Tarea N°3: Codificadores

Valentina Paz Campos Olguín

I. Resumen

Este informe analiza el uso de tres modelos de codificadores visuales de última generación—ResNet18, CLIP y DINOv2—para la tarea de recuperación de imágenes. Se evalúa su rendimiento en tres conjuntos de datos diversos, comparando la precisión en la recuperación de imágenes mediante métricas como el Mean Average Precision (mAP) y las curvas de precisión-recall. Los resultados obtenidos ofrecen una visión general de la efectividad de cada modelo en diferentes escenarios.

II. Introducción

La recuperación de imágenes es una tarea esencial en visión por computadora, que consiste en identificar y clasificar las imágenes más similares a una consulta dentro de una base de datos. Este proceso se ha vuelto crucial en diversas aplicaciones como motores de búsqueda, sistemas de recomendación, y análisis de imágenes en sectores como la medicina y la seguridad. Para abordar esta tarea, se utilizan modelos de codificación visual que transforman las imágenes en vectores de características, los cuales permiten calcular la similitud entre imágenes.

En este informe, se evaluarán tres modelos de codificadores visuales de última generación: ResNet18, CLIP y DINOv2. Cada modelo será probado en tres conjuntos de datos diferentes (Simple1K, VOC-Pascal y Paris) con el objetivo de comparar su rendimiento en la tarea de recuperación de imágenes. Se utilizarán métricas como el Mean Average Precision (mAP) y las curvas de precisión-recall para evaluar y comparar la efectividad de cada modelo, y así obtener conclusiones sobre cuál ofrece mejores resultados en términos de precisión y relevancia en diferentes escenarios.

Pero para ello, primero se debe saber cómo funcionan estos modelos de forma general:

1. **ResNet18** es una red convolucional que utiliza conexiones residuales, lo que permite que las capas profundas aprendan de forma más eficiente y resuelvan problemas de degradación en redes profundas. Es efectiva en tareas de clasificación y recuperación de imágenes, especialmente en conjuntos de datos sencillos [1].
2. **CLIP** es un modelo multimodal que alinea imágenes y texto en un espacio común. Utiliza un enfoque contrastivo para entrenar ambos codificadores, lo que mejora la recuperación de imágenes cuando se utiliza texto como consulta [2].
3. **DINOv2** usa Vision Transformers (ViT) con un enfoque auto-supervisado. Aprende representaciones visuales sin etiquetas, lo que le permite manejar conjuntos de datos complejos y generalizar bien en tareas de recuperación de imágenes [3].

III. Desarrollo

III-A. Materiales

Tabla I: Implementos utilizados

Materiales	Especificación
Modelo	Lenovo Yoga Slim 7
Procesador	AMD Ryzen 5 4500u
RAM	8GB
Sistema operativo	Linux

El experimento fue ejecutado en una máquina local con las especificaciones anteriormente descritas en la Tabla I.

III-B. Preprocesamiento

Los conjuntos de datos utilizados en este informe son los siguientes:

1. **Simple1K**: Contiene 1326 imágenes distribuidas en 50 clases. Este conjunto de datos está diseñado para evaluar tareas de recuperación de imágenes en un escenario controlado con clases claramente definidas.
2. **VOC-Pascal**: Este conjunto es muy conocido y utilizado en tareas de visión por computadora desde 2010. Para este informe, se utilizará únicamente el conjunto de validación, que contiene 5823 imágenes distribuidas en 20 categorías. VOC es un conjunto de datos más desafiante debido a la variabilidad en las imágenes y las clases más complejas.
3. **Paris**: Este conjunto tradicionalmente se usa para evaluar la recuperación de imágenes en contextos urbanos. En este informe, se utilizará una versión reducida del conjunto original, que contiene 1274 imágenes distribuidas en 12 clases. Este conjunto presenta una variedad de imágenes con complejidades visuales particulares, lo que lo hace ideal para evaluar la robustez de los modelos.

