

# Implementación de un sistema de Recuperación de Imágenes Basado en Contenido mediante indexación vectorial con FAISS

José de Jesús Cervantes Rodríguez, Abdelkrim Hadidi

**Abstract**—La creciente digitalización de colecciones artísticas demanda herramientas que permitan recuperar obras por su contenido visual más allá de los metadatos textuales. En este trabajo se presenta el diseño e implementación de un sistema de Recuperación de Imágenes Basada en Contenido (CBIR) orientado al dominio del arte, y se evalúan distintas estrategias de extracción de características. La metodología combina técnicas de aprendizaje profundo bajo un esquema de transferencia de aprendizaje —empleando VGG16, ResNet50 e InceptionV3— con descriptores tradicionales como histogramas de color (RGB) y Patrones Binarios Locales (LBP) para la extracción de características. Para la indexación y búsqueda se utilizó FAISS, garantizando eficiencia en espacios de alta dimensionalidad. Se presentan experimentos comparativos que ilustran las fortalezas y limitaciones relativas de cada enfoque en función de la tipología visual de las obras.

**Index Terms**—Recuperación de imágenes basada en contenido, aprendizaje profundo, transferencia de aprendizaje, extracción de características, FAISS, redes neuronales convolucionales.

## I. INTRODUCCIÓN

La era digital ha transformado completamente la generación y procesamiento de información, convirtiendo a las imágenes en uno de los tipos de datos más predominantes en el tráfico global de internet. Este crecimiento exponencial de datos no estructurados ha incrementado la necesidad de desarrollar herramientas eficientes de indexación y recuperación, evitando que estos recursos permanezcan accesibles pero inencontrables. Aprovechar adecuadamente este vasto contenido gráfico no solo mejora el procesamiento de datos, sino que también posibilita aplicaciones más inteligentes y convenientes en diversos ámbitos [1].

En este contexto, los sistemas de Recuperación de Imágenes Basada en Contenido (CBIR por sus siglas en inglés) emergen como una solución esencial, ya que permiten analizar directamente el contenido visual de los datos. Su diseño implica considerar diferentes factores como los mecanismos de indexación, las medidas de similitud y la elección de las características visuales adecuadas, ya que esto afecta todos los procesos subsiguientes.

La tecnología CBIR aborda dos problemas intrínsecos: (a) cómo describir matemáticamente una imagen (extracción de características) y (b) cómo evaluar la similitud entre un par de imágenes basándose en dichas descripciones. En la práctica, el contenido visual de la base de datos se describe mediante vectores de características multidimensionales [3]. Para recuperar imágenes, el usuario proporciona una imagen de referencia; el sistema extrae su vector de características y lo

compara con los almacenados en la base de datos, recuperando aquellas imágenes cuyos vectores presenten la mayor similitud matemática con la consulta.

Este trabajo propone la implementación de un sistema CBIR que integra cinco extractores de características distintos. Abarcando desde descriptores clásicos de color (Histograma RGB) y textura (LBP), hasta tres arquitecturas basadas en redes neuronales convolucionales. Para gestionar la alta dimensionalidad de los vectores resultantes, se emplea la librería FAISS, optimizando así los tiempos de indexación y recuperación.

## II. METODOLOGÍA

En esta sección se detalla la propuesta metodológica para el desarrollo del sistema CBIR. Para facilitar la comprensión del flujo de trabajo, se presenta primero una visión general de la arquitectura del sistema, seguida de una descripción detallada de cada uno de sus componentes, desde la preparación de los datos hasta la estrategia de indexación.

La Figura 1 muestra la arquitectura general propuesta. El sistema se divide en dos etapas fundamentales distintas: la fase offline y la fase online.

