FACULTAD 50 AÑOS DE INGENIERÍA UNaM
2024 JIDeTEV ISSN 2591-4219
XIV Jornadas de Investigación, Desarrollo Tecnológico, Extensión, Vinculación y Muestra de la Producción

Optimización del Análisis de Imágenes de Cámaras Trampa para Fauna: Modelos de Agrupamiento y Recuperación Basados en IA

Axel A. Skrauba [ab] *, Sergio E. Moya [b]

[a] *GIDE, IMAM UNaM-CONICET, Oberá, Misiones, Argentina.*
[b] *Universidad Nacional de Misiones, Facultad de Ingeniería, Oberá, Misiones, Argentina.*
e-mails: axel.skrauba@fio.unam.edu.ar, sergiomoya@fio.unam.edu.ar

Resumen

Este estudio propone el uso del modelo CLIP-ViT-B-32 para optimizar el análisis de bancos de imágenes masivos capturados por cámaras trampa, con el objetivo de filtrar falsos disparos provocados por el movimiento de la flora y otros factores ambientales. Se utiliza un enfoque multimodal que combina *embeddings* visuales y textuales para lograr una clasificación eficiente y precisa de imágenes, reduciendo significativamente el tiempo y trabajo de investigadores expertos a cargo de la clasificación de la fauna fotografiada.

Palabras Clave *– Aprendizaje Profundo, Biodiversidad, Cámaras Trampa, Clustering, Conservación de Fauna, Filtrado de Imágenes, Modelo CLIP-ViT-B-32*

1 Introducción

1.1 Contexto del Problema

El monitoreo de la fauna silvestre es fundamental para la conservación de la biodiversidad y el estudio de los ecosistemas a nivel global. Las cámaras trampa son equipos que, mediante un sensor de movimiento y una cámara digital, pueden capturar imágenes de todo aquello que se mueva en su área de influencia. Estos instrumentos han emergido como una herramienta indispensable para la captura automática de imágenes de vida silvestre en su hábitat natural sin intervención humana [1]. Sin embargo, estas cámaras en la gran mayoría de los casos generan un enorme volumen de fotografías, muchas de las cuales son irrelevantes para los objetivos de monitoreo debido a falsos disparos provocados por el movimiento de la flora, cambios de luz y otros factores ambientales [2].

Los falsos disparos no solo aumentan el volumen de datos a procesar, sino que también representan un desafío significativo para los investigadores, quienes deben invertir tiempo considerable en la revisión y filtrado manual de las imágenes [3]. Este proceso laborioso y costoso limita la capacidad de los científicos para enfocarse en el análisis detallado de las imágenes que realmente contienen fauna o el objetivo de estudio específico [4]. Además, la eficiencia del monitoreo se ve comprometida, lo que puede retrasar la toma de decisiones cruciales para la conservación [5].

En la última década, la implementación de técnicas de inteligencia artificial (IA) ha comenzado a transformar el análisis de imágenes de cámaras trampa, facilitando la identificación automática de animales y la filtración de datos irrelevantes [6]. Aún con estos avances, persisten desafíos significativos. Los métodos actuales a menudo requieren grandes volúmenes de datos etiquetados y

* axel.skrauba@fio.unam.edu.ar

pueden no generalizar bien a diferentes condiciones ambientales [7]. Además, la alta variabilidad en las condiciones de captura y la presencia de múltiples especies en entornos complejos dificultan la implementación de soluciones robustas y escalables [8].

1.2 Estado del Arte

El uso de modelos de aprendizaje profundo ha demostrado ser efectivos en la clasificación de imágenes de fauna, con redes neuronales convolucionales (CNN) liderando este avance [9]. Modelos como Inception, ResNet y EfficientNet han sido ampliamente utilizados para detectar y clasificar especies en imágenes, logrando altos niveles de precisión en condiciones controladas [10]. Sin embargo, estos modelos requieren una gran cantidad de datos de entrenamiento etiquetados, lo que no siempre es viable en entornos de campo reales en donde no se cuente o se pueda siquiera pensar en disponer de datos correctamente etiquetados, ya sea por cuestiones de costos o simplemente por ausencia de fotografías en caso de especies raras o desconocidas [11].

Una solución emergente en este contexto es el uso de modelos multimodales como CLIP (*Contrastive Language–Image Pretraining*), que combinan el procesamiento de imágenes y texto para mejorar la capacidad de los modelos de IA para entender y clasificar imágenes sin necesidad de una gran cantidad de datos etiquetados [12]. CLIP ha demostrado una capacidad notable para generalizar a nuevas tareas y entornos, superando en muchos casos a los modelos supervisados tradicionales [13]. En particular, la integración de CLIP con arquitecturas de *Transformer* como *Vision Transformer* (*ViT*) ha permitido una mejora significativa en la extracción de características de imágenes complejas y en la discriminación de patrones visuales [14].

El modelo CLIP-ViT-B-32, en particular, ha emergido como una herramienta prometedora para el análisis de imágenes en contextos de alta variabilidad, como los entornos de cámaras trampa. Este modelo combina la robustez del aprendizaje contrastivo con la capacidad de *ViT* para manejar datos visuales de alta complejidad, proporcionando una solución eficiente y escalable para el análisis automatizado de grandes volúmenes de datos de imágenes.

1.3 Justificación del Artículo

Dado el creciente volumen de datos generados por cámaras trampa y la necesidad urgente de mejorar la eficiencia en las tareas de monitoreo de fauna, este artículo propone explorar la implementación de CLIP-ViT-B-32 como una solución innovadora para el filtrado y análisis de imágenes. La capacidad de este modelo para generalizar a nuevas tareas sin necesidad de datos de entrenamiento extensivos lo posiciona como una herramienta clave para superar las limitaciones actuales en el procesamiento de imágenes de cámaras trampa.

