

Proyecto 1:

Detección de frailejones

🎯 *Red neuronal para detectar automáticamente frailejones sobre imagen aerea páramo de Chingaza y Cruz Verde.*



Luis Alejandro Garzón Ramirez
Zorayda Acevedo Jimenez
Yulieth Danitza Aguillón Ortega



Contenido

1. Entendimiento de la situación
2. Objetivo de negocio y objetivos minería de datos
3. Entendimiento de los datos
4. Preparación de los datos
5. Modelo
6. Evaluación



1. Entendimiento de la situación



2. Objetivo de negocio

1 Estimar la Probabilidad de que una Planta sea un Frailejón

Desarrollar un modelo de detección que permita predecir con precisión la probabilidad de que una planta observada sea un frailejón de la especie *Espeletia* optimizando la identificación y clasificación de esta especie.

2 Calcular la Densidad de Frailejones en un Área Determinada

Analizar y cuantificar la distribución de frailejones en distintas regiones para evaluar su densidad poblacional, facilitando estudios de conservación y monitoreo ambiental.

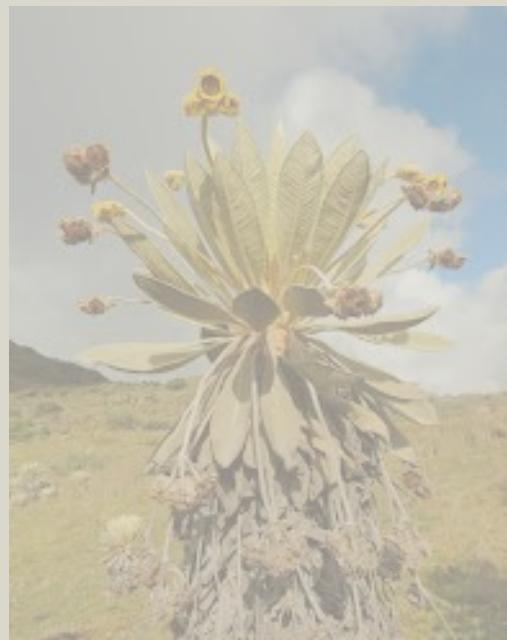


2. Objetivos de minería de datos

Objetivo general: construir una red neuronal para detectar automáticamente plantas nativas (frailejones) sobre imágenes aérea del páramo de Chingaza y Cruz Verde.

Objetivos específicos:

- 1 Diseñar cinco (5) propuestas de modelos usando redes con diferentes arquitecturas y/o hiperparámetros.
- 2 Obtener un desempeño en los modelos medido por AUC de la siguiente forma:
 - Red Sencilla: > 0.88
 - Multi-capas: > 0.95
 - Convolucional: > 0.95
 - Convolucional + VGG16 > 0.99
 - Convolucional con transferencia VGG-16 y ajuste de hiperparámetros
- 3 Verificar de forma visual (cualitativa) la detección de frailejones por medio del mejor modelo



3. Entendimiento de los datos

Bordes detectados en imagen de Frailejon y No frailejon

Contornos Frailejón 1 Contornos Frailejón 2 Contornos Frailejón 3 Contornos Frailejón 4



Contornos No Frailejón 1 Contornos No Frailejón 2 Contornos No Frailejón 3 Contornos No Frailejón 4