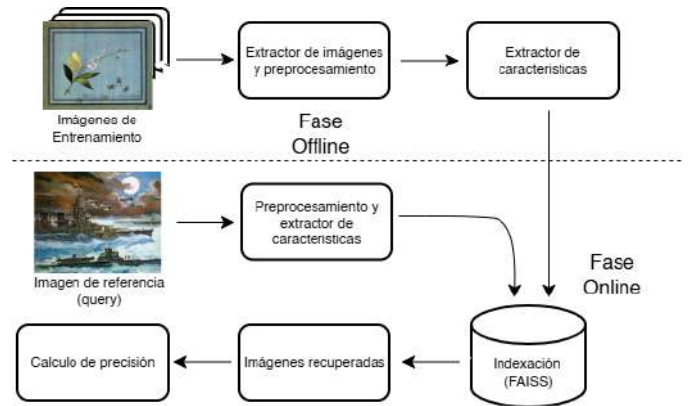


Fig. 1. Arquitectura general del sistema CBIR propuesto, mostrando las fases offline y online.

### A. Conjunto de Datos

Para el desarrollo de este proyecto, se utilizó el conjunto de datos denominado "Art Images: Drawing/Painting/Sculptures/Engravings". Este dataset fue seleccionado debido a su diversidad y a la especificidad

de su categorización, lo cual permitió entrenar modelos robustos capaces de distinguir entre distintos estilos visuales.

El conjunto de datos consta de aproximadamente 9,000 imágenes, distribuidas en cinco categorías principales:

- **Drawings and Watercolours (Dibujos y Acuarelas):** Incluye obras realizadas mediante técnicas secas, como lápiz y carbón, así como técnicas húmedas, principalmente acuarela.
- **Works of Painting (Obras de Pintura):** Abarca pinturas tradicionales, predominando técnicas como el óleo y el acrílico.
- **Sculpture (Escultura):** Contiene representaciones bidimensionales de obras tridimensionales, ya sean talladas o modeladas.
- **Graphic Art (Arte Gráfico):** Engloba grabados, impresiones y otras manifestaciones del arte visual gráfico.
- **Iconography (Iconografía):** Categoría centrada específicamente en el arte antiguo ruso, caracterizado por su marcado estilo religioso y simbólico.

### B. Preprocesamiento y limpieza de datos

Con el fin de asegurar la calidad de la información que ingresa al modelo, se implementó un flujo de trabajo automatizado para la carga y normalización de las imágenes. Este proceso se aplicó de manera sistemática a las cinco categorías del conjunto de entrenamiento, garantizando que los datos cumplieran con las especificaciones necesarias para la extracción de características. Dicho preprocesamiento consistió en las siguientes etapas:

- 1) **Balanceo de clases:** Durante la carga de datos, se estableció un límite uniforme de muestras por categoría (400). Esta medida se tomó para evitar sesgos en el entrenamiento provocados por clases con mayor representación, asegurando así una distribución equitativa entre los distintos estilos artísticos.
- 2) **Validación de integridad:** Se incorporaron mecanismos para identificar y descartar archivos corruptos o ilegibles. Únicamente aquellas imágenes que pudieron ser decodificadas correctamente fueron admitidas en el flujo de trabajo, garantizando la estabilidad del proceso.
- 3) **Estandarización de canales de color:** Como medida de precaución, se verificó que todas las imágenes se encontraran en el espacio de color RGB. Este paso es un requisito indispensable para la arquitectura de los modelos utilizados.
- 4) **Normalización de dimensiones:** Se realizó el redimensionamiento de todas las imágenes a un tamaño estándar fijo de  $224 \times 224$  píxeles. Esto permitió generar tensores de entrada uniformes, ajustando los datos a las dimensiones espaciales requeridas por la red neuronal.
- 5) **Eliminación de redundancias:** Con el objetivo de evitar el sobreajuste causado por la repetición de datos, se aplicó una verificación de unicidad basada en el contenido mediante *hashing*. Se generó una huella digital única para cada imagen; aquellas que presentaron coincidencias exactas con muestras previamente procesadas

fueron identificadas como duplicadas y eliminadas del conjunto.