El objetivo de este estudio es doble: en primer lugar, evaluar el desempeño de CLIP-ViT-B-32 en la reducción de falsos positivos y la mejora de la detección de fauna en imágenes capturadas por cámaras trampa; y, en segundo lugar, proporcionar una base para futuras investigaciones que busquen

aplicar modelos multimodales en el análisis de datos ecológicos complejos, como ser fotogramas de videos de cámara trampa, detección, clasificación y agrupamiento de eventos sonoros, etc. Este enfoque no solo busca mejorar la eficiencia operativa de los investigadores especialistas en fauna, sino que también busca contribuir significativamente a la conservación de la biodiversidad mediante una monitorización precisa y oportuna de la fauna.

2 Metodología

2.1 *Conjunto de Datos*

El conjunto de datos utilizado en este estudio comprende un extenso banco de imágenes capturadas por cámaras trampa en el Parque Federal Campo San Juan, ecorregión de pastizales del sur de la provincia de Misiones. Esta área presenta un desafío particular para el análisis automatizado debido a la alta incidencia de falsos disparos de la cámara causados por el movimiento de los pastos debido al viento. El conjunto de datos consta de aproximadamente 30.000 imágenes recolectadas durante agosto del 2023, abarcando diversas condiciones climáticas y horarios. Algunos ejemplos se observan en la Fig. 1, denotando la diversidad de condiciones y algunas especies capturadas.

Las imágenes fueron capturadas utilizando cuatro cámaras trampa de la marca Bushnell, cuyas imágenes fueron cedidas con fines de investigación por el equipo técnico del área protegida. Cada cámara fue configurada para activarse mediante detección de movimiento y fueron distribuidas estratégicamente para maximizar la cobertura y representatividad del ambiente natural bajo estudio.

Fig. 1. Ejemplos de imágenes de cámaras trampa sin procesar, ilustrando (a) falsos disparos causados por movimiento de pastizales, (b) fauna objetivo capturada con éxito y poca luz, (c) fauna capturada con condiciones ambientales desafiantes como niebla, y (d) falso disparo nocturno.

2.2 Preprocesamiento de Datos

El preprocesamiento de las imágenes se llevó a cabo para optimizar la calidad y consistencia de los datos de entrada para el modelo CLIP. Este proceso constó de las siguientes etapas:

- **Normalización de Tamaño**: Todas las imágenes fueron redimensionadas a una resolución uniforme de $224x224$ píxeles, manteniendo la relación de aspecto original mediante relleno con negro cuando fue necesario. Esta estandarización es crucial para la compatibilidad con la arquitectura de entrada del modelo, ya que, de lo contrario, se producirían distorsiones al rescalar con un *aspect ratio* distinto al del modelo.
- **Ajuste de Contraste**: Se aplicó un algoritmo de ecualización de histograma adaptativo con contraste limitado (*CLAHE, Contrast Limited Adaptive Histogram Equalization*) para mejorar el contraste local de las imágenes, especialmente en aquellas capturadas en condiciones de baja iluminación o niebla, ídem para el amanecer o atardecer. En la Fig. 2 puede apreciarse un ejemplo de este proceso.
- **Eliminación de Metadatos**: Se removieron los metadatos incrustados por las cámaras en las imágenes.

Fig. 2. Ejemplo de ajuste de contraste adaptativo, ilustrando (a) imagen original, (b) imagen procesada.

2.3 Extracción de Características con CLIP-ViT-B-32

Para la extracción de características de las imágenes preprocesadas, se empleó el modelo CLIP-ViT-B-32. Se utilizan los pesos pre entrenados, ya que lo que interesa es poder representar al conjunto de imágenes en el espacio latente del modelo, es decir; estructurar la información.

Por lo tanto, las imágenes procesadas se introdujeron en el codificador de visión del modelo y se extrajo el *embedding* de la capa final del *transformer*, resultando en un vector de características de 512 dimensiones para cada imagen. Los vectores de características resultantes se normalizaron utilizando la norma L2, para garantizar la consistencia en las comparaciones subsiguientes.

Además, la naturaleza multimodal de CLIP, que permite asociaciones entre texto e imagen, ofrece la ventaja adicional de facilitar consultas basadas en texto para el filtrado posterior de imágenes, una característica que se explorará en etapas subsiguientes del análisis. En pocas palabras, esto significa que puede mapear texto en múltiples idiomas, al mismo espacio latente que las imágenes procesadas.

2.4 Algoritmo de Clustering Basado en Densidad

Para identificar distintas especies de fauna en los datos de imágenes de cámaras trampa, se implementó un algoritmo de *clustering* basado en densidad. Este enfoque es particularmente adecuado para nuestro conjunto de datos, ya que puede descubrir *clusters* de formas arbitrarias y es robusto frente a ruido y valores atípicos, características comunes en imágenes de cámaras trampa.

El algoritmo desarrollado se basa en los principios de DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [15], pero con modificaciones para optimizar su rendimiento en espacios de alta dimensionalidad y para manejar eficientemente grandes volúmenes de datos. El proceso consta de las siguientes etapas:

- **Cálculo de similitudes**: Para cada par de vectores de *embeddings* (e_i) y (e_j), se calcula la similitud coseno (1). Esta medida se elige por su eficacia en espacios de alta dimensionalidad y su invariancia a la escala.