A pesar de la estructura bien definida de estos conjuntos, el desbalanceo de clases es un desafío común en tareas de recuperación de imágenes, especialmente cuando las clases no están distribuidas uniformemente. En los conjuntos de datos utilizados en este informe (Paris, VOC-Pascal y Simple1K), se observa una distribución desigual de imágenes entre las diferentes clases. Como se puede observar en las gráficas de distribución de clases, algunas clases como “general” en Paris, o “person” en VOC-Pascal, tienen una mayor frecuencia que otras.

Sin embargo, en este informe, no se implementaron técnicas de remuestreo (como sobresampling o undersampling) para equilibrar la distribución de clases. La razón de esta decisión es que el objetivo principal de este trabajo es evaluar el rendimiento de los modelos en condiciones naturales de los conjuntos de datos, es decir, tal y como se presentan. Alterar la distribución de clases mediante técnicas de remuestreo podría modificar las características inherentes de los conjuntos de datos, lo que desviaría el enfoque del análisis original.

Para la evaluación, se utilizó la estrategia de evaluación leave-one-out, donde cada imagen se trató como una consulta y se calculó la similitud con las demás imágenes del conjunto de datos. Esto permitió obtener una medición precisa de cómo los modelos manejan las clases desbalanceadas sin intervenciones externas en el proceso de entrenamiento o preprocesamiento.

Este enfoque garantiza que las métricas de evaluación, como el Mean Average Precision (mAP) y los gráficos de precisión-recall (PR), reflejen de manera fiel el desempeño de los modelos ante datos con una distribución desbalanceada.

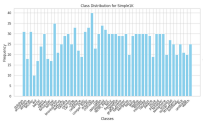

Figura 1: Simple1K

Figura 2: VOC-Pascal

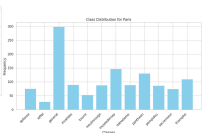

Figura 3: Paris

Figura 4: Distribución de clases para los conjuntos de datos Simple1K, VOC y Paris.

III-C. Diseño del modelo e implementación

En este proyecto, se evaluaron tres codificadores visuales del estado del arte (SOTA) para la tarea de recuperación de imágenes: ResNet18, CLIP y DINOv2. Estos modelos fueron seleccionados por su eficacia en tareas de visión por computadora, particularmente en la extracción de características relevantes de las imágenes y la representación en un espacio latente que facilita la comparación entre imágenes.

1. **ResNet18**: Este modelo se basa en una red neuronal convolucional profunda que utiliza conexiones residuales. Se empleó para extraer características de las imágenes a partir de la arquitectura preentrenada sobre el conjunto de datos ImageNet, lo que proporciona una base sólida para la comparación de características visuales.
2. **CLIP**: CLIP es un modelo multimodal que alinea las representaciones de texto e imagen. En esta tarea, solo se utilizó el codificador visual para extraer características, lo que permite que las imágenes sean representadas en un espacio compartido con texto, facilitando la comparación de imágenes.
3. **DINOv2**: Este modelo de auto-supervisión utiliza un enfoque basado en Vision Transformers (ViT) para generar representaciones de imágenes. DINOv2 se entrena sin etiquetas explícitas y captura información global y local de las imágenes, lo que lo hace adecuado para tareas de recuperación visual.

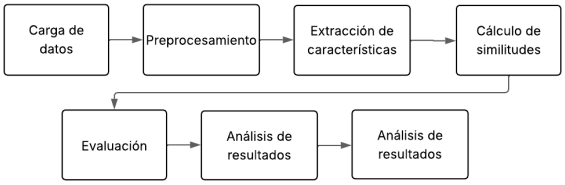

Figura 5: Diagrama de bloques de solución

Para la implementación de estos modelos, se utilizó PyTorch como marco de trabajo para redes neuronales, que proporciona las herramientas necesarias para cargar, preprocesar y procesar las imágenes de manera eficiente. Se utilizaron transformaciones estándar, como el redimensionamiento de imágenes y la normalización según las estadísticas de ImageNet, para asegurar que las imágenes fueran procesadas de manera coherente con los requisitos de cada modelo.