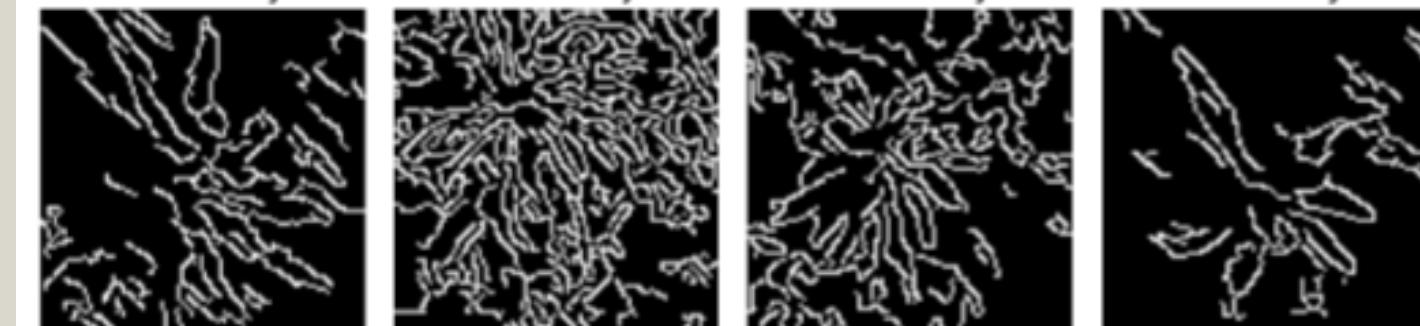


Frailejon y No frailejon

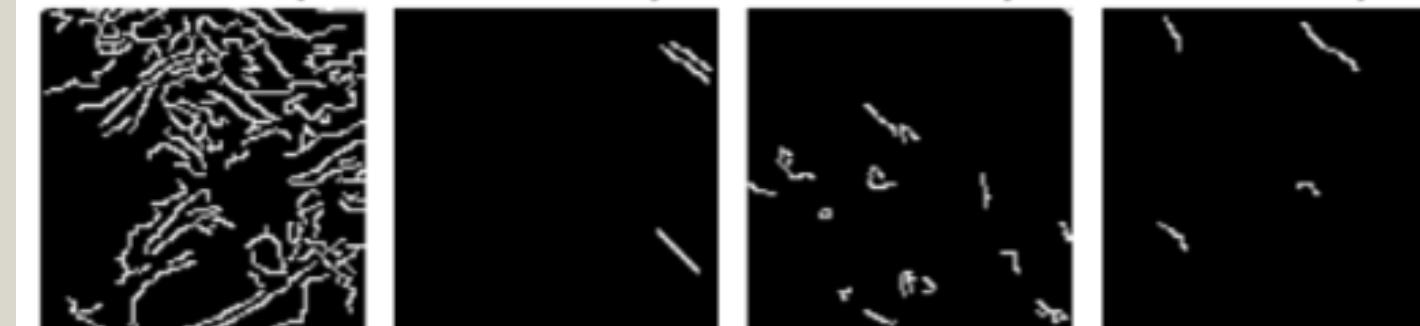


Contornos detectados en imagen de Frailejon y No frailejon

Bordes Frailejón 1 Bordes Frailejón 2 Bordes Frailejón 3 Bordes Frailejón 4



Bordes No Frailejón 1 Bordes No Frailejón 2 Bordes No Frailejón 3 Bordes No Frailejón 4



4. Preparación de los datos

- **División de la imagen**

Dividir el set en 250 imágenes de 70*70 mediante la función 'Import_Imagenes()'

- **Separación entre datos de entrenamiento y prueba**

Conjunto de entrenamiento consta de 175 imágenes, mientras que el de prueba incluye 75 imágenes.

- **Normalización de los valores RGB**

Formato RGB, con valores que varían entre 0 y 255 para cada canal. Para normalizar estos valores, se dividen entre 255, lo que reescalará los datos al rango de 0 a 1, representados como valores decimales.

- **Reescalamiento de las imágenes**

Necesario ajustar el tamaño de 70 x 70 píxeles, a 224 x 224 píxeles para su utilización en modelos preentrenados como VGG16 o MobileNetV2.

- **Aumento de datos (Data Augmentation)**

Incluye rotaciones en un rango de 20 grados, desplazamientos verticales y horizontales hasta un 20%, cambio el brillo de la imagen en un 20% e invierte la imagen con efecto espejo.



5. Modelos

Modelo 1 - Red neuronal sencilla

Se tiene una capa oculta con 64 neuronas con función de activación ReLU. La capa de salida tiene 1 neurona con activación sigmoide.

Modelo 2 - Multi-capas:

```
{'batch_size': 64, 'epochs': 50, 'learning_rate': 0.001, 'neuronas1': 64, 'neuronas2': 64, 'neuronas3': 16}
```

Modelo 3 - Convolucional:

3 Capas convolucionales (Conv2D) + seguidas de capas maxpooling2Dn (32, 64 y 128 filtros respectivamente) con kernel de 3x3 con activación ReLU y Dropout de 0,5 y early stopping para prevenir el sobreajuste, posteriormente una capa de aplanamiento (Flatten) una capa de salida con activación sigmoide para la clasificación binaria.

Modelo 4 - Convolucional + VGG16:

La base VGG16, una capa de aplanamiento (Flatten) para convertir las salidas en un vector, una capa densa (Dense) con 128 neuronas y activación ReLU, una capa de Dropout 0.5 para prevenir el sobreajuste y una capa de salida con activación sigmoide.

Modelo 5 - Convolucional con transferencia VGG-16 y ajuste de hiperparámetros:

libreria Optuna los mejores hiperparámetros encontrados son: {'neuronas': 384, 'dropout': 0.4, 'optimizer': 'adam', 'learning_rate': 0.004183979572176086}.