$$sim(e_i, e_j) = \frac{e_i * e_j}{|e_i||e_j|} \quad (1)$$

- **Identificación de vecindarios densos**: Se define un umbral de similitud (ϵ) y un número mínimo de puntos ($MinPts$) para considerar un vecindario como denso. Para cada punto (p), se identifica su ($\epsilon - vecindario(N_\epsilon(p))$, donde:

$$N_\epsilon(p) = q \in D | sim(p, q) \geq \epsilon \quad (2)$$

siendo (D) el conjunto de todos los puntos. Un punto (p) se considera como punto central si:

$$|N_\epsilon(p)| \geq MinPts \quad (3)$$

- **Formación de *Clusters* iniciales**: A partir de cada punto central identificado, se forma un *cluster* inicial (C) mediante un proceso de expansión:
  - Inicializar ($C = p$)
  - Para cada punto ($q \in N_\epsilon(p)$):
    - Si (q) no ha sido visitado, marcarlo como visitado
    - Si (q) es un punto central, añadir ($N_\epsilon(p)$) a la lista de puntos por explorar
    - Si (q) no pertenece a ningún *cluster*, añadirlo a (C)

  Este proceso se repite hasta que no queden puntos por explorar o se alcance un tamaño máximo de *cluster* predefinido.

- **Ordenación y filtrado de *clusters***: Los *clusters* formados se ordenan por tamaño de forma descendente. Para resolver solapamientos y asegurar que cada punto pertenezca a un único *cluster* mayor, se aplica un proceso de filtrado:
    - Iniciar con el *cluster* más grande
    - Para cada *cluster* subsiguiente:
        - Eliminar puntos que ya pertenecen a un *cluster* más grande
        - Si el tamaño resultante es menor que ($MinPts$), descartar el *cluster*

Este algoritmo permite identificar eficientemente grupos de imágenes similares, separando las imágenes de fauna de los falsos disparos y agrupando imágenes de especies similares. Los grupos que se forman no presuponen ningún tipo de distribución (como lo harían algoritmos basados en Kmeans, por dar un ejemplo) y, a su vez, se tienen las potenciales anomalías identificadas.

2.5 Detección de Anomalías o Especies Raras

La detección de anomalías se implementa como un paso complementario al *clustering* para identificar imágenes que podrían representar especies raras o eventos inusuales. El método se basa en la premisa de que las anomalías son puntos que no se ajustan bien a ninguno de los *clusters* identificados.

El proceso de detección de anomalías consta de los siguientes pasos:

- **Cálculo de distancias a centroides**: Para cada punto (p) no asignado a un *cluster*, se calcula su distancia al centroide más cercano:

$$d(p) = \min_{C_i \in Clusters} (1 - sim(p, centroid(C_i))) \quad (4)$$

  donde ($centroid(C_i)$) es el vector promedio de todos los puntos en el *cluster* (C_i)
- **Estimación de umbral de anomalía**: Se utiliza el método de Tukey para definir un umbral de anomalía basado en la distribución de las distancias calculadas:

$$umbral = Q_3 + k(Q_3 - Q_1) \quad (5)$$

  donde (Q_1) y (Q_3) son el primer y tercer cuartil de las distancias, respectivamente, y (k) es un factor de escala. Como el método de Tukey se basa en el uso de percentiles y del rango intercuartil, funciona cuando la distribución de las observaciones no necesariamente sigue una distribución Gaussiana.
- **Identificación de anomalías**: Si bien, con el algoritmo de *clustering* ya se discriminan las anomalías y pueden procesarse por separado, no quita que, por ejemplo, se deseen considerar los puntos más alejados de cada *cluster* como tentativos a revisión manual por un personal especializado. Además, según el caso, el número de anomalías puede ser elevado, y es necesario ranquearlas de alguna manera. Por lo tanto, los puntos con una distancia superior al umbral se clasifican como anomalías plausibles de revisión:

$$anomalias = p | d(p) > umbral \quad (6)$$

- **Ranking de anomalías**: Las anomalías identificadas se ordenan según su grado de "rareza", definido por su distancia normalizada al umbral:

$$rareza(p) = \frac{d(p) - umbral}{umbral} \quad (7)$$

Este enfoque permite identificar imágenes que son significativamente diferentes del contenido típico capturado por las cámaras trampa, potencialmente revelando especies raras, comportamientos inusuales o eventos de interés para los investigadores. Además, en caso de múltiples capturas de la misma especie, las mismas quedan agrupadas en su respectivo *cluster*, facilitando el etiquetado manual posterior.

La combinación del algoritmo de *clustering* basado en densidad y el método de detección de anomalías, proporciona una herramienta poderosa para el análisis automatizado de grandes conjuntos de imágenes de cámaras trampa, permitiendo tanto la categorización eficiente de imágenes comunes como la identificación de contenido potencialmente valioso pero infrecuente.

2.6 Mapeo Texto-Imagen para Filtrado de Usuario

La capacidad multimodal del modelo CLIP-ViT-B-32 se aprovechó para implementar un sistema de filtrado basado en texto, permitiendo a los usuarios buscar y clasificar imágenes utilizando descripciones textuales. Este enfoque se basa en la proyección tanto de imágenes como de texto en un espacio de *embedding* común, facilitando la comparación directa entre consultas textuales y contenido visual. Concretamente, se utilizó CLIP-ViT-B-32-multilingual-v1 para este estudio.

El proceso de mapeo texto-imagen, en resumidas cuentas, consta de computar distancias en el espacio latente, de modo de buscar y ranquear las mejores coincidencias para los vectores estructurados de las imágenes y las consultas textuales. Consiste de los siguientes pasos:

- **Preparación de consultas**: Se define un conjunto de consultas textuales relevantes para el contexto de las cámaras trampa, incluyendo nombres de especies, descripciones de comportamientos y características ambientales (ej., “venado en pastizal”, “zorro corriendo”, “amanecer sin animales”, “ave”, “roedores”, etc.). Esto depende de lo que se desea buscar en el banco de imágenes bajo análisis.
- ***Embedding* de texto**: Cada consulta textual se procesa a través del codificador de texto de CLIP, generando un vector de *embeddings* de 512 dimensiones.
- **Cálculo de similitud**: Para cada consulta, se calcula la similitud coseno entre su *embedding* y los *embeddings* de todas las imágenes del conjunto de datos.
- **Ranking y filtrado**: Las imágenes se ordenan según su similitud con la consulta, permitiendo a los usuarios acceder rápidamente a las más relevantes o descartar resultados por no encontrar similitudes dentro de un umbral definido.