III-D. Estructura de código

El código se organiza de manera modular, con funciones dedicadas a cada tarea específica, desde el preprocesamiento de imágenes hasta la evaluación de los resultados. A continuación, se describe la estructura general del código:

1. **Carga y preparación de los datos**: La primera etapa se encarga de cargar y preparar los datos necesarios para la evaluación. Esto incluye la lectura de las imágenes y sus respectivas etiquetas desde archivos de texto que contienen la ruta de la imagen y la clase a la que pertenece. Cabe destacar que para el conjunto de datos Paris se tuvo que crear las carpetas respectivas de las clases ya que ese es el formato que seguía el archivo de *list of images*, esto permitió cargar adecuadamente el conjunto de datos para su posterior procesamiento. Este paso es fundamental porque permite organizar y acceder a los datos de manera eficiente, lo que facilita el proceso de entrenamiento y evaluación. El conjunto de datos se organiza a través de una clase personalizada que permite la aplicación de transformaciones necesarias (como redimensionar o normalizar las imágenes) de forma flexible.
2. **Preprocesamiento de imágenes**: En esta etapa, las imágenes se procesan para adecuarlas al formato y requerimientos específicos de los modelos de codificación visual. Esto incluye el cambio de tamaño de las imágenes (a 224x224 píxeles) y la normalización de los valores de los píxeles utilizando los valores de media y desviación estándar correspondientes a los conjuntos de datos con los que se entrenaron los modelos preentrenados (por ejemplo, ImageNet). Este preprocesamiento asegura que las

imágenes puedan ser procesadas correctamente por los modelos y facilita la comparación entre ellas.

3. **Extracción de características**: Aquí, cada imagen preprocesada se pasa a través de uno de los modelos visuales seleccionados (ResNet18, CLIP, DINOv2). El objetivo de esta etapa es obtener una representación latente (embeddings) de la imagen, es decir, un vector numérico que resume las características más relevantes de la imagen en un espacio de alta dimensión. Estas representaciones permiten comparar imágenes de manera eficiente, ya que encapsulan sus características visuales en vectores que pueden ser medidos matemáticamente.
4. **Cálculo de similitudes**: Una vez que se han extraído las características de todas las imágenes, la siguiente etapa consiste en calcular la similitud entre la consulta (una imagen específica de la base de datos) y las demás imágenes en el conjunto de datos. Para esto, se utiliza la similitud coseno, una medida que evalúa cuán similares son dos vectores en el espacio latente. Esta medida es crucial para la tarea de recuperación de imágenes, ya que permite identificar las imágenes más cercanas a la consulta según las características visuales.
5. **Evaluación de resultados**: Después de calcular las similitudes, el siguiente paso es evaluar el rendimiento de los modelos en la tarea de recuperación de imágenes. Para ello, se calculan métricas como la Media de Precisión Promedio (mAP), que mide la capacidad del modelo para recuperar imágenes relevantes para una consulta. Además, se generan curvas de precisión-recall (PR) para visualizar cómo varía la precisión del modelo a medida que se ajustan los umbrales de similitud. Estas métricas y gráficas proporcionan una visión clara de cómo cada modelo maneja la tarea de recuperación de imágenes.

IV. Resultados experimentales

IV-A. Vista general del experimento

Para cada conjunto de datos, se utilizó la estrategia de evaluación leave-one-out, en la que cada imagen se considera una consulta y se evalúan todas las demás imágenes en el catálogo para determinar su relevancia. Las imágenes relevantes se definen como aquellas que pertenecen a la misma clase que la consulta, y la similitud se calculó utilizando la similitud coseno entre los vectores de características obtenidos por cada codificador.

IV-B. Precisión de cada modelo por clase

Para evaluar el rendimiento de los modelos en distintos subconjuntos de datos, se calculó la precisión por clase de cada modelo. La precisión por clase es una métrica que indica la exactitud de un modelo en relación con una clase específica, y se calcula como la proporción de predicciones correctas (verdaderos positivos) sobre el total de predicciones realizadas para esa clase, es decir, el total de verdaderos positivos y falsos positivos.

