Preprocesado

* resize

train (100)

5 extractores de características

el preprocesado es offline, la consulta es online

tengo la imagen de test y le tengo que aplicar el extractor con el archivo indice faiss

### **1. Investigación Preliminar**

Primero, investiga los conceptos básicos de los sistemas CBIR (Content-Based Image Retrieval). Esto incluye:

* Qué es CBIR y cómo se diferencia de los sistemas basados en metadatos.
* Los tipos de características visuales más comunes utilizadas (color, textura, forma).
* Métodos y algoritmos de extracción de características, como los histogramas de color, descriptores de texturas y las redes neuronales convolucionales (CNNs).

### **2. Adquisición y Preparación del Conjunto de Datos**

Ya has seleccionado el dataset de Kaggle, que debe contener al menos 100 imágenes (con 10 adicionales para pruebas). Asegúrate de realizar las siguientes tareas de preprocesamiento:

* **Normalización**: Escala los valores de los píxeles para que estén en un rango consistente, como [0, 1].
* **Redimensionamiento**: Asegúrate de que todas las imágenes tengan el mismo tamaño, por ejemplo, 224x224 píxeles (esto es útil si usas una CNN preentrenada).
* **Eliminación de duplicados**: Revisa que no haya imágenes repetidas.

### **3. Extracción de Características**

Vas a usar **cinco extractores de características** para cada imagen. Aquí te doy una guía para cada uno:

1. **Histograma de Color**: Extrae la distribución de los colores presentes en la imagen. Esto se puede hacer dividiendo el espacio de color en bins y calculando el número de píxeles en cada uno.
   * **Implementación**: Usa OpenCV o NumPy para calcular histogramas de colores.
2. **Descriptores de Textura**: Utiliza técnicas como **Patrones Binarios Locales (LBP)** para capturar la textura de las imágenes.
   * **Implementación**: OpenCV ofrece funciones para calcular descriptores LBP.
3. **Descriptores de Forma**: Extrae información de la forma o los contornos de las imágenes, como con **Momentos Hu**.
   * **Implementación**: Usa OpenCV para detectar bordes y calcular los momentos.
4. **Red Neuronal Convolucional (CNN)**: Usa una **CNN preentrenada** (como ResNet o VGG) para extraer características de alto nivel de las imágenes.
   * **Implementación**: Con Keras y TensorFlow, puedes cargar modelos preentrenados y obtener características de las capas intermedias.
5. **SIFT o SURF**: Son métodos basados en puntos clave que describen las características locales de una imagen.
   * **Implementación**: OpenCV proporciona soporte para SIFT o SURF (aunque SIFT puede requerir instalaciones adicionales).

### **4. Normalización de Características**

Es **crucial normalizar** los vectores de características para mejorar la precisión de las búsquedas. Esto se puede hacer usando técnicas como la **normalización Z-score** (restando la media y dividiendo por la desviación estándar) o normalizando en el rango [0, 1].

* **Implementación**: Utiliza Scikit-learn para aplicar la normalización (por ejemplo, StandardScaler o MinMaxScaler).

### **5. Creación del Índice con FAISS**

Aquí es donde usarás **FAISS** para indexar y buscar en el espacio de características. Debes hacer lo siguiente:

1. **Familiarizarse con FAISS**: Estudia la documentación oficial de FAISS para entender los tipos de índices que puedes usar:
   * **Flat Index**: Para búsquedas exactas (recomendado para conjuntos pequeños).
   * **Inverted File (IVF)**: Si tu dataset es grande, esto mejora la velocidad al dividir los datos en clústeres.
   * **Product Quantization (PQ)**: Para búsquedas aproximadas en datasets grandes, utilizando menos memoria.
2. **Creación del Índice**: Una vez que hayas extraído las características de las imágenes y las hayas normalizado, puedes agregar estos vectores al índice FAISS.
   * **Implementación**: Usa FAISS para crear un índice con faiss.IndexFlatL2() o un método más avanzado como faiss.IndexIVFFlat(). Asegúrate de entrenar el índice si es necesario (index.train(vectors)).
3. **Búsqueda en el Índice**: Después de crear el índice, puedes utilizar FAISS para buscar imágenes similares a partir de un vector de características de una imagen de consulta.