Modelo 6 - MobileNetV2

Dos capas de activacion de 512, 256 y dropout de 0,3 y 0,4 y salida de activacion sigmoid.

6. Evaluación

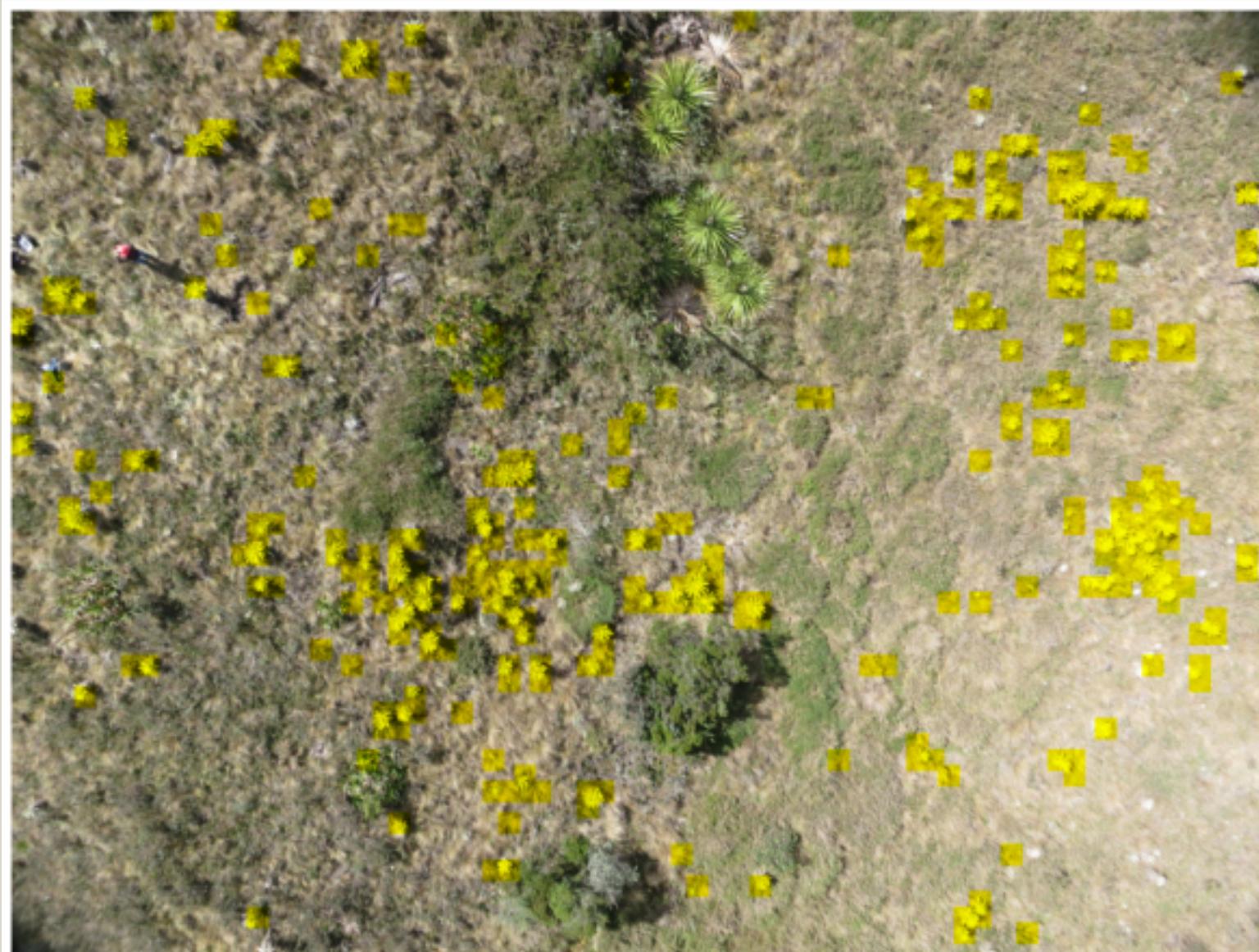
- Desempeños de los modelos:

Nombre del modelo	AUC en entrenamiento	AUC en validación	AUC objetivo
Modelo 1 - Red neuronal sencilla	0.98	0.88	0.88
Modelo 2 - Multi-capas	0.81	0.89	0.95
Modelo 3 - Convolucional	0.96	0.96	0.95
Modelo 4 - Convolucional + VGG16	0.99	0.99	0.99
Modelo 5 - Convolucional con transferencia VGG-16 y ajuste de hiperparámetros	1.00	1.00	No aplica
Modelo 6 - MobileNetV2	0.98	0.87	No aplica



