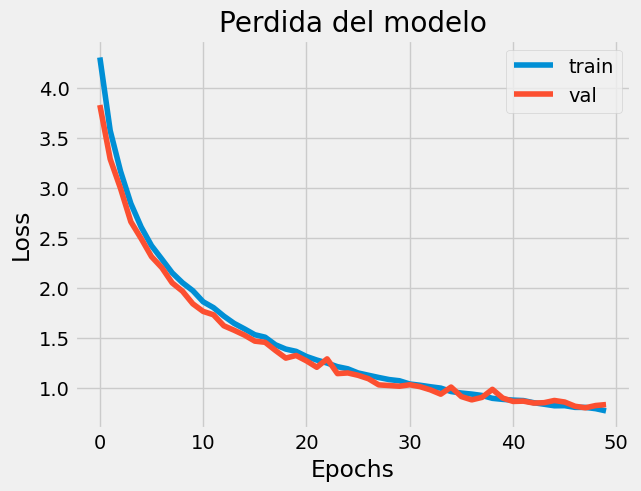
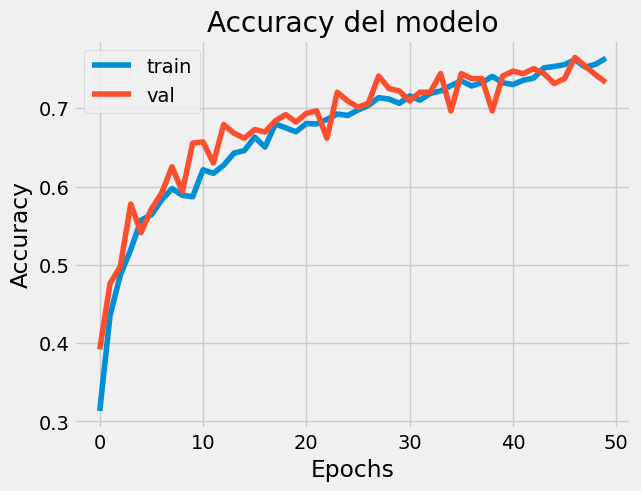
Se aplica regularización L2 en todas las capas convolucionales y densas, lo que agrega un término penalizador a la función de pérdida para reducir el sobreajuste. La regularización L2 minimiza los pesos grandes al incluir su suma cuadrática multiplicada por un factor (λ=0.01), haciendo que el modelo sea más robusto frente al ruido en los datos de entrenamiento.



### **Imagen 5.** Resultado del modelo optimizado con regularización L2

## Dropout y L2

El modelo combina regularización L2 y Dropout para mejorar la generalización y prevenir el sobreajuste. La arquitectura consta de tres bloques convolucionales con activación ReLU, regularización L2 y MaxPooling2D para reducir dimensionalidad, seguido de Dropout que desconecta aleatoriamente una proporción de las neuronas. La salida se aplana con una capa Flatten, conectándose a dos capas densas con regularización L2, activación ReLU o tanh, y Dropout adicional. Finalmente, una capa de salida con activación softmax permite la clasificación en cinco clases. Esta configuración equilibra robustez y rendimiento, optimizando la generalización en datos no vistos.

### **Imagen 6.** Resultado del modelo optimizado con regularización L2 y Dropout

# Elección del modelo

Tras aplicar los métodos de regularización (Dropout, L2 y su combinación), se seleccionó el modelo con los mejores resultados en precisión y generalización. Este modelo no solo alcanzó un desempeño óptimo en la clasificación de las cinco especies de flores (margaritas, dientes de león, rosas, girasoles y tulipanes), sino que también mejorará la experiencia de los visitantes en parques naturales y jardines botánicos.

La CNN base mostró alta precisión en el conjunto de entrenamiento, pero un rendimiento inferior en el conjunto de validación, sugiriendo sobreajuste. Con la optimización de hiper parámetros, se mejoró la precisión en ambos conjuntos. Ajustar los hiper parámetros ayudó a equilibrar el sesgo y la varianza, mejorando la robustez del modelo y su capacidad de generalización. Finalmente, la combinación de Dropout y L2 resultó en la mejor precisión de validación y una disminución constante de la pérdida de validación. Esta combinación aprovechó las fortalezas de ambas técnicas, creando un modelo robusto y simple con la mejor capacidad de generalización.

**Resultados de la evaluación del modelo en el conjunto de testeo:**

Los resultados de la evaluación en el conjunto de prueba muestran que el modelo alcanzó una precisión (Test Accuracy) del 73.10% y una pérdida (Test Loss) de 0.8451.

La precisión en el conjunto de prueba es consistente con la observada durante el entrenamiento y validación, lo que indica que el modelo generaliza razonablemente bien y no presenta un sobreajuste evidente. Aunque el modelo ofrece un desempeño aceptable, podrían aplicarse estrategias adicionales para mejorar la precisión, como:

* Aumentar el número de épocas para permitir un entrenamiento más prolongado.
* Incorporar más datos para mejorar la capacidad del modelo de captar la diversidad en las imágenes.

# Conclusiones

* Al observar los gráficos, se identifica que tanto la pérdida como la precisión (accuracy) del modelo continúan en una tendencia de mejora al finalizar las 50 épocas de entrenamiento. Esto sugiere que el modelo no ha convergido completamente y podría beneficiarse de un entrenamiento adicional. Incrementar el número de épocas permitiría al modelo explorar más el espacio de parámetros, lo cual es especialmente beneficioso en conjuntos de datos con características complejas como este. Aunque los resultados actuales son sólidos, el análisis del comportamiento del modelo indica que hay margen de mejora con un entrenamiento adicional.
* El desarrollo de una red neuronal convolucional para la identificación de flores comunes representa una solución innovadora y efectiva a las limitaciones que enfrentan los visitantes de parques naturales y jardines botánicos. La combinación de técnicas avanzadas, como la regularización mediante Dropout y L2, junto con un proceso riguroso de optimización de hiper parámetros, ha permitido construir un modelo con robustez. Esto demuestra la capacidad del modelo para generalizar y reconocer patrones en imágenes de distintas especies florales, cumpliendo con los objetivos planteados.
* Al proporcionar una experiencia más educativa y enriquecedora para los visitantes, esta herramienta no solo mejorará la apreciación de la biodiversidad, sino que también optimiza la gestión de recursos humanos en estos espacios naturales. La implementación de una red neuronal eficiente y precisa permite a los usuarios identificar rápidamente las especies florales simplemente tomando una fotografía, lo que enriquece su experiencia educativa y reduce la necesidad de personal especializado en cada visita.