### **6. Interfaz de Usuario (UI)**

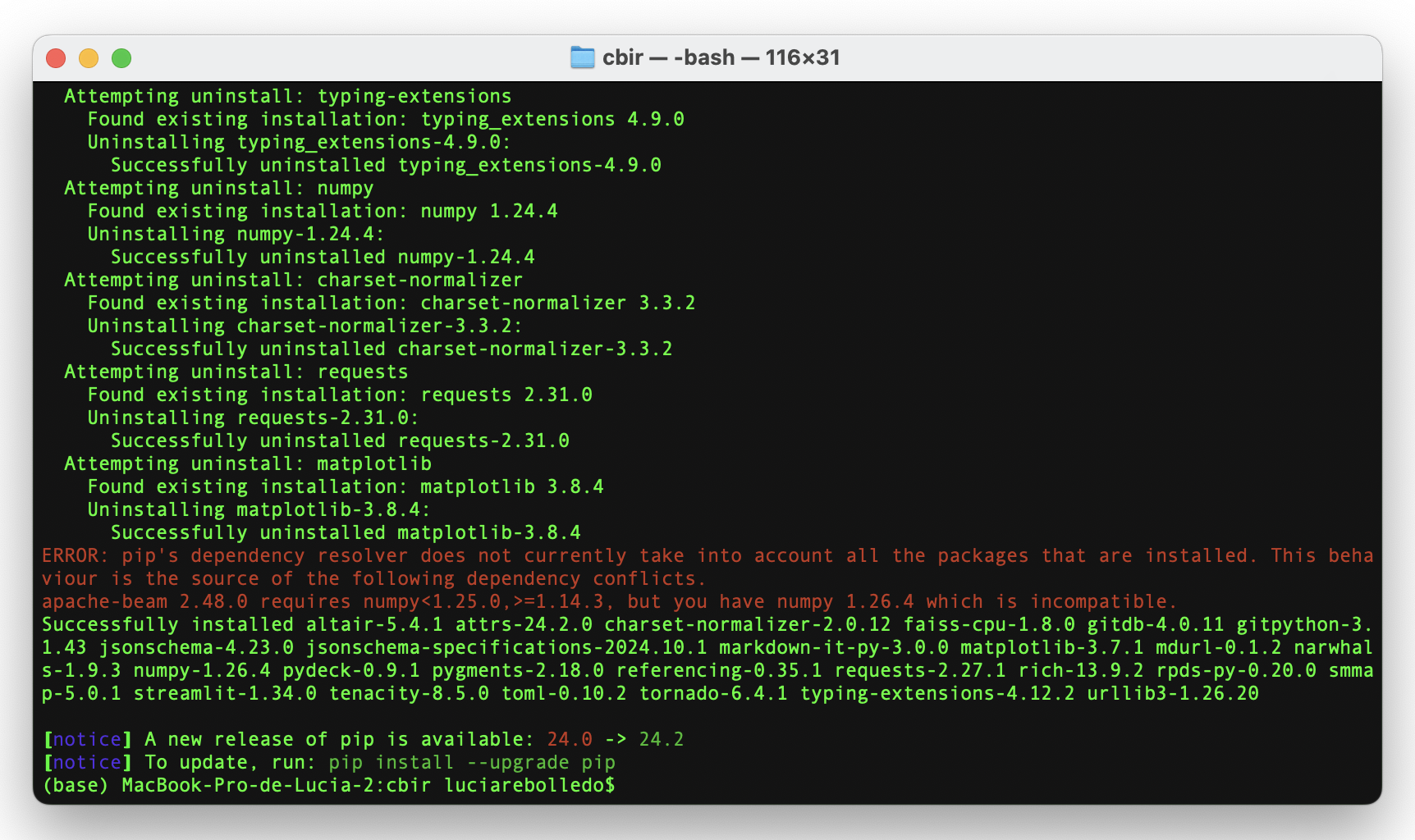
Adaptarás la interfaz proporcionada en Github para mostrar los resultados. Esto implica cargar la imagen de consulta, buscar en FAISS, y mostrar las imágenes más similares basadas en las características extraídas.