3 Resultados

3.1 *Visualización del Espacio de Embeddings con UMAP*

Se utilizó la técnica UMAP (*Uniform Manifold Approximation and Projection*) [16], para reducir la dimensionalidad de los *embeddings* desde las 512 dimensiones generados por el modelo CLIP a 2 dimensiones, permitiendo una visualización en 2D que facilita la interpretación de las relaciones entre las imágenes.

En la representación con UMAP de la Fig. 3, se observan varios *clusters* bien definidos, cada uno visualizado con un color diferente. La proximidad entre puntos, implica semejanza entre las imágenes que los representan (ver los ejemplos para el *cluster* 5 en la Fig. 3, a simple vista, parecieran ser las mismas imágenes). Además, las anomalías aparecen dispersas y alejadas de los *clusters* principales, lo que sugiere su rareza y diferencia significativa respecto a las imágenes comunes. En el caso concreto del grupo de imágenes analizado en la Fig. 3, aparece un grupo pequeño con características muy próximas (señalado con la flecha roja), lo que implica alta probabilidad de haber capturado a un mismo individuo en varias fotografías.

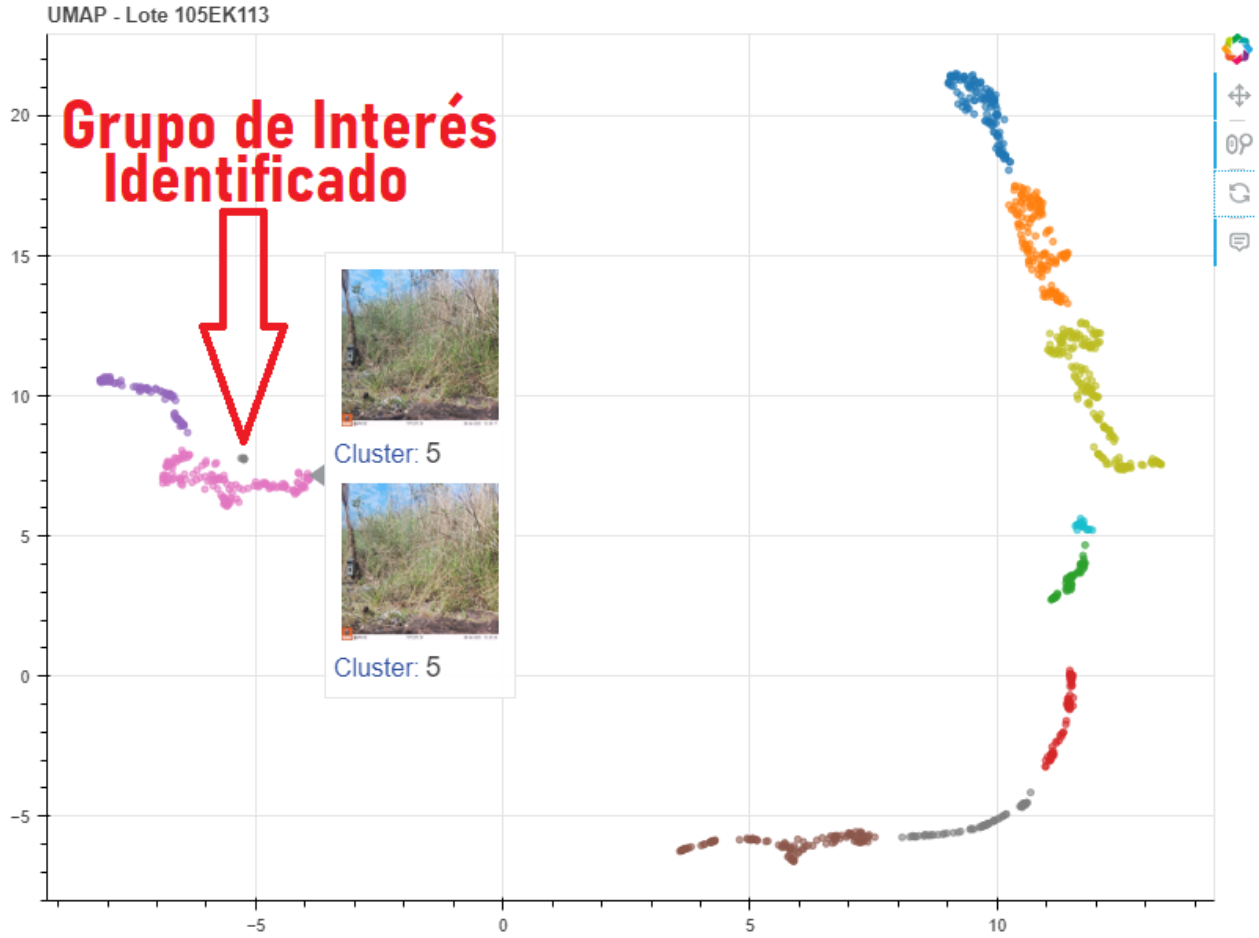

Fig. 3. Mapa de *embeddings* 2D utilizando UMAP, para un lote de 1000 imágenes.

Notar que si bien, la mayoría de las imágenes son muy parecidas (por ser tomas hechas en el mismo sitio), el modelo logra mapear las pequeñas sutilezas presentes en su vector de características, de modo de poder localizarlas con el posprocesamiento adecuado. En este caso, el pequeño grupo localizado, corresponde a una misma ave que ha sido capturada un total de 5 veces y se ejemplifica en la Fig. 4.