- 6) **Reducción de ruido y suavizado:** Finalmente, se aplicó un filtro de suavizado a las imágenes. Esta técnica tuvo como objetivo atenuar el ruido de alta frecuencia y homogeneizar las texturas, facilitando que el modelo se enfoque en patrones estructurales relevantes.

Como resultado de esta fase de limpieza y normalización, se obtuvo un conjunto de datos final compuesto por 2,000 imágenes procesadas, las cuales constituyen la entrada validada para la etapa de extracción de características.

### C. Extractor de características

Esta fase constituye un componente imprescindible del proyecto, ya que su objetivo es transformar la información visual cruda —representada por la matriz de píxeles de la imagen— en una representación matemática compacta y densa, denominada *vector de características*.

En el contexto del arte, resulta imperativo que dicha representación capture patrones visuales determinantes pero sutiles, tales como la textura de la pincelada, las variaciones de iluminación y la composición estructural de la obra. Para lograr una caracterización adecuada, se empleó una estrategia que abarca diferentes niveles de abstracción, integrando desde descriptores básicos como histogramas de color y Patrones Binarios Locales (*Local Binary Patterns*, LBP) para el análisis de propiedades estadísticas y de textura, hasta técnicas de *Aprendizaje Profundo*. En este último enfoque, se implementaron Redes Neuronales Convolucionales (*Convolutional Neural Network*, CNN) bajo el esquema de *Transferencia de Aprendizaje*, lo que permitió aprovechar el conocimiento previo de modelos entrenados sobre grandes volúmenes de datos para la identificación de características de alto nivel.

A continuación, se describen en detalle las arquitecturas y algoritmos específicos seleccionados para este sistema, así como las particularidades de su implementación.

1) **Histograma de Color (RGB):** En el ámbito del procesamiento digital, un histograma de imagen se define como una representación gráfica de la distribución tonal presente en una imagen digital [7]. Esta técnica modela la frecuencia de aparición de las intensidades de color, ofreciendo una representación global que resulta ideal para comparar la paleta cromática entre diferentes obras, independientemente de la disposición espacial de los objetos.

Para la extracción de características cromáticas, se desarrolló un procedimiento basado en histogramas de color. El proceso comienza con la normalización de los valores de intensidad de la imagen, asegurando su pertenencia al rango  $[0, 1]$ . Posteriormente, se calcula un histograma independiente para cada canal de color, capturando la distribución global de intensidades. (ver Figura 2)

Finalmente, los histogramas obtenidos se normalizan y concatenan, dando lugar a un vector de características unidimensional de 192 elementos ( $64 \times 3$ ), el cual representa de manera compacta la información cromática global de la imagen.

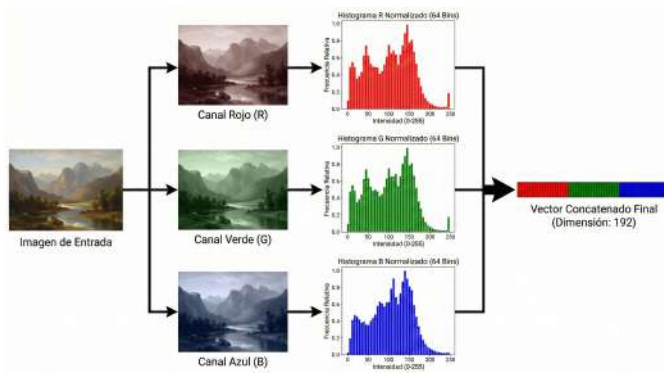


Fig. 2. Esquema de extracción del descriptor de color RGB. Se ilustra el procesamiento paralelo de los tres canales.

2) *Patrones Binarios Locales*: Se incorporó un análisis de textura mediante un LBP. Este operador, propuesto originalmente para la clasificación de texturas, se fundamenta en la descripción de la estructura espacial local de la imagen comparando cada píxel central con sus vecinos circundantes.