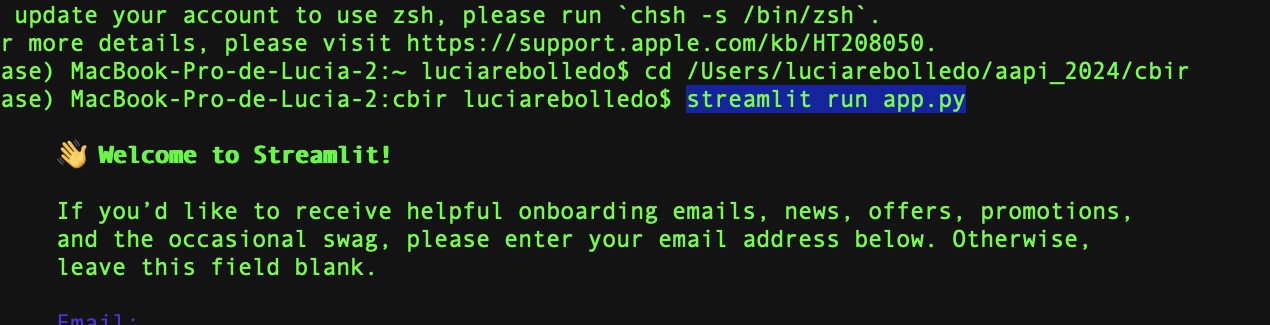
### **7. Evaluación del Sistema**

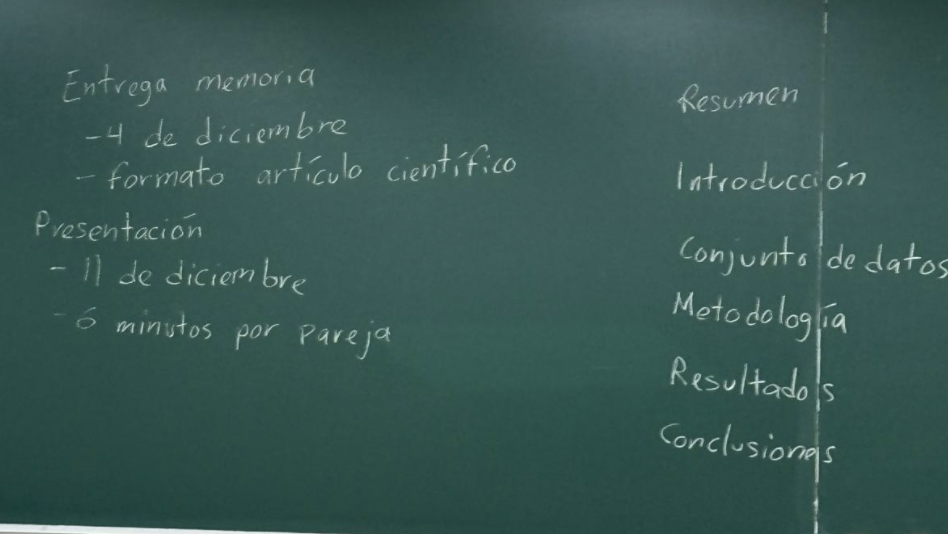
Para medir el rendimiento del sistema CBIR, puedes utilizar la **precisión en los primeros N resultados** (por ejemplo, precisión@5, precisión@10). También es útil comparar los resultados con métodos manuales o etiquetas de clasificación.

### **8. Documentación y Presentación**

Asegúrate de documentar todo el proceso, desde la selección del dataset hasta la evaluación. El código debe estar bien comentado y el proyecto debe subirse a un repositorio público de GitHub. Además, deberás preparar una presentación para exponer los resultados de tu proyecto ante la clase.







* bibliografia

### **Título**

**CBIR en Monumentos: Un Sistema de Recuperación de Imágenes Basado en Contenido**

### **Abstract**

El proyecto describe un sistema CBIR diseñado para recuperar imágenes de monumentos arquitectónicos mediante descriptores visuales. El sistema utiliza diferentes métodos de extracción de características (histogramas de color, texturas, momentos de Hu, redes neuronales convolucionales y SIFT) para representar las imágenes. Estas características son indexadas con FAISS para búsquedas eficientes. El rendimiento se evalúa utilizando métricas como precisión, recall y F1-score. Este enfoque tiene aplicaciones en turismo, catalogación histórica y análisis arquitectónico.

### **Estructura del Documento**

#### **1. Introducción**

* Explica el objetivo del trabajo: la recuperación de imágenes de monumentos.
* Contextualiza la importancia de CBIR en el análisis de datos visuales, turismo y conservación histórica.
* Introduce los métodos utilizados en el sistema.