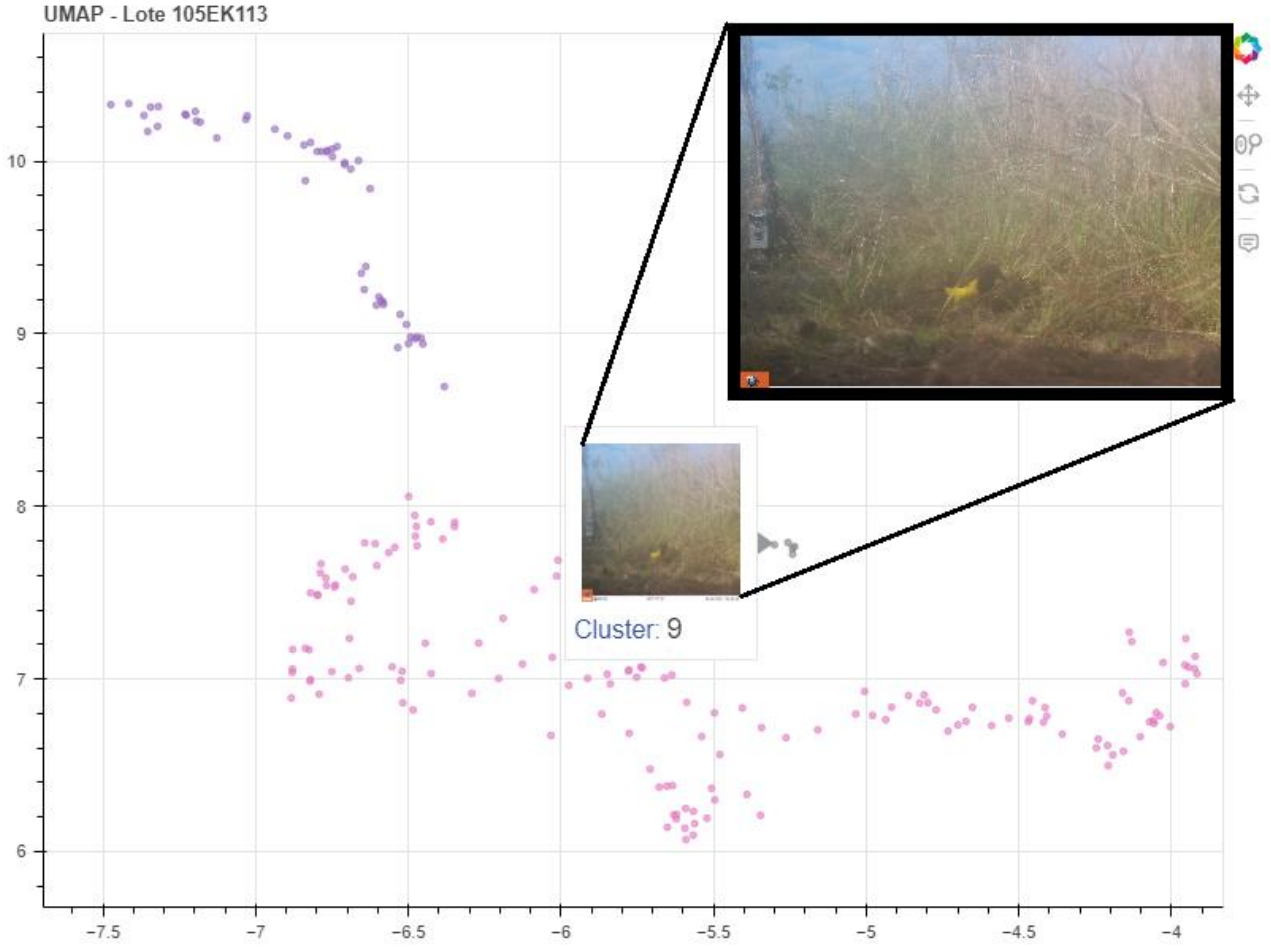

Fig. 4. Zoom sobre mapa de *embeddings* 2D de la Fig. 3, región de interés.

Cada tipo de agrupamiento puede explicarse de ser necesario, es decir, determinar las características que comparten en común. Por dar un ejemplo, en la Fig. 5 se aprecian varias muestras tanto del *cluster* 2 como del 4, mostrando que comparten las proyecciones de las sombras en el suelo, es decir, capturas que son similares en cuanto a la franja horaria de los distintos días. Se ha señalado esto en una de las 4 fotografías de ejemplo para cada *cluster*, y es sencillo de visualizar lo mismo en las demás. Hasta este punto, es evidente que, de las 1000 imágenes, si estuviéramos interesados en aves, solamente 5 eran de utilidad. Una exploración manual, puede ser costosa y extenuante.