Se utilizaron tres modelos: ResNet18, CLIP y DINOv2, y se calcularon sus precisiones por clase en dos subconjuntos de datos: Simple1K y VOC-Pascal. Estos subconjuntos fueron elegidos para evaluar cómo cada modelo se comporta en diferentes tipos de datos, permitiendo una comparación más rica y detallada.

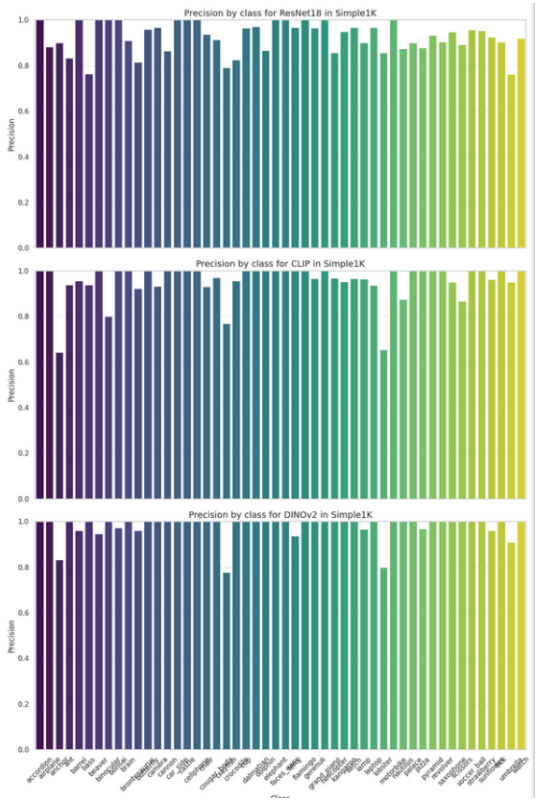

Figura 6: Simple1K

ResNet18 destaca en conjuntos de datos homogéneos como Simple1K, donde su arquitectura de red convolucional profunda permite obtener una alta precisión. Sin embargo, su rendimiento disminuye en conjuntos más complejos como VOC y Paris, debido a la mayor diversidad y complejidad de las clases.

CLIP, al ser un modelo multimodal que combina imágenes y texto, muestra un buen desempeño en conjuntos como Simple1K y VOC-Pascal, donde las descripciones textuales claras mejoran su precisión. No obstante, en Paris, donde las descripciones son menos precisas, su rendimiento decae ligeramente.

Por su parte, DINOv2, basado en aprendizaje auto-supervisado, sobresale en VOC-Pascal y Paris, ya que su capacidad para aprender representaciones visuales sin etiquetas le permite manejar clases más complejas y diversas. Aunque su desempeño en Simple1K es competitivo, es más efectivo en conjuntos más desafiantes.

En términos generales, DINOv2 podría considerarse el modelo más versátil, ya que muestra un rendimiento destacado en VOC-Pascal y Paris debido a su capacidad para aprender representaciones visuales robustas. ResNet18 es más adecuado para conjuntos de datos más sencillos y homogéneos como Simple1K, donde las clases son claras y