Para la extracción de características de textura, las imágenes se transformaron a escala de grises. El operador LBP se configuró con un radio  $R = 3$  y  $P = 24$  puntos de muestreo circular, utilizando la variante *uniform LBP* para reducir la dimensionalidad del descriptor.

A partir del mapa de patrones obtenido, se calculó un histograma normalizado de frecuencias, dando lugar a un vector de características unidimensional de 26 elementos ( $P + 2$ ) (ver Figura 3).

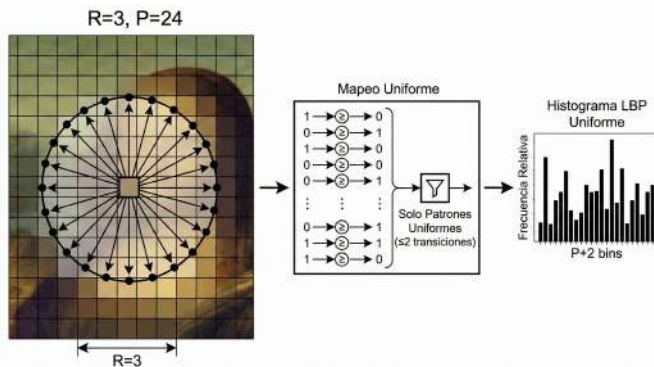


Fig. 3. Extracción de características de textura LBP. El uso de patrones uniformes condensa la información en un vector de 26 dimensiones.

3) *VGG16*: Esta arquitectura se distingue por su profundidad de 16 capas con pesos aprendibles, estructurada secuencialmente mediante bloques que integran series de tres capas convolucionales seguidas por una operación de *Max Pooling* [4]. La implementación conserva únicamente la base convolucional de la arquitectura pre-entrenada con el conjunto `imagenet1k_features`, descartando las capas densas de clasificación final y manteniendo los parámetros congelados en modo de evaluación (ver Figura 4). Como resultado, la red procesa las imágenes para generar mapas de características que, al ser aplanados, producen un vector unidimensional de 25,088 elementos por cada imagen.

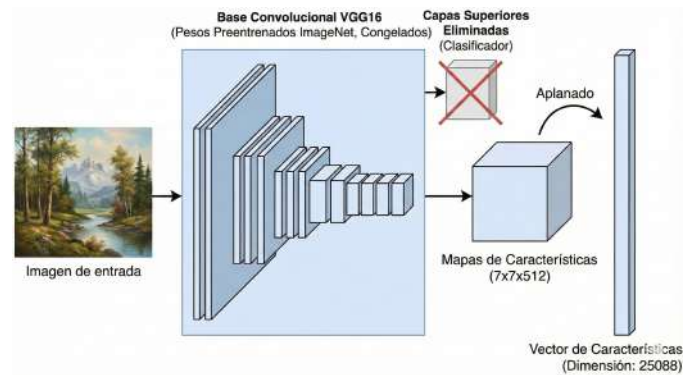


Fig. 4. Esquema de la extracción de características mediante VGG16

4) *ResNet50*: Este segundo modelo, es una red neuronal convolucional de 50 capas compuesta por 48 capas de convolución, una de Max Pooling y una de Average Pooling. Su principal peculiaridad radica en la incorporación de un marco de aprendizaje residual mediante 'conexiones de atajo' (*shortcut connections*), las cuales permiten omitir ciertas capas para mitigar la degradación del error de entrenamiento en arquitecturas profundas [4], [5].

Para la extracción de características, el modelo se inicializó con los pesos preentrenados en `imagenet1k_v2` y se configuró en modo de evaluación, congelando todos sus parámetros. Al igual que en el caso anterior, se eliminó la capa de clasificación final, conservando la estructura de bloques residuales como base de la red (ver Figura 5).

La salida de esta arquitectura, tras el procesamiento y la operación de aplanamiento (*Flatten*), da lugar a un vector de características unidimensional compuesto por 2,048 elementos.