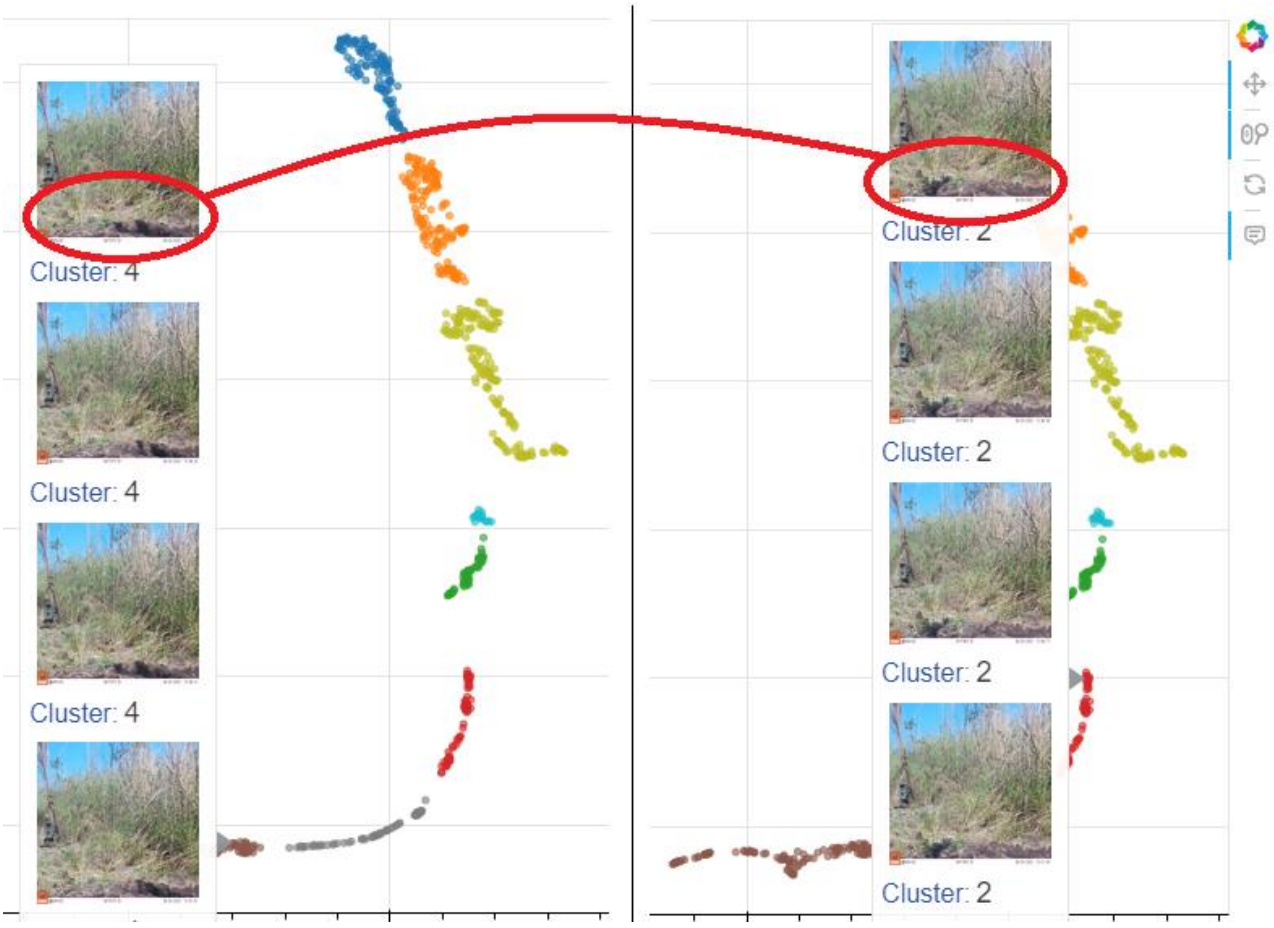

Fig. 5. Zoom sobre mapa de *embeddings* 2D de la Fig. 3, contraste de ejemplares de distintos *clusters*.

3.2 Mapeo de Texto a Imagen

El mapeo de texto a imagen es una funcionalidad crítica que permite la búsqueda y recuperación de imágenes basadas en descripciones textuales utilizando los *embeddings* generados por el modelo CLIP. Esta técnica aprovecha la capacidad del modelo para proyectar tanto texto como imágenes en un espacio de características común, donde la proximidad entre un texto y una imagen indica su similitud semántica.

Para ilustrar esta capacidad, se realizaron búsquedas utilizando términos específicos como "*bird*", "*insect*", "insecto", "*butterfly*" y "mariposa". Estos términos fueron seleccionados por su relevancia y diversidad, lo que permite evaluar la precisión del modelo al identificar imágenes correspondientes a diferentes categorías de fauna, incluyendo conceptos específicos o más abstractos (mariposa vs insecto, por ejemplo). En la Fig. 6, se presentan algunos ejemplos, denotando dónde se encuentra el objeto de interés hallado.

Fig. 6. Ejemplos de resultados de búsqueda para los términos, (a) "*bird*" y (b) "*butterfly*".

Los resultados muestran que el modelo es capaz de recuperar imágenes altamente relevantes para los términos de búsqueda proporcionados. Por ejemplo, al buscar "*bird*", el modelo identificó varias imágenes de aves en diferentes poses (las detectadas anteriormente en el *cluster* pequeño). Del mismo modo, al buscar "*butterfly*", se recuperaron imágenes de mariposas en diversas posiciones y perspectivas. Para este tipo de casos, el problema se asemeja a "buscar una aguja en un pajar", ya que, en un sublote de 1.000 imágenes procesadas, solo 4 contenían mariposas. Además, el área ocupada por el individuo de interés es extremadamente pequeña en comparación con el paisaje general de las imágenes, lo que subraya la complejidad de la tarea. En la Fig. 7 se presentan más ejemplos para este caso. Encontrar manualmente este tipo de objetivos es altamente susceptible a errores humanos, debido al cansancio mental que implica prestar atención a detalles tan sutiles en un número tan elevado de fotografías.

Además, se destaca la capacidad del modelo para manejar la búsqueda en diferentes idiomas. Al utilizar los términos "*insect*" e "insecto", el modelo proporcionó conjuntos de imágenes altamente similares, demostrando su robustez y capacidad para generalizar a través de diferentes lenguajes.