#### **2. Metodología**

**2.1. Extracción de Características**

* **Color:** Histogramas HSV para representar información de color.
* **Textura:** Local Binary Patterns (LBP) para describir patrones locales.
* **Forma:** Momentos de Hu para la forma geométrica.
* **CNN:** Uso de VGG16 para extraer características profundas.
* **SIFT:** Descriptores basados en características locales.

**2.2. Indexación**

* Describe el uso de FAISS para indexar características y acelerar la búsqueda.
* Explica cómo las características se organizan y se buscan.

**2.3. Métricas de Evaluación**

* **Precision@K, Recall@K y F1-Score:** Cómo se calculan para evaluar el rendimiento del sistema.

#### **3. Implementación**

**3.1. Preprocesamiento**

* Normalización y redimensionamiento de imágenes.
* Transformaciones necesarias para cada extractor.

**3.2. Pipeline de Recuperación**

* Flujo del sistema: desde la imagen de consulta hasta la visualización de resultados.
* Ejemplo del uso de streamlit para la interfaz de usuario.

#### **4. Experimentos y Resultados**

**4.1. Dataset**

* Breve descripción del conjunto de datos utilizado (número de imágenes, categorías).

**4.2. Evaluación**

* Presentación de resultados para cada tipo de descriptor.
* Comparación de descriptores en términos de precisión y velocidad.

#### **5. Conclusión y Trabajo Futuro**

* Resumen de los hallazgos principales.
* Aplicaciones prácticas del sistema.
* Posibles extensiones: añadir otros descriptores, mejorar la evaluación con más datos, o implementar aprendizaje supervisado para clasificar imágenes.

### **Ajustes para la Implementación**

1. **Claridad en el Código**:
   * Divide el código en módulos separados (features\_extractor.py, faiss\_indexing.py).
   * Documenta cada función con comentarios claros.
2. **Interfaz con Streamlit**:
   * Mejora la presentación visual en la interfaz para usuarios no técnicos.
   * Añade gráficos para mostrar métricas como Precision@K y Recall@K.
3. **Expansión del Dataset**:
   * Si es posible, utiliza datasets como Google Landmark para complementar las imágenes.
4. **Resultados Visuales**:
   * En el documento, incluye ejemplos claros de consultas y las imágenes recuperadas para ilustrar el funcionamiento del sistema.

**CBIR (Content-Based Image Retrieval) en Monumentos: Un Sistema de Recuperación de Imágenes Basado en Contenido**

**1. Resumen**

En este proyecto se desarrolló un sistema de Recuperación de Imágenes Basado en Contenido (CBIR) utilizando características visuales como color, textura, forma y descriptores extraídos mediante redes neuronales convolucionales (CNN) y SIFT. Este sistema permite realizar búsquedas de imágenes similares basándose en su contenido visual, en lugar de depender de etiquetas o descripciones textuales.

Para su implementación se empleó Python, FAISS para la indexación eficiente, y una interfaz interactiva en Streamlit que facilita su uso. Se trabajó con un conjunto de imágenes etiquetadas sobre monumentos, preprocesadas para extraer descriptores visuales que alimentaron los índices de búsqueda.

El rendimiento del sistema fue evaluado con métricas como Precision@K, Recall@K y F1-Score@K, obteniendo resultados favorables en la recuperación de imágenes relevantes.

**2. Introducción**

En la actualidad, la cantidad de imágenes generadas y almacenadas digitalmente ha crecido exponencialmente, generando la necesidad de desarrollar sistemas eficientes para la organización y recuperación de información visual. Tradicionalmente, estos sistemas han dependido de etiquetas o descripciones textuales, las cuales presentan limitaciones cuando son insuficientes, erróneas o inexistentes. En este contexto, los sistemas de Recuperación de Imágenes Basado en Contenido (CBIR, por sus siglas en inglés) han surgido como una alternativa innovadora, permitiendo realizar búsquedas basadas en las características visuales de las imágenes.