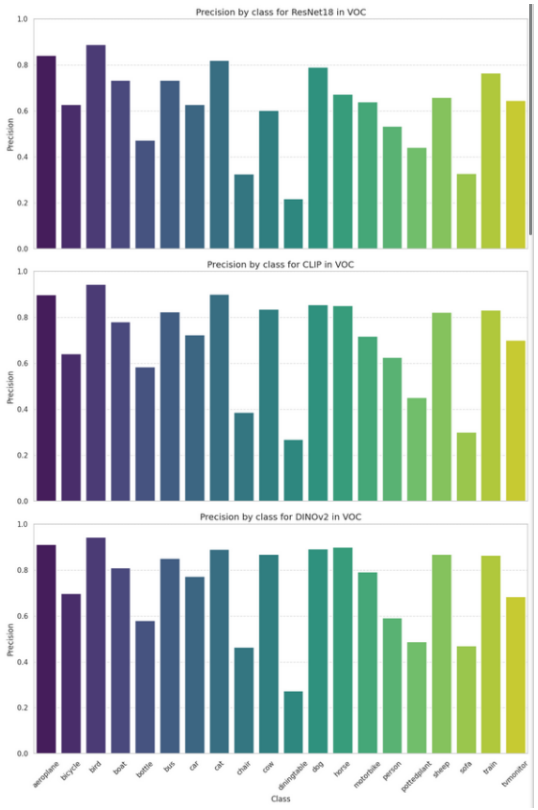

Figura 7: VOC

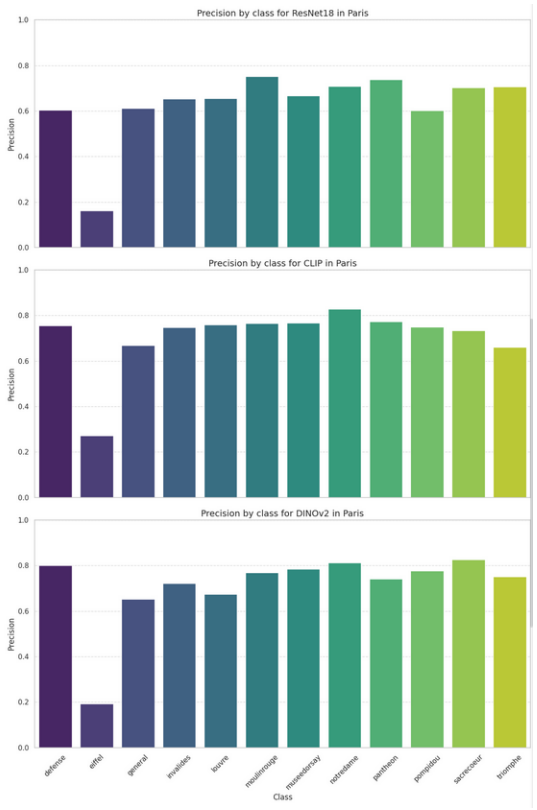

Figura 8: Paris

fáciles de diferenciar. Finalmente, CLIP brilla cuando se tienen descripciones textuales claras y bien definidas para cada clase, lo que lo hace eficaz en tareas donde el texto complementa la imagen, aunque puede tener dificultades con datasets más complejos y difíciles de describir. Cada modelo tiene sus fortalezas dependiendo del tipo de conjunto de datos y la naturaleza de la tarea de clasificación.

IV-C. Resultados de mAP (Mean Average Precision)

Modelo	Simple1K	VOC-Pascal	Paris
ResNet18	0.77	0.39	0.26
CLIP	0.83	0.46	0.32
DINOv2	0.93	0.47	0.40

Tabla II: Comparación de mAP por modelo y conjunto de datos

Considerando las métricas obtenidas de la Tabla II se muestran los resultados de mAP para los modelos ResNet18, CLIP y DINOv2 en los conjuntos Simple1K, VOC-Pascal y Paris.

IV-C1. Calidad

En cuanto a la calidad de la recuperación de imágenes, medida a través del Mean Average Precision (mAP), DINOv2 se destaca como el modelo con el mejor rendimiento, obteniendo un mAP de 0.93 en Simple1K, 0.47 en VOC-Pascal, y 0.40 en Paris. Este desempeño resalta la capacidad de DINOv2 para generar representaciones visuales efectivas, que permiten una correcta clasificación de las imágenes incluso en conjuntos de datos más complejos y con mayor variabilidad visual, como VOC-Pascal y Paris.

CLIP también muestra un buen rendimiento, aunque inferior al de DINOv2, con un mAP de 0.83 en Simple1K, 0.46 en VOC-Pascal, y 0.32 en Paris. La relación entre texto e imagen que utiliza CLIP es muy beneficiosa en conjuntos como Simple1K, donde las clases tienen descripciones textuales claras. Sin embargo, su rendimiento disminuye en VOC-Pascal y Paris, donde la complejidad visual y la ambigüedad de las clases afectan su capacidad de recuperación.