6. Evaluación

- Cualitativa:



Densidad en Grupo 1:

$$118 / 4661 = 0.025316 \text{ cuadros con frailejones/cuadro total}$$

Densidad en Grupo 2:

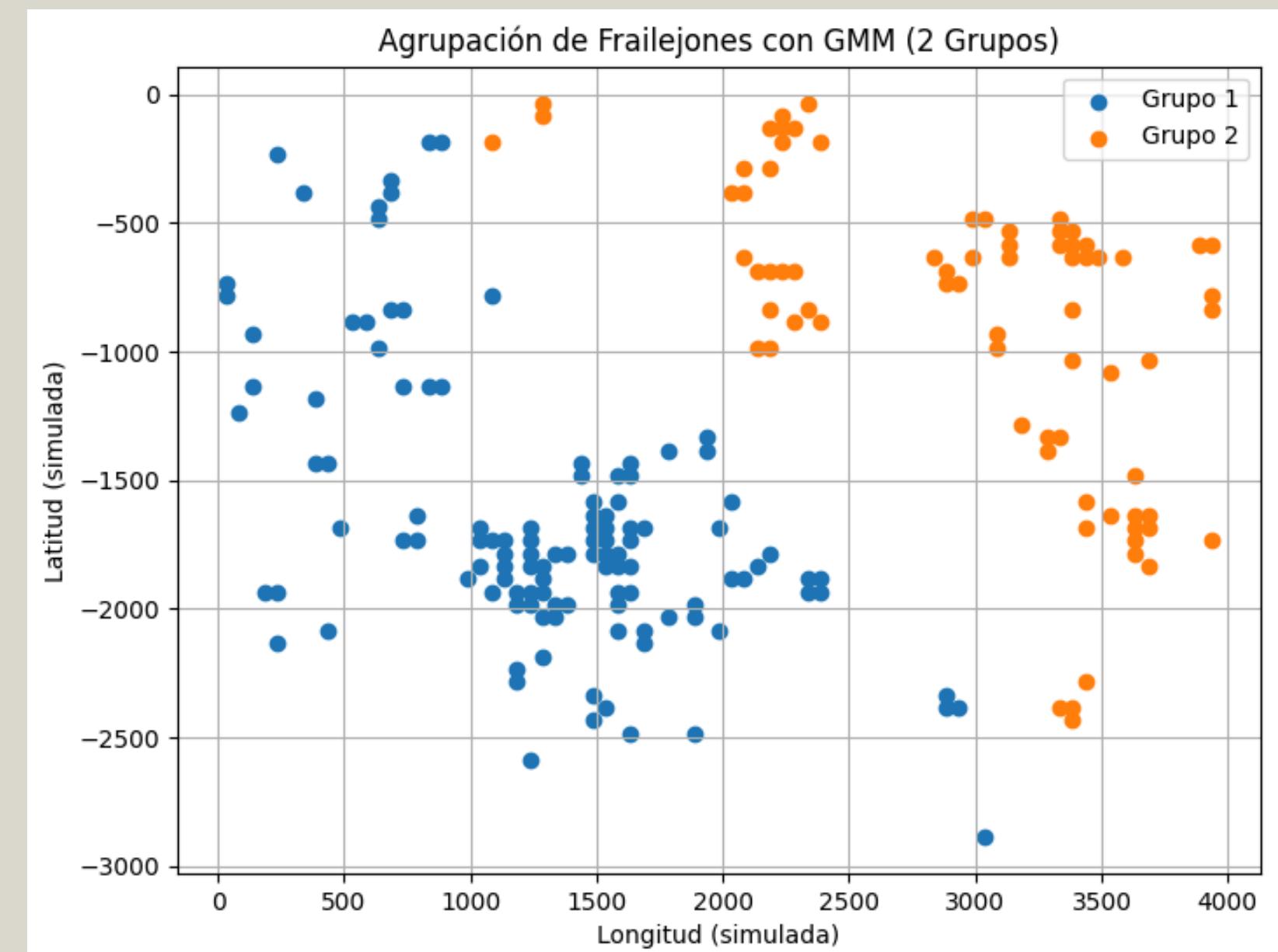
$$75 / 4661 = 0.016091 \text{ cuadros con frailejones/cuadro total}$$

Densidad Total:

$$193 / 4661 = 0.041407 \text{ cuadros con frailejones/cuadro total}$$

Identificación cualitativa del mejor modelo resultante:

Modelo 5 - Convolucional con transferencia VGG-16 y ajuste de hiperparámetros



Gracias.