Este proyecto se centra en el desarrollo de un sistema CBIR diseñado específicamente para identificar y recuperar imágenes de monumentos históricos. Para lograrlo, se implementaron descriptores visuales que capturan diferentes aspectos de las imágenes, incluyendo histogramas de color, patrones de texturas, formas a través de Momentos de Hu, y características profundas extraídas mediante redes neuronales convolucionales (CNN) y SIFT (Scale-Invariant Feature Transform). Estos descriptores fueron gestionados mediante FAISS, una herramienta eficiente para búsquedas en espacios de alta dimensionalidad, y se integraron en una interfaz interactiva basada en Streamlit para facilitar su uso.

El objetivo principal es evaluar la capacidad del sistema para recuperar imágenes relevantes de monumentos como el Coliseo Romano, la Torre Eiffel, las Pirámides de Giza, el Burj Khalifa y el Taj Mahal. Este artículo detalla las etapas clave del desarrollo, presenta los resultados obtenidos y discute las fortalezas y limitaciones del sistema en aplicaciones prácticas.

**3. Conjunto de datos**

El conjunto de datos utilizado en este proyecto fue extraído de la colección "Wonders of the World Image Classification" disponible en Kaggle.

<https://www.kaggle.com/datasets/balabaskar/wonders-of-the-world-image-classification>

Este dataset contiene imágenes de monumentos y maravillas arquitectónicas icónicas de todo el mundo, organizadas en categorías específicas. Su estructura y diversidad lo convierten en una excelente fuente para desarrollar y evaluar sistemas de Recuperación de Imágenes Basado en Contenido (CBIR).

#### Descripción del Dataset

El conjunto de datos incluye muchas imágenes, pero nosotras hemos seleccionado un total de 110 imágenes organizadas en siete categorías que corresponden a monumentos icónicos reconocidos a nivel mundial. Cada imagen está clasificada en una de las siguientes categorías: el Coliseo Romano, la Torre Eiffel, las Pirámides de Giza, el Burj Khalifa, el Cristo Redentor y el Taj Mahal.

Las imágenes están disponibles en formato estándar como JPG y presentan resoluciones variables, lo que permitió realizar un preprocesamiento previo para homogeneizar las dimensiones y optimizar su uso en el sistema.

#### Organización

Se organizó en dos subconjuntos principales para facilitar su uso en el desarrollo y evaluación del sistema CBIR. El primer subconjunto corresponde al conjunto de entrenamiento, que contiene un total de 22 imágenes por categoría, sumando 110 imágenes en total. Estas imágenes fueron utilizadas para construir los índices FAISS y extraer los descriptores visuales necesarios, incluyendo color, textura, forma, características obtenidas con redes neuronales convolucionales (CNN) y descriptores SIFT.

El segundo subconjunto es el de prueba, que incluye 3 imágenes por categoría, totalizando 15 imágenes. Este conjunto fue empleado exclusivamente para evaluar el rendimiento del sistema CBIR mediante la realización de consultas visuales.

**4. Metodología**

El desarrollo del sistema de Recuperación de Imágenes Basado en Contenido (CBIR) se llevó a cabo siguiendo una metodología estructurada que abarcó desde la preparación de datos hasta la implementación y evaluación del sistema. Este apartado describe los pasos realizados y las técnicas utilizadas:

#### 1. Preprocesamiento de Imágenes

Para asegurar la uniformidad y la calidad de los datos utilizados en el proyecto, se implementaron varios pasos de preprocesamiento. En primer lugar, las imágenes del conjunto de datos fueron redimensionadas a un tamaño estándar de 256x256 píxeles, con el objetivo de homogenizar el formato de entrada y facilitar el procesamiento posterior. Además, los valores de los píxeles fueron normalizados, escalándolos al rango [0, 1], lo que permitió una extracción de características visuales más eficiente y consistente.

#### 2. Extracción de Características Visuales

Se emplearon diversos métodos para capturar las características visuales de las imágenes, todos implementados en el archivo 'features\_extractor.py'.