ResNet18, por otro lado, presenta el rendimiento más bajo, con un mAP de 0.77 en Simple1K, 0.39 en VOC-Pascal, y 0.26 en Paris. Esto indica que, aunque ResNet18 es eficaz en tareas con clases bien definidas y conjuntos simples como Simple1K, su capacidad para manejar conjuntos de datos más complejos, con mayor variabilidad visual y desbalanceo en las clases, es limitada.

IV-C2. Robustez

En términos de robustez, DINOv2 demuestra ser el modelo más versátil y robusto, ya que mantiene un rendimiento consistente en todos los conjuntos de datos evaluados. Su capacidad para aprender representaciones visuales sin la necesidad de etiquetas explícitas le permite manejar con éxito clases más complejas y variadas, como las que se encuentran en VOC-Pascal y Paris. Esta propiedad de aprendizaje auto-supervisado es especialmente valiosa cuando las clases son difíciles de definir visualmente o cuando la base de datos contiene imágenes con alta

variabilidad.

CLIP, a pesar de ser competitivo en Simple1K, muestra una caída más pronunciada en rendimiento cuando se evalúa en conjuntos de datos más complejos. Su dependencia de descripciones textuales claras limita su desempeño en escenarios como VOC-Pascal y Paris, donde las clases son más ambiguas y las descripciones no siempre son precisas.

ResNet18 es más efectivo en tareas sencillas con clases homogéneas, como Simple1K, pero su rendimiento disminuye significativamente en VOC-Pascal y Paris, lo que refleja una menor robustez al enfrentarse a conjuntos más complejos y diversos. Esto sugiere que ResNet18 es adecuado para escenarios donde las clases son claras y fáciles de distinguir, pero no maneja bien la variabilidad de las clases en conjuntos más complejos.

IV-D. Resultados 5 mejores y peores queries

En el conjunto de datos Simple1K, ResNet18 sobresale en la clase “car side”, logrando AP perfectos, lo que indica que las imágenes de esta clase son fácilmente diferenciables. Sin embargo, presenta bajos AP en las clases “chair” y “bass”, debido a la complejidad visual de estas clases. CLIP también tiene buenos resultados en “brain”, pero enfrenta dificultades en “lobster” y “cannon”, donde las características visuales son menos claras. DINOv2 destaca en “brain” con AP perfectos, pero muestra bajos AP en “cup” y “palace”, lo que sugiere una mayor variabilidad visual en estas clases.

Para VOC-Pascal, ResNet18 tiene un buen desempeño en “aeroplane”, pero los AP son bajos en “pottedplant” y “sheep”, probablemente por la baja variabilidad visual en estas clases. CLIP sigue un patrón similar, destacándose en “bird” y “aeroplane”, pero con dificultades en “bus” y “pottedplant”. DINOv2 también obtiene buenos resultados en las clases “bird” y “aeroplane”, pero presenta dificultades en “bus” y “pottedplant”, lo que refleja la similitud entre las imágenes de estas clases.

En Paris, ResNet18 se destaca en “moulinrouge”, pero tiene bajos AP en “eiffel”, probablemente debido a la similitud visual de las imágenes de la Torre Eiffel. CLIP también obtiene buenos resultados en “moulinrouge”, pero tiene dificultades con la clase “defense”, que tiene características menos claras. DINOv2 muestra un rendimiento sobresaliente en “moulinrouge”, pero presenta dificultades con “sacrecoeur”, lo que puede deberse a una representación menos efectiva de las características visuales de esta clase.

IV-E. Curvas Precision-Recall

En los resultados experimentales, se evaluaron las curvas Precision-Recall (RP) de los modelos ResNet18, CLIP y DINOv2 en los 3 conjuntos de datos mencionados anteriormente. En las Figuras 9, 10 y 11 se puede ver a detalle el resultado por cada conjunto de datos.