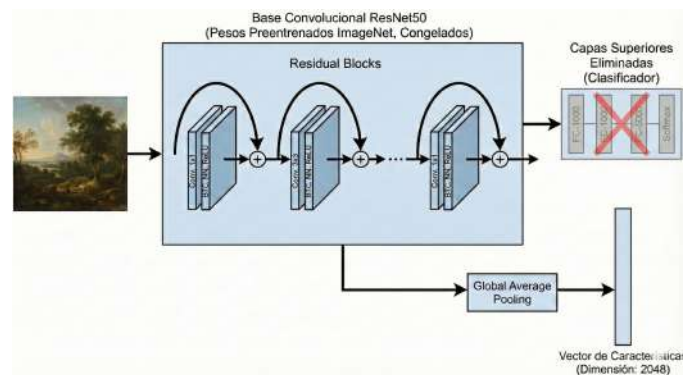


Fig. 5. Esquema de extracción con ResNet50. Destaca el uso de bloques residuales y la obtención del vector de características de 2,048.

5) *InceptionV3*: La particularidad de este modelo reside en el diseño de los módulos *Inception*, los cuales implementan la factorización de convoluciones: en lugar de utilizar filtros espaciales grandes, la red descompone las operaciones en convoluciones asimétricas más pequeñas. Esta estrategia permite reducir significativamente el número de parámetros y el costo computacional, al mismo tiempo que mejora la capacidad de generalización del modelo [6].

En este caso, el modelo se inicializó con los pesos preentrenados IMAGENET1K\_V1 y configurado en un esquema de evaluación. A diferencia de las arquitecturas anteriores, InceptionV3 requiere una resolución espacial mayor para operar correctamente los filtros factorizados; por ello, se integró una etapa de preprocesamiento que aplica interpolación bilineal para redimensionar los tensores de entrada a  $299 \times 299$  píxeles (ver Figura 6).

Estructuralmente, se sustituyó la capa densa original (`fc`) por una capa de identidad (`nn.Identity`), permitiendo el paso directo de las características extraídas tras el agrupamiento global. Finalmente, la salida se somete a una operación de aplanado (*Flatten*), obteniendo un vector descriptor unidimensional compuesto por 2,048 elementos.

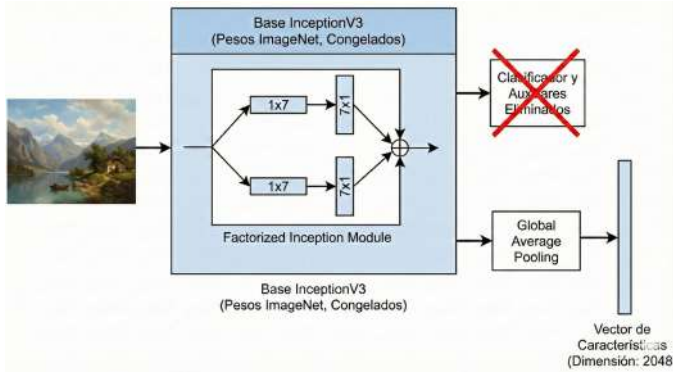


Fig. 6. Esquema de extracción de características con InceptionV3.

#### D. Indexación de Vectores y Búsqueda de Similitud (FAISS)

Para gestionar la recuperación de información en un espacio vectorial de alta dimensionalidad, se implementó la biblioteca FAISS (*Facebook AI Similarity Search*). La estructura fundamental de esta herramienta reside en un índice que admite múltiples implementaciones optimizadas para distintos escenarios de memoria y velocidad [8]. Dicho índice actúa como un almacén estructurado en el que los vectores de la base de datos se agregan de manera progresiva.

En tiempo de búsqueda, el sistema opera recibiendo un vector de consulta (*query vector*). El índice procesa esta entrada y retorna los vectores de la base de datos que se encuentran más próximos a la consulta, generalmente minimizando la distancia euclidiana (ver Figura 7). La arquitectura de FAISS ofrece una alta flexibilidad funcional: además de identificar al vecino más cercano, permite recuperar los  $k$  vecinos más cercanos ( $k$ -NN), realizar búsquedas dentro de un rango de distancia específico y procesar lotes de vectores en paralelo.

Para la construcción de los índices, se realizó inicialmente la conversión de los vectores de características al formato de precisión flotante de 32 bits, requisito indispensable para el funcionamiento correcto de la librería FAISS. Posteriormente, se instanció un índice de tipo `IndexFlatL2`, configurado con la dimensión  $d$  correspondiente a la salida del modelo extractor.

La elección de este tipo de índice se debe a su capacidad de realizar búsquedas exactas mediante el cálculo directo

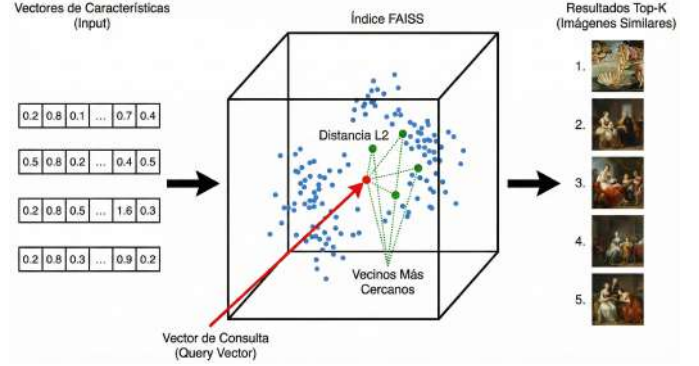


Fig. 7. Esquema de funcionamiento de FAISS. Un vector de consulta ingresa al índice métrico, el cual calcula la distancia Euclidiana para recuperar los  $k$  vectores más similares de la base de datos.

de la distancia euclidiana entre el vector de consulta y la totalidad de los vectores almacenados, sin emplear técnicas de compresión o aproximación que puedan introducir errores en la recuperación. Esta configuración resulta especialmente adecuada para conjuntos de datos de tamaño moderado, donde la prioridad es maximizar la precisión de la similitud.

Asimismo, la utilización de la distancia euclidiana ( $L_2$ ) se justifica por la naturaleza de los vectores de características generados, los cuales se encuentran normalizados. En este contexto, la métrica  $L_2$  permite medir de manera efectiva la proximidad global entre representaciones visuales, siendo ampliamente utilizada y validada en sistemas de recuperación de imágenes basados en contenido.

Una vez poblado con los vectores de la base de datos, los índices fueron almacenadas para su posterior carga y utilización durante la etapa de consulta.

#### E. Proceso de Consulta y Recuperación

Para la operación en tiempo real de nuestro sistema CBIR, se diseñó un flujo de trabajo (*pipeline*) de consulta que garantice la consistencia entre la fase de indexación y la fase de recuperación. Este proceso se estructura en tres etapas secuenciales:

- 1) **Preprocesamiento:** Cuando una imagen de consulta (*query image*) ingresa al sistema, es sometida a las mismas transformaciones de normalización y redimensionamiento aplicadas al conjunto de datos original (véase Sección 2.B). Esto asegura la coherencia del espacio de características.
- 2) **Extracción y búsqueda vectorial:** Se extrae el vector de características utilizando el modelo seleccionado (por ejemplo, VGG16 o LBP) y se proyecta contra el índice FAISS cargado en la base de datos. El sistema ejecuta una búsqueda de los  $k$  vecinos más cercanos ( $k$ -NN), retornando dos arreglos fundamentales: las distancias euclidianas calculadas, que denotan el grado de disimilitud, y los índices posicionales de los vectores dentro de la base de datos.
- 3) **Mapeo de metadatos:** Dado que FAISS opera exclusivamente con identificadores numéricos, se implementó un mecanismo de consulta sobre una tabla de búsqueda

basada en metadatos en formato JSON. Este procedimiento permite traducir los índices recuperados a las rutas de archivo y etiquetas correspondientes, finalizando el proceso con la visualización de las obras de arte más similares.