* **Color**: Se emplearon histogramas de color para analizar y representar la distribución cromática de las imágenes. Estos histogramas se generaron en el espacio de color HSV, que es más robusto frente a cambios de iluminación en comparación con el espacio RGB.
* **Textura**: Para describir la textura, se utilizaron Patrones Binarios Locales (LBP), un método que analiza relaciones locales entre los píxeles de una imagen. Esta técnica es particularmente eficaz para identificar patrones texturales repetitivos.
* **Forma**: Las características relacionadas con la forma se capturaron utilizando los Momentos de Hu, que son descriptores geométricos invariantes a rotación, escala y traslación. Estos momentos se calcularon a partir de los contornos de los objetos presentes en las imágenes.
* **Redes Neuronales (CNN)**: Se utilizó un modelo de red neuronal convolucional preentrenado (VGG16) para extraer características profundas de las imágenes. Este enfoque permitió identificar patrones complejos y abstractos de alto nivel, como texturas detalladas, estructuras o combinaciones de elementos visuales, que son difíciles de capturar mediante métodos tradicionales.
* **SIFT (Scale-Invariant Feature Transform)**: Este método permitió detectar puntos clave y calcular descriptores únicos para cada imagen, proporcionando una representación que es invariante a escala, rotación y cambios de iluminación. Además, los descriptores SIFT fueron promediados para generar una representación compacta, facilitando las comparaciones entre imágenes.

Cada uno de estos métodos fue seleccionado por su capacidad de capturar diferentes aspectos visuales, garantizando así un análisis integral de las imágenes en el sistema CBIR.

#### 3. Construcción de Índices FAISS

Para optimizar la recuperación de imágenes en el sistema CBIR, se utilizaron índices FAISS, que permiten realizar búsquedas eficientes en espacios de alta dimensionalidad.

La construcción de estos índices se inició con la creación de un diccionario maestro diseñado para almacenar las características visuales extraídas de las imágenes de entrenamiento. Este diccionario contiene claves específicas para cada tipo de descriptor visual, como color, textura, forma, CNN y SIFT, cada una asociada a un sub-diccionario que relaciona los nombres de las imágenes con sus respectivas características.

Posteriormente, se construyeron los índices FAISS para cada tipo de descriptor utilizando la métrica de distancia euclidiana (IndexFlatL2), ideal para calcular similitudes en espacios vectoriales. La dimensionalidad de cada índice se definió en función del tamaño de los vectores de características generados por cada método.

Una vez construidos, los índices se rellenaron con los vectores de características del diccionario maestro. Al completar este proceso, los índices FAISS contenían todas las características de las imágenes de entrenamiento, permitiendo realizar búsquedas rápidas y precisas basadas en el contenido visual de las imágenes.

#### 4. Implementación de la Interfaz

#### La implementación de la interfaz del sistema CBIR comienza con una configuración inicial que garantiza la integración adecuada de los recursos necesarios para realizar búsquedas y evaluaciones. En este paso, se cargan los índices FAISS correspondientes a los descriptores de color, textura, forma, CNN y SIFT, los cuales fueron previamente generados y almacenados. Estos índices permiten realizar búsquedas rápidas y precisas sobre las características visuales de las imágenes. Además, se carga un archivo CSV que contiene los nombres y las etiquetas de las imágenes del conjunto de entrenamiento, facilitando la comparación y evaluación de los resultados obtenidos durante las consultas.

#### La interfaz proporciona opciones al usuario para configurar los parámetros de cada consulta. En primer lugar, se ofrece la posibilidad de seleccionar uno de los cinco descriptores visuales disponibles (color, textura, forma, CNN o SIFT) en función de las características que se desean priorizar en la búsqueda. Asimismo, se habilita una funcionalidad para subir una imagen desde el dispositivo del usuario, que servirá como consulta para buscar imágenes similares. También, se añade una pestaña en la que el usuario selecciona la categoría a la que pertenece la imagen subida, lo que nos facilitará el cálculo de las métricas. Antes de realizar la consulta, la imagen puede recortarse utilizando una herramienta interactiva incluida en la interfaz, lo que permite enfocar la búsqueda en áreas específicas de la imagen.

#### Tras configurar los parámetros de la consulta, el sistema procede a recuperar las imágenes más similares basándose en las características visuales extraídas de la imagen de consulta. Estas características se analizan mediante el descriptor seleccionado y se comparan con los vectores almacenados en los índices FAISS.

#### 5. Evaluación del Sistema

El rendimiento del sistema se evaluó utilizando métricas estándar:

* Precision@K: Porcentaje de imágenes relevantes entre las K recuperadas.
* Recall@K: Porcentaje de imágenes relevantes recuperadas en relación al total de relevantes disponibles.
* F1-Score@K: Combinación de precisión y recall para medir el balance entre ambos.

Estas métricas se calcularon para diferentes valores de K, permitiendo analizar el comportamiento del sistema con cada descriptor.

#### 6. Visualización de Resultados

Una vez que el usuario sube una imagen de consulta y selecciona el descriptor visual deseado, el sistema realiza una búsqueda en los índices FAISS y recupera las imágenes más similares. Estas imágenes se presentan en un formato de cuadrícula, donde cada resultado incluye una vista previa de la imagen recuperada y su posición en el ranking de similitud. La imagen de consulta se muestra en una posición destacada para facilitar la comparación visual con las imágenes recuperadas.

Para brindar mayor información sobre el rendimiento del sistema, la interfaz también muestra las métricas de evaluación correspondientes, como Precision@K, Recall@K y F1-Score@K, calculadas en función de las imágenes recuperadas.

Finalmente, el tiempo total de procesamiento de la consulta se muestra al usuario, proporcionando un indicador adicional de la eficiencia del sistema.

**5. Resultados**

1. Resultados por categoría

* **Burj Khalifa**: Los mejores resultados para el Burj Khalifa se obtuvieron utilizando los descriptores CNN y SIFT, destacando la importancia de emplear métodos que puedan capturar patrones complejos y puntos clave distintivos en imágenes de estructuras icónicas. Por otro lado, los descriptores basados en color, textura y forma presentaron limitaciones en este caso, debido a las características específicas del edificio y las condiciones de iluminación de las imágenes.
* **Coliseo**: En este caso, SIFT es el método más eficaz porque su enfoque en puntos clave es ideal para monumentos con detalles estructurales y texturas complejas. Los Momentos de Hu y el histograma de color también funcionaron bien al aprovechar las formas distintivas y los colores del Coliseo, respectivamente. En contraste, los descriptores basados en textura y CNN presentaron limitaciones, debido a la diversidad y complejidad visual de las imágenes del monumento.
* **Torre Eiffel**: El descriptor CNN es el más eficaz para identificar imágenes de la Torre Eiffel, debido a su capacidad para capturar patrones complejos y características visuales de alto nivel. Los descriptores basados en color también aportaron buenos resultados, aunque con limitaciones derivadas de las variaciones en iluminación y fondos. En contraste, los descriptores de textura, forma y SIFT tuvieron un desempeño más bajo.
* **Pirámides de Giza**: Los descriptores basados en color y CNN demostraron ser los más efectivos para identificar imágenes de las Pirámides de Giza, gracias a la consistencia de las tonalidades y las características visuales complejas que estos métodos son capaces de capturar. Los Momentos de Hu ofrecieron un rendimiento razonable al capturar las formas geométricas simples, mientras que los descriptores basados en textura y SIFT tuvieron un desempeño más limitado debido a la simplicidad de las texturas y la cantidad moderada de puntos clave distintivos en las imágenes.
* **Taj Mahal**: Los descriptores CNN son los más efectivos para capturar la riqueza visual y los detalles complejos del monumento. SIFT también demostró ser un método eficaz, gracias a su capacidad para identificar puntos clave en los detalles arquitectónicos. En contraste, los descriptores basados en color, textura y forma ofrecieron un rendimiento limitado debido a la simplicidad tonal y la uniformidad de las superficies del Taj Mahal, así como a la complejidad arquitectónica que estos métodos no logran capturar completamente.

2. Resultados por extractor

* **Color**: El histograma de color fue particularmente efectivo para las Pirámides de Giza y, en menor medida, para el Coliseo y la Torre Eiffel. Este método demostró ser más adecuado para monumentos con tonalidades distintivas y consistentes, como las pirámides y el mármol del Taj Mahal. Sin embargo, su rendimiento se vio afectado por variaciones de iluminación y fondos en imágenes de otros monumentos. En general, el color funcionó bien cuando las tonalidades eran un factor clave para la identificación.
* **Textura**: El descriptor de textura basado en Patrones Binarios Locales (LBP) mostró limitaciones significativas en la mayoría de los monumentos, con la excepción de las Pirámides de Giza, donde los patrones texturales de las superficies fueron más evidentes. Para estructuras con texturas regulares y lisas, como la Torre Eiffel y el Taj Mahal, su desempeño fue bajo, reflejando su incapacidad para capturar detalles complejos.
* **Forma**: Los Momentos de Hu destacaron principalmente en el Coliseo, donde las formas curvas y los contornos del anfiteatro proporcionaron características geométricas distintivas. En las Pirámides de Giza y el Taj Mahal, los resultados fueron razonables debido a sus formas simples y simétricas. Sin embargo, este descriptor presentó limitaciones para monumentos con estructuras más complejas, como la Torre Eiffel y el Burj Khalifa.
* **CNN**: El descriptor basado en redes neuronales convolucionales (CNN) fue consistentemente el más efectivo en todas las categorías, sobresaliendo especialmente en el Taj Mahal, las Pirámides de Giza y la Torre Eiffel. Su capacidad para capturar patrones visuales complejos y características de alto nivel permitió una identificación precisa incluso en monumentos con detalles arquitectónicos y decorativos intrincados. Aunque computacionalmente más costoso, su desempeño justificó su uso como uno de los mejores métodos.
* **SIFT**: El método SIFT mostró resultados variados dependiendo del monumento. Fue extremadamente eficaz en el Coliseo, donde la abundancia de puntos clave permitió una recuperación precisa, y tuvo un buen desempeño en el Taj Mahal. Sin embargo, su efectividad disminuyó en monumentos como las Pirámides de Giza y la Torre Eiffel, donde los puntos clave eran más escasos o menos distintivos. A pesar de estas limitaciones, SIFT sigue siendo una herramienta robusta para identificar imágenes con detalles arquitectónicos y texturales complejos.

3. Resultados totales

En general, los resultados del sistema CBIR muestran un desempeño variable dependiendo del monumento y del tipo de descriptor visual utilizado.

Los descriptores basados en redes neuronales convolucionales (CNN) destacaron como los más consistentes y efectivos, logrando identificar imágenes relevantes en todas las categorías gracias a su capacidad para capturar patrones complejos y características de alto nivel.

SIFT también mostró un excelente rendimiento en monumentos con detalles estructurales y texturales distintivos, como el Coliseo Romano y el Taj Mahal, aunque presentó limitaciones en estructuras más simples como las Pirámides de Giza.

Los descriptores basados en color fueron altamente efectivos en monumentos con tonalidades consistentes y características, como las Pirámides de Giza, pero su rendimiento disminuyó en categorías con variaciones de iluminación y fondos más diversos, como el Burj Khalifa.

Por otro lado, los descriptores de textura y forma ofrecieron un desempeño más limitado, sobresaliendo solo en casos específicos donde las características visuales eran particularmente adecuadas para estos métodos, como las texturas de las Pirámides y las formas del Coliseo.

En conjunto, el sistema demostró ser robusto para recuperar imágenes relevantes, con los descriptores CNN y SIFT como las opciones más confiables y versátiles.

4. Análisis de métricas

TABLE I. Métricas

| Monument | Color | Textura | Forma | CNN | SIFT |
| --- | --- | --- | --- | --- | --- |
| Burj Khalifa | 0.27 | 0.09 | 0.09 | 0.72 | 0.81 |
| Eiffel Tower | 0.36 | 0.27 | 0 | 0.81 | 0.36 |
| Colosseum | 0.36 | 0.27 | 0.72 | 0.63 | 0.90 |
| Taj Mahal | 0.27 | 0.27 | 0.27 | 1 | 0.72 |
| Pyramids of Giza | 0.90 | 0.36 | 0.36 | 1 | 0.18 |

**6. Conclusiones**

En este proyecto, se diseñó e implementó un sistema de Recuperación de Imágenes Basado en Contenido (CBIR) enfocado en la identificación de monumentos, utilizando descriptores visuales de color, textura, forma, redes neuronales convolucionales (CNN) y SIFT. Los resultados obtenidos destacan que los descriptores CNN y SIFT fueron las herramientas más efectivas para capturar patrones visuales complejos y detalles arquitectónicos distintivos, demostrando un rendimiento consistente en casi todas las categorías analizadas. Por otro lado, los descriptores basados en color fueron efectivos en monumentos con tonalidades uniformes y características, mientras que los descriptores de textura y forma ofrecieron un rendimiento más limitado.

En resumen, el sistema CBIR desarrollado, al integrar múltiples descriptores, logró una solución robusta y versátil para la recuperación de imágenes relevantes, permitiendo evaluar las fortalezas y limitaciones de cada enfoque.