Para Simple1K, un conjunto homogéneo con clases claramente distinguibles, el modelo ResNet18 mostró una alta precisión desde el inicio de la curva, manteniendo un buen desempeño a lo largo del recall. Esto sugiere que ResNet18 es eficaz en este tipo de datos sencillos. CLIP y DINOv2 también lograron buenos resultados, pero DINOv2 presentó una caída más pronunciada en precisión a medida que aumentaba el recall, lo cual es un comportamiento esperado en este tipo de conjunto más simple, donde los modelos pueden distinguir fácilmente las clases.

En el conjunto VOC-Pascal, que presenta mayor diversidad y complejidad, todos los modelos mostraron una caída más gradual en la precisión a medida que avanzaba el recall. Sin embargo, DINOv2 se mantuvo superior a los demás, destacándose por su capacidad de generalizar mejor en un conjunto más complejo. CLIP y ResNet18 exhibieron un rendimiento más similar entre sí, con un descenso más marcado en la precisión, especialmente en las clases más difíciles de distinguir.

Finalmente, en el conjunto Paris, que presenta un nivel elevado de variabilidad visual y clases difíciles de identificar, DINOv2 mostró una ventaja considerable sobre los demás, con una caída menos pronunciada en la curva de precisión-recall. Este modelo, que no depende de etiquetas explícitas y se basa en el aprendizaje auto-supervisado, logró adaptarse mejor a la complejidad del conjunto. Por otro lado, ResNet18 y CLIP tuvieron un rendimiento inferior, con una caída más rápida en precisión, especialmente en las clases más difíciles de diferenciar.

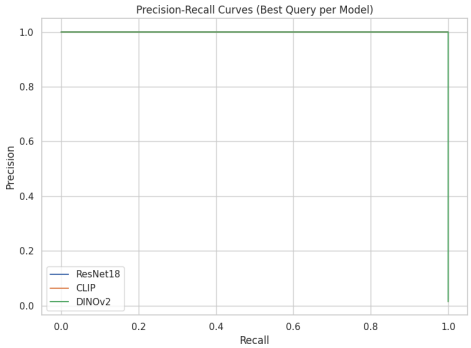

Figura 9: Curva Precision-Recall para el conjunto de datos Simple1K

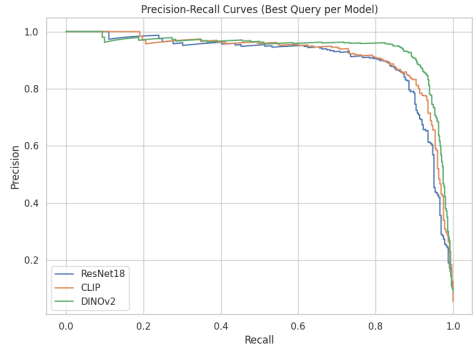

Figura 10: Curva de Precision-Recall para el conjunto de datos VOC-Pascal

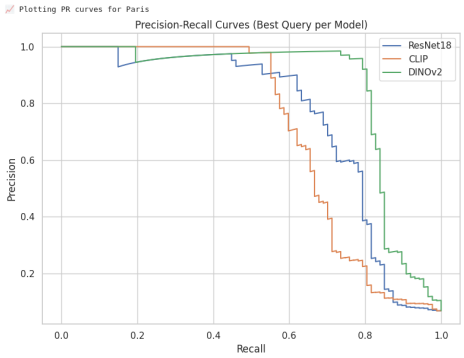

Figura 11: Curva de Precision-Recall para el conjunto de datos Paris

V. Conclusiones

En este informe, se evaluaron tres modelos de codificadores visuales de última generación, ResNet18, CLIP y DINOv2, en tres conjuntos de datos diferentes: Simple1K, VOC y Paris. Cada uno de estos modelos fue probado en la tarea de recuperación de imágenes, donde el objetivo era comparar su rendimiento utilizando métricas como el Mean Average Precision (mAP) y las curvas Precision-Recall (PR).

Los resultados experimentales demostraron que DINOv2 es el modelo más versátil y efectivo en todos los conjuntos de datos. Su enfoque basado en aprendizaje auto-supervisado, que no requiere etiquetas explícitas, le permitió manejar de manera efectiva las clases complejas y diversas presentes en los conjuntos de datos VOC y Paris. En particular, DINOv2 obtuvo los mejores resultados en mAP, con un puntaje de 0.93 en Simple1K, 0.47 en VOC y 0.40 en Paris, reflejando su capacidad para generalizar a través de diferentes tipos de datos y su robustez frente a la variabilidad en las imágenes. Además, las curvas Precision-Recall mostraron un comportamiento consistente, con DINOv2 destacándose especialmente en VOC y Paris, donde otros modelos tenían una caída más pronunciada en precisión.

Por otro lado, ResNet18, aunque mostró un rendimiento impresionante en Simple1K, con un mAP de 0.77, experimentó dificultades en los conjuntos más complejos como VOC y Paris, con un mAP de 0.39 y 0.26, respectivamente. Este modelo, basado en una red convolucional profunda, es muy efectivo para conjuntos de datos homogéneos, donde las clases son fáciles de distinguir. Sin embargo, en conjuntos de datos más diversos, donde las clases tienen más variabilidad visual, su rendimiento disminuye, como se refleja en las métricas obtenidas y las curvas Precision-Recall.

CLIP, siendo un modelo multimodal que combina imágenes y texto, mostró un desempeño intermedio. En Simple1K, donde las clases tienen descripciones textuales claras, CLIP alcanzó un mAP de 0.83, superior al de ResNet18. Sin embargo, en VOC y Paris, donde las imágenes son más complejas y las descripciones textuales no siempre son precisas o claras, CLIP tuvo un rendimiento inferior, con mAP de 0.46 en VOC y 0.32 en Paris. En este sentido, las curvas Precision-Recall mostraron que, a medida que aumentaba la dificultad de las clases, la precisión de CLIP disminuía de manera más pronunciada en comparación con DINOv2.

En términos generales, DINOv2 demostró la mayor capacidad de generalización, con un rendimiento destacado en VOC y Paris, donde las clases son más complejas y difíciles de diferenciar. Su enfoque auto-supervisado, que permite aprender representaciones visuales robustas sin la necesidad de etiquetas explícitas, lo hace especialmente adecuado para tareas de recuperación de imágenes en escenarios donde las clases son más difíciles de definir visualmente. En contraste, ResNet18 sigue siendo eficaz en escenarios más controlados, como Simple1K, donde las clases son claramente distinguibles, pero su rendimiento disminuye en escenarios más variados. CLIP, por su parte, muestra un buen desempeño cuando las clases están bien definidas por descripciones textuales claras, pero su capacidad para manejar clases más ambiguas es limitada en conjuntos como VOC y Paris.

Para futuras investigaciones, se podría explorar la integración de técnicas de optimización, como la reducción de dimensionalidad (por ejemplo, utilizando PCA o UMAP), o el ajuste de hiperparámetros (como el número de capas y el tamaño de las capas en las redes neuronales), para mejorar aún más el rendimiento de los modelos en conjuntos de datos más desafiantes. Además, sería interesante evaluar cómo la combinación de diferentes modelos, como CLIP y DINOv2, podría mejorar el rendimiento en conjuntos de datos complejos mediante enfoques híbridos o multimodales, aprovechando tanto las representaciones visuales como las textuales.

Referencias

[1] He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. Recuperado de https://arxiv.org/abs/1512.03385

[2] Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 8748–8763. Recuperado de https://arxiv.org/abs/2103.00020

[3] Caron, M., Touvron, H., Cord, M., et al. (2023). Emerging Properties in Self-Supervised Vision Transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2860-2870. Recuperado de https://arxiv.org/abs/2304.07193

[4] Radford, A., Kim, J. W. (2021). CLIP: Contrastive Language-Image Pretraining. Recuperado de https://github.com/openai/CLIP