En muchos casos, si el individuo de interés tiene dimensiones pequeñas o sus colores no son perceptibles por "ojos humanos", resulta complicado identificar elementos, incluso en las imágenes ya catalogadas como plausibles de contenerlos. En esos casos, realizar contrastes con imágenes cercanas puede facilitar la tarea. Por dar un ejemplo, en la Fig. 7 (d), sobre la roca señalada, se observa una especie de chicharra que normalmente pasaría desapercibida. Sin embargo, en contraste con los ejemplos (a, b, c) de la misma figura, su presencia se vuelve evidente. Un caso similar, aunque con un contraste inverso, ocurre al localizar las mariposas señaladas.

Este mapeo de texto a imagen no solo facilita la recuperación rápida y precisa de imágenes, sino que también ofrece un enfoque poderoso para el análisis de grandes volúmenes de datos visuales, en contextos donde las anotaciones manuales no son prácticas.

Fig. 7. Comparativa de resultados para términos de búsqueda en inglés y español, (a, b) "mariposa", (c) "insecto", y (d) "*insect*".

3.3 Eficiencia Computacional de la Implementación

La eficiencia computacional es un factor clave al implementar modelos avanzados como CLIP, especialmente en entornos con recursos limitados. Las pruebas se llevaron a cabo en el entorno de Google Colab utilizando únicamente CPU (sin GPU) para emular el rendimiento en un equipo portátil típico.

3.3.1 *Hardware y Configuración*

Las pruebas se realizaron en una instancia de Google Colab con la siguiente configuración:

- Procesador: Intel Xeon CPU con 2 vCPUs a $2{,}20 GHz$.
- Memoria RAM: $13\ GB$.

3.3.2 *Procesamiento de Imágenes*

El modelo CLIP-ViT-B-32 ocupa $605\ MB$ en memoria para almacenar los pesos de la topología. Esta carga de memoria condiciona la cantidad de imágenes que pueden ser procesadas simultáneamente en sistemas con recursos limitados. Es decir, como mínimo deben precargar en RAM los pesos del modelo más las librerías de gestión, y el espacio restante puede emplearse para

precargar los tensores para las imágenes a procesar en lotes. Para simular un entorno típico de un equipo portátil, se procesaron las imágenes en lotes de $96\ (32\ x\ 3)$ para evitar sobrepasar la capacidad de la RAM (de una *notebook* estándar a la fecha, $8\ GB$ de RAM).

- Tiempo promedio de procesamiento por lote: 42 segundos (incluyendo preprocesamiento y extracción de *embeddings*).
- Total de imágenes procesadas en las demostraciones: 1.000.
- Espacio en disco utilizado por las imágenes: $\sim 3\ GB$.
- Espacio en disco utilizado por los *embeddings* generados: $\sim 1{,}95\ MB$.

Es importante destacar que el uso de hardware con más hilos de procesamiento, mayor cantidad de RAM, o hardware de paralelización como GPUs, mejora significativamente estos tiempos.

3.3.3 *Reducción de Dimensionalidad y Clustering*

Para la reducción de dimensionalidad, se utilizó UMAP, que transformó los *embeddings* de 512 dimensiones a un espacio de 2 dimensiones para la visualización.

- Tiempo de procesamiento promedio de UMAP: 4 a 6 minutos (para 1.000 imágenes).

El *clustering* sobre los *embeddings* de 512 dimensiones con el algoritmo implementado fue más que satisfactorio:

- Tiempo promedio de *clustering*: 2 segundos (para 1.000 imágenes).

3.3.4 *Procesamiento de Texto*

El modelo CLIP-ViT-B-32-multilingual-v1 utilizado para procesar texto ocupa $539\ MB$ en disco. Al igual que ocurre con el modelo para imágenes, esto condiciona el mínimo de RAM necesario para la carga del modelo. Aunque las pruebas se realizaron en un entorno sin GPU, los tiempos de inferencia fueron extremadamente rápidos en el procesamiento de texto, en el orden de los ms.

- Tiempos de inferencia para texto: Prácticamente despreciables, incluso al procesar múltiples términos de búsqueda simultáneamente.

3.3.5 *Gestión de Recursos*

Para optimizar el uso de recursos en sistemas con limitaciones de hardware, como podría ser un equipo portátil convencional, es viable cargar los modelos CLIP en sesiones separadas (mayor demora por carga y descarga de modelos a RAM, depende de la velocidad del disco). Esta estrategia permite gestionar la memoria RAM de manera más eficiente, asegurando que los modelos se carguen y utilicen de manera secuencial, sin exceder la capacidad disponible (si fuese necesario implementar todo el *pipeline* en un equipo de menores prestaciones a las mencionadas). Lo importante para el análisis, terminan siendo los vectores de 512 elementos, que, tanto en disco como en RAM, son ínfimos en espacio utilizado, al contrastarlos con fuente de datos original (imágenes).

En resumen, las pruebas realizadas en Google Colab no requirieron más de *8 GB* de RAM en ningún momento, lo que demuestra la viabilidad de implementar este enfoque, incluso en entornos con recursos limitados.

4 Conclusiones

En este estudio se evaluó la capacidad del modelo CLIP-ViT-B-32 para extraer representaciones visuales significativas de imágenes capturadas en un entorno específico, utilizando la técnica UMAP para la reducción de dimensionalidad y visualización de los *embeddings* resultantes. Además, se realizó un análisis cualitativo de las distribuciones en el espacio reducido a 2D, identificando agrupamientos naturales y anomalías dentro de los datos.

El estudio demostró la eficacia del modelo CLIP en la búsqueda y recuperación de imágenes basadas en términos de texto. Lo que proporciona una herramienta invaluable para la exploración y análisis de grandes conjuntos de imágenes, permitiendo búsquedas semánticas precisas y eficientes que pueden adaptarse a diferentes necesidades de investigación y aplicaciones prácticas. Esto resalta el potencial de CLIP en aplicaciones de búsqueda visual y recuperación de información multimodal, donde la precisión y la rapidez en la respuesta son esenciales.