#### F. Métrica de evaluación: Precisión@k

Para cuantificar el desempeño del sistema de recuperación, se empleó la métrica de Precisión@k ( $P@k$ ). Dado que el conjunto de datos se encuentra etiquetado por categorías (esculturas, pinturas y dibujos), la evaluación se basa en la coincidencia categórica entre la imagen de consulta y los resultados recuperados.

El proceso de evaluación se realiza de la siguiente manera: a partir de una imagen de consulta con categoría conocida  $C_{query}$ , el sistema recupera los  $k$  vecinos más cercanos. La precisión se calcula como la proporción de imágenes recuperadas que pertenecen a la misma categoría  $C_{query}$ , según la expresión:

$$P@k = \frac{\text{número de imágenes relevantes recuperadas}}{k} \quad (1)$$

Esta métrica evalúa la consistencia categórica del sistema. En consecuencia, imágenes visualmente similares que pertenecen a una categoría distinta se consideran falsos positivos, aun cuando reflejen una correcta asociación visual desde un punto de vista perceptual.

### III. RESULTADOS

Se llevaron a cabo evaluaciones empleando imágenes no vistas previamente por los modelos, correspondientes al conjunto de validación, con el objetivo de analizar la capacidad de generalización del sistema CBIR. A continuación, se describe el comportamiento de los distintos extractores de características en función de la tipología de la obra de arte.

#### A. Desempeño de modelos de aprendizaje profundo

En términos generales, las arquitecturas basadas en redes neuronales convolucionales (CNN) demostraron una elevada robustez frente a imágenes nuevas, evidenciando una adecuada capacidad para capturar la semántica global de las obras. No obstante, se identificaron diferencias de desempeño relevantes entre las arquitecturas evaluadas, lo que sugiere una especialización dependiente del tipo de contenido visual.

- **ResNet50:** Esta arquitectura presentó un rendimiento superior en categorías caracterizadas por formas geométricas definidas y estructuras volumétricas, como esculturas y algunos dibujos de rostros. Su diseño basado en conexiones residuales parece favorecer la abstracción de contornos, relaciones espaciales y estructuras tridimensionales, priorizando la forma sobre la información cromática.
- **VGG16:** En contraste, este modelo mostró un mejor desempeño en obras pictóricas, tales como óleos y acuarelas, así como en el arte ruso. Los resultados indican que VGG16 es particularmente eficaz para capturar texturas complejas y patrones de pincelada, superando a ResNet50 en escenas con superficies planas y alta riqueza cromática.

#### B. Eficacia de descriptores tradicionales

El análisis de los descriptores de bajo nivel reveló un comportamiento fuertemente dependiente del contenido de la imagen de consulta. A diferencia de los modelos de aprendizaje profundo, estos métodos no capturan información semántica de alto nivel, sino que operan exclusivamente sobre propiedades visuales locales y globales, como la distribución de intensidades, el contraste y el color.

En el caso del Histograma de Color, los resultados de recuperación se encuentran directamente condicionados por la similitud cromática entre la imagen de consulta y las imágenes almacenadas en la base de datos. Como consecuencia, el sistema tiende a recuperar obras con paletas de color similares, independientemente de su categoría artística o contenido semántico. Esto implica que la calidad de los resultados depende en gran medida de la imagen ingresada: consultas con una paleta cromática distintiva producen recuperaciones visualmente coherentes, mientras que imágenes con colores comunes o poco contrastados generan resultados menos discriminativos.

De manera similar, los Patrones Binarios Locales (LBP) presentan una fuerte sensibilidad a la textura y a la distribución de luminancia. Para imágenes en blanco y negro, como bocetos o grabados, este comportamiento resulta favorable, ya que la recuperación se basa en patrones estructurales y contrastes tonales bien definidos. Sin embargo, en imágenes con mayor complejidad semántica o variaciones estilísticas sutiles, LBP tiende a agrupar imágenes únicamente por similitudes texturales locales, sin distinguir adecuadamente el significado visual de la obra.

### IV. CONCLUSIONES

La implementación y evaluación de este sistema de Recuperación de Imágenes Basada en Contenido (CBIR) para obras de arte nos permitió extraer hallazgos relevantes sobre la aplicabilidad de técnicas de extracción de características. El proyecto demuestra que la estrategia de *Transferencia de Aprendizaje* resulta fundamental para superar la limitación de datos en conjuntos artísticos, permitiendo que modelos entrenados en objetos genéricos (ImageNet) extraigan representaciones útiles de pinceladas, estilos y composiciones complejas.

A partir de los resultados experimentales, se derivan las siguientes conclusiones:

- **Especialización de las arquitecturas profundas:** Los resultados muestran una clara diferencia en lo que cada modelo detecta mejor. ResNet50 es experto en formas y contornos, resultando muy útil para esculturas. Por otro lado, VGG16 retiene mejor la información sobre texturas y superficies, lo que lo hace perfecto para analizar pinturas. Esto nos indica que debemos elegir el modelo según nos interese buscar por forma o por técnica artística.
- **La brecha semántica en descriptores tradicionales:** Si bien los descriptores de bajo nivel (Histogramas y LBP) son computacionalmente eficientes, su incapacidad para interpretar el contexto semántico limita su aplicabilidad general. No obstante, estos métodos siguen siendo el

estándar para nichos específicos de baja complejidad semántica, como el dibujo en blanco y negro, donde superan en precisión y eficiencia a las redes profundas.

- **Eficiencia en la recuperación:** La integración de la biblioteca FAISS demostró que es posible gestionar vectores de alta dimensionalidad (superiores a 25,000 dimensiones en el caso de VGG16) sin comprometer la experiencia del usuario.

En conclusión, deducimos que un sistema CBIR para arte no debe depender de un modelo único, sino tender hacia un enfoque híbrido. La estrategia de seleccionar dinámicamente el extractor de características resulta fundamental para maximizar la similitud visual de los resultados. Al procesar cada obra con el algoritmo que mejor interpreta su naturaleza, se logra una representación numérica mucho más fiel. Esto permite que el sistema encuentre coincidencias más precisas y relevantes, ya que evalúa cada pieza bajo el criterio óptico que mejor la define, evitando las limitaciones de un enfoque generalista.

## REFERENCES

- [1] X. Li, J. Yang, and J. Ma, "Recent developments of content-based image retrieval (CBIR)," *Neurocomputing*, vol. 452, pp. 675–689, 2021.
- [2] K. Le Nhat, "Content-based Image Retrieval," Honor Program, Faculty of Computer Science, University of Information and Technology, Vietnam National University, Ho Chi Minh City, June 2023.
- [3] M. Madugunki, D. S. Bormane, S. Bhadoria, and C. G. Deth, "Comparison of different CBIR techniques," in *Proc. 3rd Int. Conf. on Electronics Computer Technology (ICECT)*, Kanyakumari, India, 2011.
- [4] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification," in *Proc. Int. Conf. on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India, 2021.
- [5] M. A. Aboali, I. Elmaddah, and H. E. Abdelmunim, "Neural textual features composition for CBIR," *IEEE Access*, vol. 11, pp. 28506–28521, 2023.
- [6] S. Degadwala, D. Vyas, H. Biswas, U. Chakraborty, and S. Saha, "Image captioning using Inception V3 transfer learning model," in *Proc. 6th Int. Conf. on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2021.
- [7] F. S. Mohamad, A. A. Manaf, and S. Chuprat, "Nearest neighbor for histogram-based feature extraction," *Procedia Computer Science*, vol. 4, pp. 1296–1305, 2011.
- [8] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The Faiss library," *arXiv preprint arXiv:2401.08281*, 2024.

## CÓDIGO DISPONIBLE

El código fuente y los datos del proyecto están disponibles en el repositorio de GitHub: <https://github.com/JoseCervantes06/CBIR--EquipoC.git>