Detección automática de frailejones de imagen aérea tomada del páramo de chingaza y cruz verde

Daniela Parra[a,c], Manuel Hernández [a,c], Paula Jaimes [a,c],
Lina Varon[a,c] , Jenny Ortiz [a,c]
Sergio Alberto Mora[b,c]

[a]*Estudiantes de Maestría en Analítica para la Inteligencia de Negocios*
[b]*Profesor, Departamento de Ingeniería Industrial*
[c]*Pontificia Universidad Javeriana, Bogotá, Colombia*

ENTENDIMIENTO DEL NEGOCIO

Background

Los frailejones desempeñan un papel fundamental en el equilibrio ecológico debido a su capacidad para regular el ciclo hídrico y mitigar los efectos del cambio climático. Su capacidad para retener agua en las hojas y actuar como sumideros de carbono ayuda a mantener el equilibrio ambiental y reducir las emisiones de gases de efecto invernadero (CAR, 2024). Por esta razón, las preocupaciones por el incremento de incendios en las zonas de páramo -como consecuencia del cambio climático y factores humanos como las fogatas e incendios provocados- han aumentado significativamente. Desde 2020, estos incendios se han propagado considerablemente; según la Unidad Nacional para la Gestión del Riesgo de Desastres, solo en ese año se registraron 179 incendios que consumieron 2.089 hectáreas de páramo, representando la mayor pérdida de cobertura vegetal en los últimos 16 años. Más recientemente, en 2024, el incendio en el Páramo de Berlín afectó 317 hectáreas, destruyendo áreas con especies nativas como los frailejones; esta situación despertó una gran preocupación ante las autoridades ambientales que lo catalogaron como desastre ambiental irreparable pues el frailejón apenas crece un centímetro al año, lo que implica que se recuperación tardará muchos años.

Bajo este panorama, contar con una herramienta analítica que permita detectar la densidad y/o distribución de frailejones en una zona específica, puede ser muy útil para evaluar el estado de un páramo y monitorear su conservación y preservación. Dicho esto, el presente proyecto tiene como objetivo construir una herramienta analítica que permita detectar la presencia de frailejones sobre imágenes aéreas del páramo de Chingaza y Cruz Verde, los detalles se resumen en el siguiente apartado.

Business goal

- Determinar la cantidad y densidad de frailejones en los páramos de Chingaza y Cruz Verde para apoyar la preservación y la reconstrucción del ecosistema en áreas con baja densidad de esta especie.

Data mining goal

- Desarrollar una red neuronal sencilla que permita detectar frailejones en imágenes aéreas de los páramos de Chingaza y Cruz Verde.
- Desarrollar una red neuronal multicapa que permita detectar frailejones en imágenes aéreas de los páramos de Chingaza y Cruz Verde.
- Desarrollar una red neuronal convolucional que permita detectar frailejones en imágenes aéreas de los páramos de Chingaza y Cruz Verde.
- Desarrollar una red neuronal Convolucional + VGG16 que permita detectar frailejones en imágenes aéreas de los páramos de Chingaza y Cruz Verde.

Data mining success criteria

- KPI 1: La red neuronal sencilla debe superar un AUC (Área bajo la curva) de 0,88.
- KPI 2: La red neuronal multicapa debe superar un AUC (Área bajo la curva) de 0,95.
- KPI 3: La red neuronal convolucional debe superar un AUC (Área bajo la curva) de 0,95.
- KPI 4: La red neuronal Convolucional + VGG16 debe superar un AUC (Área bajo la curva) de 0,99.

ENTENDIMIENTO DE LOS DATOS

Para la realización del proyecto, fue entregada una imagen aérea tomada por un dron sobre los páramos de Chingaza y Cruz verde. Para el entendimiento de los datos, se revisó la imagen proporcionada en la que se debe identificar la ubicación y densidad poblacional del grupo de frailejones.

PREPARACIÓN DE LOS DATOS

Fue proporcionado previamente el módulo ImportImagenes.py que divide la data en entrenamiento y validación con una proporción de 70-30 tal y como se expone a continuación:

- CE_x: Corresponde a una matriz que almacena los 14700 pixeles de cada una de las 175 imágenes de la data de entrenamiento.
- CV_x: Corresponde a una matriz que almacena los 14700 pixeles de cada una de las 75 imágenes de la data de validación.
- CE_y: Corresponde a una matriz con 175 etiquetas que identifican si cada imagen de entrenamiento corresponde o no a un Frailejón.
- CV_y: Corresponde a una matriz con 75 etiquetas que identifica si cada imagen de validación corresponde o no a un Frailejón.

Para la preparación de los datos en los modelos de redes convolucionales estas matrices se convirtieron a tensores así: la data de entrenamiento consta de 175 imágenes RGB de 70x70 pixeles, por lo que se construye un tensor final con las dimensiones (175,70,70,3). La data de validación consta de 75 imágenes RGB de 70x70 pixeles y se construye un tensor de dimensiones (75,70,70,3).

Adicional a esta preparación, para las redes preentrenadas RESNET50 y VGG-16 se realiza una preparación adicional que consiste en redimensionar las imágenes a un tamaño de 224x224 pixeles ya que este es el tamaño con el cual estas redes fueron entrenadas.

MODELADAMIENTO

Red Sencilla

Una Red Neuronal Sencilla está compuesta por una capa de entrada, una única capa oculta y una capa de salida. A continuación, se describe la red generada para la detección de Frailejones:

- Capa de entrada: Consta de 14700 neuronas asociadas a los pixeles de cada imagen.
- Capa oculta: Realiza cálculos y transforma las entradas por medio de una función de activación. La cantidad de neuronas de esta capa y la función de activación fueron definidos con ayuda de la librería optuna.
- Capa de salida: Produce las predicciones finales, usando una función de activación sigmoide por tratarse de un problema de clasificación binaria.

Además de la función de activación de la capa oculta y su número de neuronas, en la sintonización de parámetros realizada con optuna se definieron los hiperparámetros: optimizer, l2_reg y lr. Ver Figura 1.

Figura 1. Resultado de sintonización de hiperparámetros con optuna.

```
Best trial:
  Value: 0.8989825248718262
  Params:
    units: 567
    activation: tanh
    l2_reg: 0.0004375082465372553
    optimizer: adam
    lr: 0.00012178586759012545
```

Fuente: Elaboración propia.

El optimizador se refiere al algoritmo utilizado para ajustar los pesos del modelo durante el entrenamiento. Cada optimizador tiene sus propias estrategias para actualizar los pesos y puede afectar significativamente el rendimiento del modelo.

El término l2_reg controla la intensidad de la penalización al valor de los pesos del modelo en la función de pérdida. Un valor más alto de l2_reg implica una mayor penalización, lo que tiende a reducir los valores de los pesos y, por ende, a simplificar el modelo, mientras que un valor más bajo permite que el modelo ajuste más libremente. Por lo tanto, este l2_reg más alto ayuda a mejorar la capacidad de generalización del modelo y evitar sobreajuste.

El término lr se refiere a la tasa de aprendizaje que controla el tamaño de los ajustes que se hacen a los pesos del modelo durante el entrenamiento y determina la velocidad con la que el modelo se adapta a los datos.

El modelo fue configurado utilizando estos parámetros, logrando un AUC de 0.9590 en el conjunto de entrenamiento y 0.9268 en el conjunto de validación.

Red multicapa

Ahora bien, con respecto a la red multicapa siendo esta una estructura con una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa está formada por nodos o neuronas conectados a través de pesos que se ajustan durante el entrenamiento para minimizar el error en la predicción. Este tipo de red es capaz de aprender y modelar relaciones complejas entre entradas y salidas mediante el uso de funciones de activación no lineales.

En la construcción del modelo de red neuronal sencilla, se emplea Optuna para la sintonización de hiperparámetros con el objetivo de identificar la configuración óptima de los parámetros del modelo, permitiendo mejorar su rendimiento y eficacia. Para la construcción de la red neuronal multicapa, utilizamos el mismo sintonizador para ajustar dinámicamente el número de capas ocultas, la cantidad de neuronas por capa, la función de activación y el valor de regularización l2.

La red neuronal optimizada con Optuna logró un valor de evaluación de 0.9117, lo que indica un buen rendimiento en el objetivo de clasificación binaria. La red está compuesta por dos capas ocultas con 104 y 107 neuronas, utilizando funciones de activación tanh y relu. La regularización l2, aunque pequeña, se aplicó a ambas capas, lo que contribuye a prevenir el sobreajuste y mejorar la capacidad de generalización del modelo. El optimizador RMSprop junto con una tasa de aprendizaje baja muestra una dirección minuciosa para la actualización de pesos, permitiendo que el modelo se adapte de manera controlada.

Figura 2. Resultado de sintonización de hiperparámetros con optuna

```
Best trial:
  Value: 0.9117006063461304
  Params:
    num_layers: 2
    units_0: 104
    activation_0: tanh
    l2_reg_0: 1.007705249012496e-06
    units_1: 107
    activation_1: relu
    l2_reg_1: 1.0384879612712697e-06
    optimizer: rmsprop
    lr: 1.0666331576509997e-05
```

Fuente: Elaboración propia.

Con la ejecución de esta red neuronal sintonizada, implementamos los mejores parámetros encontrados con Optuna y evaluamos el rendimiento utilizando la métrica AUC, tanto en el conjunto de entrenamiento como en el de validación.

El modelo muestra un rendimiento de AUC del 96.51% en el conjunto de entrenamiento y un 91.06% en el conjunto de validación, lo que indica una alta capacidad del modelo para identificar un Frailejón. Aunque hay una reducción en el AUC al pasar de entrenamiento a validación, el modelo sigue siendo robusto, manteniendo un buen equilibrio entre la precisión y la generalización.

Red Convolucional

La red convolucional o CNN, es una red de aprendizaje profundo diseñada para procesar imágenes en formato de grilla. Su funcionamiento se da a partir de la aplicación de filtros sobre imágenes para extraer características como bordes, formas y colores, una vez identificadas, se combinan en capas realizar predicciones de imagen, ya sea para clasificación o detección. En nuestro caso, se realizó un cambio en la estructura de entrada de datos para cumplir con los parámetros de entrada de CNN, donde la red recibe en entrenamiento y validación el número de imágenes: 175, alto de la imagen: 70, ancho de la imagen: 70 y formato de la imagen a color: 3 y, por otro lado, CE_y y CV_y se convirtieron en arreglos unidimensionales.

Figura 3. Estructura de imágenes de entradas redimensionadas

```
CE_x shape: (175, 70, 70, 3)
CV_x shape: (75, 70, 70, 3)
CE_y shape: (175,)
CV_y shape: (75,)
```

Fuente: Elaboración propia

Teniendo el formato adecuado para diseñar la red, se probaron dos escenarios, en primer lugar, se utilizó optuna para la sintonización de hiperparámetros y, en segundo lugar, se diseñó una red convolusional sin ningún tipo de sintonización.

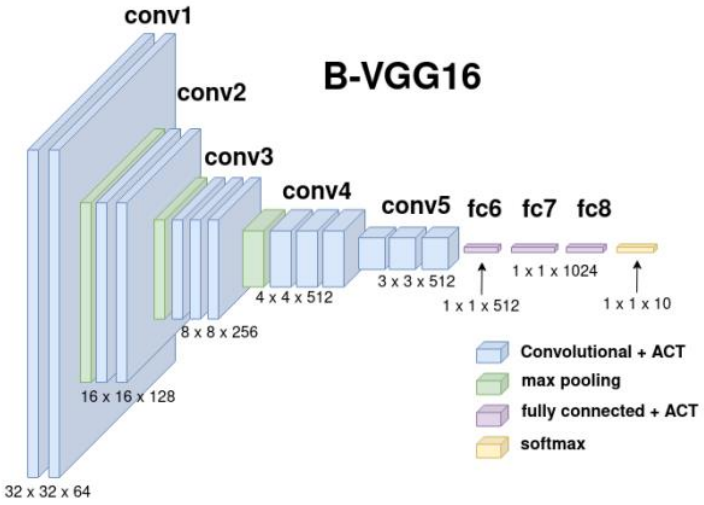
Para el primer escenario, se sintonizaron los siguientes hiperparámetros:

- Kernel_size: Dimensiones del filtro que se desliza sobre la imagen en las capas convolucionales, tiene la forma de (n,n).
- Stride: Define cuanto se mueve el kernel a través de la imagen en cada paso.
- Activation: La función de activación se encarga de que la red aprenda patrones complejos
- L2_reg:

Red VGG-16

La red VGG-16 (Visual Geometry Group 16) es una red neuronal convolucional compuesta por 16 capas, organizadas en cinco bloques de capas convolucionales, cada uno seguido de una capa de Max Pooling para reducir la dimensionalidad de los mapas de características; particularmente, se utiliza el Max Pooling 2x2 con un stride de 2. Esto significa que la operación de pooling se realiza en ventanas de tamaño 2x2 y se mueve dos píxeles a la vez sobre el mapa de características, extrayendo el valor máximo en cada ventana con lo cual se retienen las características más importantes que las capas convolucionales han detectado. Por último, la red culmina con tres capas completamente conectadas, donde la última capa aplica una función de activación softmax o sigmoide dependiendo de si se trata de un problema de clasificación multiclase o un problema de clasificación binaria; para el problema del presente proyecto, la función de activación a utilizar sería la sigmoidal. Dicha estructura se puede evidenciar a continuación.

Figura 3. Arquitectura de la red VGG-16



Fuente: (Lacomi et al., 2022)

En este contexto, la implementación de la red neuronal VGG16 implica, en términos generales, cargar el modelo preentrenado y ajustar la forma de entrada esperada por el mismo. Específicamente, esto implica redimensionar las imágenes de entrada a un tamaño de 224x224 píxeles, ya que VGG16 está diseñada para procesar imágenes de esta dimensión. Si las imágenes no tienen este tamaño, el modelo no podrá procesarlas adecuadamente. Tras aplicar este redimensionamiento y tras realizar la partición de test y train, con un 70% para entrenamiento, se obtiene la siguiente estructura.

Figura 4. Estructura de imágenes de entradas redimensionadas

```
Forma de CE_x_resized: (175, 224, 224, 3)
Forma de CV_x_resized: (75, 224, 224, 3)
Forma de CE_y: (175, 1)
Forma de CV_y: (75, 1)
```

Fuente: Elaboración propia

Como se puede evidenciar, CE_x_resized, que corresponde a la data de entrenamiento tiene una forma de (175, 224, 224, 3), lo que significa que contiene 175 imágenes de tamaño 224x224 píxeles con 3 canales de color (RGB); CV_x_resized tiene una forma de (75, 224, 224, 3), indicando 75 imágenes del mismo tamaño y formato; CE_y tiene una forma de (175, 1), lo que corresponde a 175 etiquetas, cada una asociada a una imagen en CE_x_resized; y CV_y tiene una forma de (75, 1), correspondiente a 75 etiquetas asociadas a las imágenes en CV_x_resized.
Una vez realizado este procesamiento de las imágenes de entrada, se procede a hacer uso de la herramienta de optimización de hiperparámetros Optuna, con el objetivo de mejorar el rendimiento de la red. Dicho esto, tras aplicar la optimización, los mejores hiperparámetros encontrados fueron:

- El número de unidades en la capa densa: 502
- La función de activación: Relu
- Valor de la regularización L2: 0.0000450
- Optimizador utilizado para entrenar el modelo: adam
- Tasa de aprendizaje del optimizador: 0.003
- Número de épocas: 9

Como se puede observar, los mejores hiperparámetros encontrados por Optuna son: 502 neuronas en la capa densa, ReLU como función de activación, la cual ayuda a mitigar el problema del desvanecimiento del gradiente durante el entrenamiento ayudando a que las actualizaciones de pesos sean más significativas y eficientes; una regularización L2 de 0.0000450 lo que significa que la penalización aplicada a los pesos será pequeña, en términos prácticos esto implica que el modelo tiene mayor capacidad para aprender representaciones complejas de los datos de entrenamiento (Algo que aumenta el riesgo de overfitting, pero se evaluará más adelante con los resultados del modelo en validación). Un optimizador Adam, el cual permite ajustar automáticamente la tasa de aprendizaje de cada parámetro del modelo durante el entrenamiento; 9 épocas de entrenamiento, y una tasa de aprendizaje de 0.003, lo que implica actualizaciones de pesos pequeñas, permitiendo una convergencia estable, aunque más lenta.

Con los hiperparámetros óptimos determinados, se procedió a cargar la red preentrenada VGG-16 sin la parte superior del modelo base para adaptarla a nuestra tarea específica. El modelo fue configurado utilizando estos parámetros, logrando un AUC de 0,969 en el conjunto de entrenamiento y 0,945 en el conjunto de validación.

Resnet50

Se utiliza el aprendizaje por transferencia al usar un modelo de Red Neuronal Convolucional Residual de 50 capas (ResNet50) que ha sido entrenado en ImageNet. Teniendo en cuenta que la Resnet50 calcula 1000 clases y el nuestro es un problema de clasificación binaria o de una clase, se modifican las capas de salida agregando un GlobalAveragePooling2D para reducir la dimensionalidad de las características extraídas por ResNet50 y se genera una capa de salida con una neurona y una función de activación sigmoide, para devolver una probabilidad de clase entre 0 y 1.

Además, para mejorar el rendimiento del modelo sobre los datos de entrenamiento propios, se descongelan algunas capas superiores para que se ajusten a las características de nuestra tarea específica. Una vez configuradas las capas personalizadas, se entrena el modelo con nuestro conjunto de datos. Se utilizó optuna para la sintonización de hiperparámetros, sin embargo, el mejor modelo fue obtenido con una configuración manual en la que se ajustaron los pesos a partir de la capa 56 con un entrenamiento en 20 épocas. El modelo fue configurado utilizando estos parámetros, logrando un AUC de 0.9551 en el conjunto de entrenamiento y 0.9277 en el conjunto de validación.

EVALUACIÓN

Desempeño de los diferentes modelos y argumentación de cuál es el mejor para poner en producción

En términos generales y teniendo en cuenta el desempeño de los diferentes modelos expuestos en la Tabla 1, se identifica una mejora muy leve en el AUC (Área bajo la curva ROC) tanto en entrenamiento como en validación cuando se usan redes convolucionales. Esto es contrario a lo esperado y puede estar relacionado al esfuerzo computacional que se requiere para sintonizar los hiperparámetros cuando las redes se vuelven muy complejas. Es decir, en redes densamente conectadas se realizaron 100 iteraciones en optuna, mientras que en las redes convolucionales la sintonización se hizo en un número mucho menor de iteraciones (máximo 30) dada la limitación computacional de las máquinas en las que fueron ejecutados.

Específicamente el mejor desempeño lo obtuvo el modelo que usó una red preentrenada VGG16 con la cual se obtuvo el mejor AUC, tanto en el conjunto de entrenamiento como en el de validación. Con un AUC de 0.9695 en entrenamiento y 0.9456 en validación, la VGG16 logró un equilibrio entre su capacidad de aprendizaje y su generalización a nuevos datos, con lo cual se puede afirmar que no hay presencia de sobreajuste.

Tabla 1. Resumen de métricas obtenidas en train y test

Modelo	AUC entrenamiento	AUC validación
Red Sencilla	0.9590	0.9268
Red Multicapa	0.9695	0.9300
Red Convolucional		
Red Convolucional VGG16	0.9695	0.9456
Red Convolucional Resnet50	0.9551	0.9277

Fuente: Elaboración propia

Justificación del aprendizaje por transferencia y relación entre la tarea base y la segunda tarea objetivo.

El aprendizaje por transferencia en redes permite que se aproveche el conocimiento de redes que han sido entrenadas en grandes conjuntos de datos, lo cual les permite identificar características generales de las imágenes, tales como bordes, texturas, y formas. Estas redes ya han ajustado sus hiperparámetros mediante el aprendizaje de dichos conjuntos de datos, lo que les permite ofrecer un rendimiento robusto desde el inicio. Al utilizarlas, solo es necesario ajustar las últimas capas para la nueva tarea específica, lo que reduce significativamente el tiempo de entrenamiento y mejora la eficiencia del modelo al adaptarse a nuevas tareas con menos datos y a un menor costo computacional.

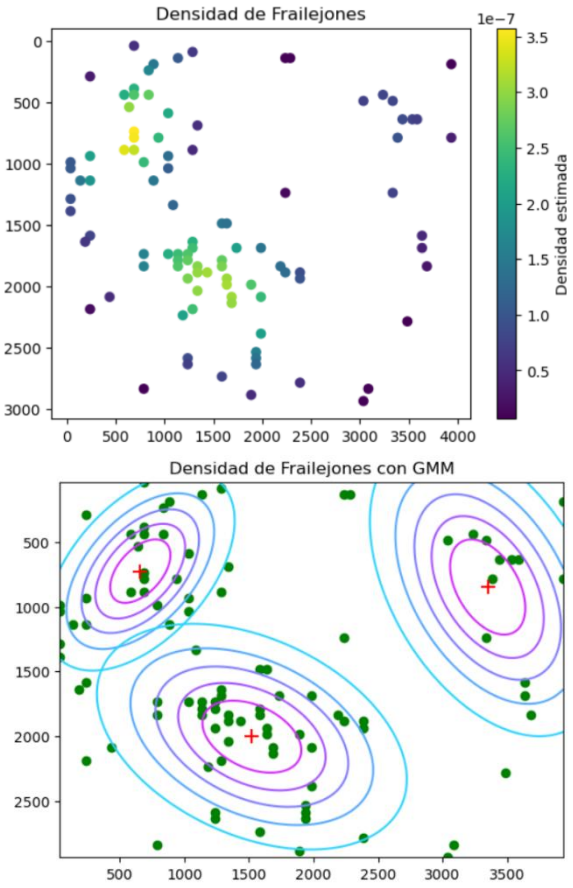
Los modelos preentrenados utilizados en este proyecto: ResNet50 y VGG-16, fueron entrenados con el conjunto de datos de ImageNet, que contiene 1.000 categorías diferentes. Estos modelos tienen la capacidad de clasificar imágenes en múltiples clases a través de su tarea base. Dicho esto, en relación con la tarea objetivo de este proyecto, que es identificar frailejones en imágenes aéreas, se puede determinar que ambas tareas comparten un enfoque de clasificación de imágenes que implica reconocer patrones específicos. En este punto, ajustando únicamente las últimas capas de estas redes, es posible adaptar los modelos para cumplir con la tarea de clasificar correctamente si una imagen contiene o no un frailejón. Por último, la literatura existente sugiere que la transferencia de aprendizaje es más efectiva cuando ambas tareas son similares, como es el caso en el presente proyecto, donde ambas son tareas de clasificación de imágenes.

CALCULO DE DENSIDAD POBLACIONAL

Considerando el análisis realizado en el apartado anterior, se observa que, en términos de AUC, la red convolucional VGG-16 es la mejor opción. No obstante, es importante mencionar que el AUC mide la capacidad del modelo para distinguir entre las clases en términos generales, pero no necesariamente indica cuán bien el modelo localiza o segmenta los objetos en una imagen. Bajo este contexto, tras la prueba cualitativa realizada con el modelo VGG-16 se evidenciaron un alto número de falsos positivos. Esto confirma que, aunque un modelo pueda presentar un alto AUC, aún puede enfrentar dificultades en la localización o segmentación precisa, lo cual se refleja en un bajo Accuracy.

Por esta razón, para el cálculo de la densidad poblacional, se tomó como referencia el resultado de la red multicapa, el cual con un AUC de 0.93 en validación y 0.96 en entrenamiento, refleja un buen comportamiento en el análisis cualitativo, tal y como se puede evidenciar en el anexo 1.

Figura 5. Cálculo de densidad poblacional



Clúster	Densidad
0	1.7574895889036337e-07
1	1.7558871644098333e-07
2	5.186005853362889e-08

Fuente: Elaboración propia.

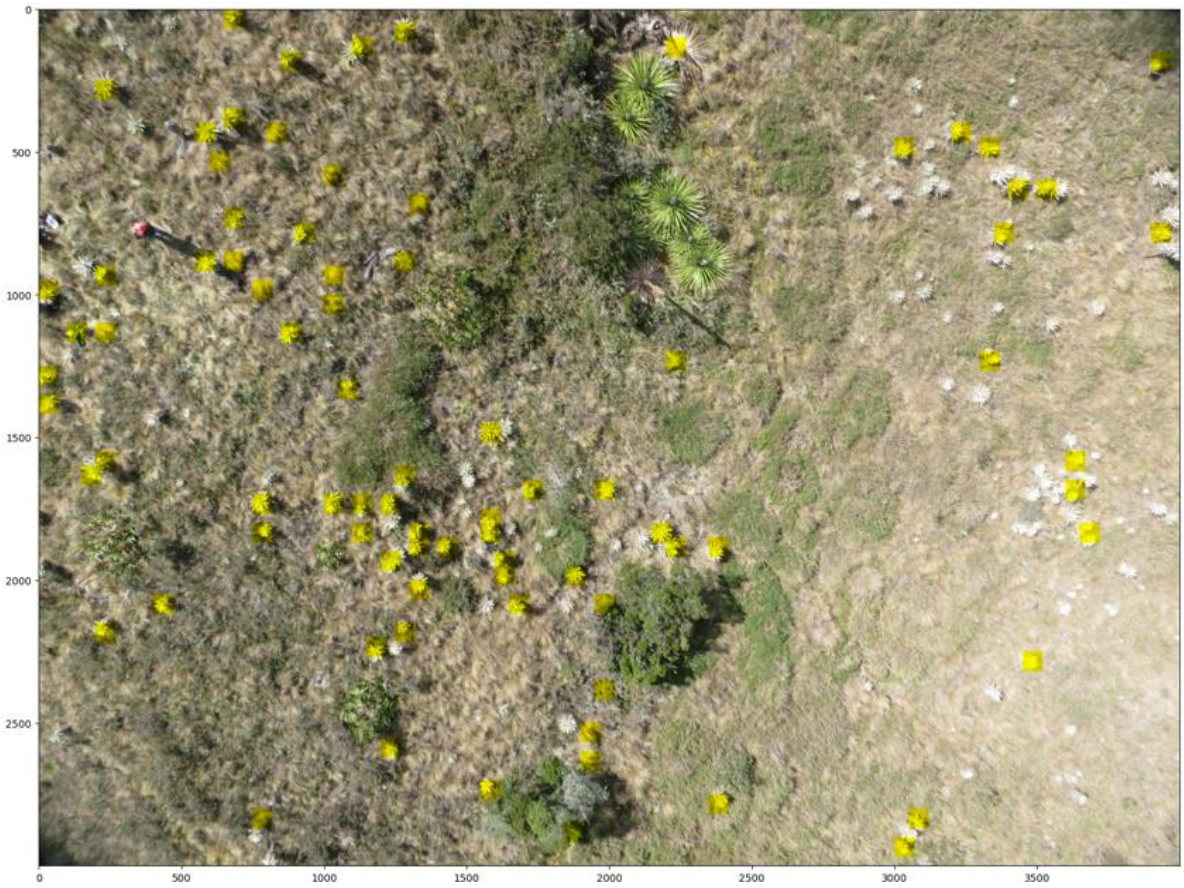
Las densidades son valores de probabilidad que indican la posibilidad de que un punto de predicción, en este caso, un frailejón, pertenezca a un clúster particular en el mapa de características. Dicho esto, y considerando que las densidades obtenidas son significativamente pequeñas, se concluye que la densidad de los puntos dentro de los clústeres es baja, siendo el clúster 2 el que presenta la menor densidad; esto implica que en este clúster los frailejones están más dispersos o hay una menor cantidad de ellos. Esto además se puede apreciar en la prueba cualitativa de la imagen de referencia, donde se observa que la parte derecha del páramo está más desolada.

Ahora bien, aunque las densidades de los clústeres 0 y 1 son ligeramente superiores a las del clúster 2; se pueden concluir dos puntos importantes. Primero, la mayor concentración de frailejones está en la parte izquierda del páramo, en proporciones relativamente parecidas. En segundo lugar, la densidad de frailejones en todo el páramo es baja, lo cual es coherente con el contexto del proyecto, el cual tiene como propósito apoyar la preservación y restauración del ecosistema en áreas con baja densidad de frailejones.

BIBLIOGRAFÍA

- DataScientest. (2024). VGG:¿Qué es este modelo?. Disponible en: https://datascientest.com/es/vgg-que-es-este-modelo-daniel-te-lo-cuenta-todo
- Corporación Autónoma Regional de Cundinamarca (CAR). (2024). ¡Alerta ambiental! Plantar frailejones fuera de su hábitat podría generar alteraciones ecológicas.
- Lacomi, H., Lavorato, M. & Urbano, N. (2022). B-VGG16: Red Neuronal de ConvoluciÓn cuantizada binariamente para la clasificación de imágenes. Revista elektron, 6 (2) pp. 107-114
- Unidad Nacional para la Gestión del Riesgo de Desastres (UNGRD). (2020). Más de 2.000 hectáreas de páramo y 3.000 frailejones destruidos en Boyacá. Disponible en: http://repositorio.gestiondelriesgo.gov.co/handle/20.500.11762/34542

Anexo 1. Resultado de la prueba cualitativa en la red multicapa



Fuente: Elaboración propia