Se evaluó la eficiencia computacional de las implementaciones en un entorno limitado, utilizando los recursos base de Google Colab. A pesar de las restricciones en hardware, el procesamiento de imágenes y texto se realizó de manera efectiva, con tiempos de procesamiento aceptables, tomando como muestras a conjuntos de 1.000 imágenes. Sin embargo, es evidente que, con mejoras en el hardware, como el uso de GPUs o sistemas con más capacidad de RAM, los tiempos podrían reducirse significativamente, haciendo la implementación más escalable y adecuada para aplicaciones en tiempo real o en grandes volúmenes de datos, si es que esto llegase a ser necesario.

Los resultados obtenidos en este estudio no solo validan la capacidad del modelo CLIP para tareas de análisis visual y multimodal, sino que también sugieren varias direcciones para futuras investigaciones. La integración de hardware más potente y la optimización de los algoritmos de reducción de dimensionalidad y *clustering* podrían mejorar aún más la precisión y eficiencia del sistema. Además, la exploración de técnicas adicionales para la detección de anomalías o la diferenciación de *clusters* podría proporcionar *insights* más detallados y valiosos en aplicaciones de monitoreo y análisis ambiental.

Actualmente, se está trabajando en el desarrollo de un software e interfaz gráfica que emplea las técnicas descritas en este estudio. El objetivo de esta herramienta es facilitar las tareas del personal dedicado a la protección, análisis e investigación de la fauna mediante cámaras trampa, proporcionándoles una herramienta eficiente y de fácil uso. Este software está siendo diseñado con un enfoque minimalista, permitiendo a los usuarios explorar grandes conjuntos de imágenes de manera intuitiva, incluyendo la capacidad de filtrar y buscar imágenes mediante términos de texto. Este desarrollo apunta a mejorar la eficiencia en la gestión y análisis de imágenes de fauna, ofreciendo una solución práctica para los desafíos cotidianos en el campo de la conservación y el monitoreo ambiental.

Agradecimientos

Este trabajo ha sido llevado a cabo gracias al aporte de imágenes de las cámaras trampa del Parque Federal Campo San Juan, quienes se vieron interesados en ayudar al desarrollo de alguna herramienta que les permita analizar sus datos. Agradecemos al intendente del Parque, Gimena Martinez y todo su equipo por la confianza en los autores de este trabajo.

Referencias

[1] F. Rovero, F. Zimmermann, D. Bersi, y P. Meek, «"Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications», 2013.

[2] A. C. Burton *et al.*, «REVIEW: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes», *J. Appl. Ecol.*, vol. 52, n.º 3, pp. 675-685, 2015, doi: 10.1111/1365-2664.12432.

[3] R. Steenweg, M. Hebblewhite, J. Whittington, P. Lukacs, y K. McKelvey, «Sampling scales define occupancy and underlying occupancy–abundance relationships in animals», *Ecology*, vol. 99, n.º 1, pp. 172-183, 2018, doi: 10.1002/ecy.2054.

[4] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, y C. Packer, «Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna», *Sci. Data*, vol. 2, n.º 1, p. 150026, jun. 2015, doi: 10.1038/sdata.2015.26.

[5] *Camera Traps in Animal Ecology*. Accedido: 10 de junio de 2024. [En línea]. Disponible en: https://link.springer.com/book/10.1007/978-4-431-99495-4

[6] M. S. Norouzzadeh *et al.*, «Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning», 15 de noviembre de 2017, *arXiv*: arXiv:1703.05830. doi: 10.48550/arXiv.1703.05830.

[7] S. Beery, G. van Horn, y P. Perona, «Recognition in Terra Incognita», 24 de julio de 2018, *arXiv*: arXiv:1807.04975. doi: 10.48550/arXiv.1807.04975.

[8] S. Schneider, G. W. Taylor, S. S. Linquist, y S. C. Kremer, «Past, Present, and Future Approaches Using Computer Vision for Animal Re-Identification from Camera Trap Data», 19 de noviembre de 2018, *arXiv*: arXiv:1811.07749. doi: 10.48550/arXiv.1811.07749.

[9] G. Chen, T. X. Han, Z. He, R. Kays, y T. Forrester, «Deep convolutional neural network based species recognition for wild animal monitoring», en *2014 IEEE International Conference on Image Processing (ICIP)*, oct. 2014, pp. 858-862. doi: 10.1109/ICIP.2014.7025172.

[10] M. A. Tabak *et al.*, «Machine learning to classify animal species in camera trap images: Applications in ecology», *Methods Ecol. Evol.*, vol. 10, n.º 4, pp. 585-590, 2019, doi: 10.1111/2041-210X.13120.

[11] M. Willi *et al.*, «Identifying animal species in camera trap images using deep learning and citizen science», *Methods Ecol. Evol.*, vol. 10, n.º 1, pp. 80-91, 2019, doi: 10.1111/2041-210X.13099.

[12] A. Radford *et al.*, «Learning Transferable Visual Models From Natural Language Supervision», 26 de febrero de 2021, *arXiv*: arXiv:2103.00020. Accedido: 10 de junio de 2024. [En línea]. Disponible en: http://arxiv.org/abs/2103.00020

[13] A. Dosovitskiy *et al.*, «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale», 3 de junio de 2021, *arXiv*: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.

[14] X. Zhai, A. Kolesnikov, N. Houlsby, y L. Beyer, «Scaling Vision Transformers», 20 de junio de 2022, *arXiv*: arXiv:2106.04560. doi: 10.48550/arXiv.2106.04560.

[15] M. Ester, H.-P. Kriegel, J. Sander, y X. Xu, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise».

[16] L. McInnes, J. Healy, y J. Melville, «UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction».