

Weather Image Recognition Using CBIR Techniques

Alberto Di Salvo y Martina Valdivielso

Abstract—Este trabajo presenta un sistema de Recuperación de Imágenes Basado en Contenido aplicado al Reconocimiento de Imágenes Meteorológicas. El sistema aborda la búsqueda de patrones climáticos similares en grandes colecciones de imágenes satelitales. Se emplea la extracción de características profundas utilizando una CNN pre-entrenada para obtener descriptores visuales robustos. Para asegurar la escalabilidad y velocidad de las búsquedas, se implementa FAISS. Los resultados muestran que la arquitectura CBIR propuesta ofrece una solución eficiente para la organización y recuperación precisa de información visual compleja dentro del dominio de la meteorología.

Keywords—CBIR, CNN, FAISS, Transfer Learning, ResNet-50, IVF-PQ, ANN, LBP, HSV, Reconocimiento de Imágenes Meteorológicas, Clima

I. INTRODUCCIÓN

En el mundo digital actual, la explosión de datos visuales y el crecimiento exponencial de imágenes han planteado desafíos significativos en términos de organización y recuperación de información visual. Este desafío es particularmente crítico en dominios como la meteorología, donde vastas bases de datos de imágenes satelitales, de radar y fotográficas de nubes requieren una clasificación y un análisis preciso y oportuno.

En este contexto, el concepto de Recuperación de Imágenes Basado en el Contenido ha emergido como una técnica poderosa para abordar estos desafíos de manera eficiente y efectiva. A diferencia de los sistemas tradicionales de recuperación de imágenes que dependen de etiquetas o metadatos textuales asociados, CBIR se basa en el contenido visual intrínseco de las imágenes, permitiendo que estas hablen por sí mismas. Este enfoque permite a los usuarios buscar imágenes similares a una consulta en función de sus características visuales, tales como colores, texturas, formas y patrones, superando así las limitaciones del gap semántico inherentes a la descripción textual.

El objetivo central de este proyecto es el desarrollo y la implementación de un sistema funcional de CBIR enfocado en la tarea de Reconocimiento de Imágenes Meteorológicas. Nuestro trabajo aplicará los conceptos teóricos y las técnicas prácticas de procesamiento de imágenes y aprendizaje automático para construir un sistema que aborda el desafío de la búsqueda de patrones climáticos similares en colecciones de imágenes satelitales.

Para lograr una recuperación eficiente, el sistema se basa en la extracción de características profundas utilizando una CNN pre-entrenada combinada con descriptores de bajo nivel. Además, con el objetivo de garantizar la escalabilidad y velocidad de las búsquedas, se implementará la arquitectura FAISS.

Con todo esto buscamos allanar el camino hacia una experiencia de búsqueda de imágenes más intuitiva y enriquecedora en colecciones de datos climáticos,

contribuyendo al avance de la tecnología de recuperación de imágenes basada en el contenido.

II. INVESTIGACIÓN PRELIMINAR

A. CBIR

El CBIR es una tecnología diseñada para la búsqueda y recuperación de imágenes a gran escala, basándose en el contenido visual intrínseco de las imágenes en lugar de los metadatos textuales asociados.

Esta aproximación resulta indispensable en dominios donde la catalogación manual o la descripción por etiquetas son subjetivas o insuficientes para capturar la complejidad del contenido, como es el caso de las imágenes meteorológicas y sus patrones sutiles.

La arquitectura de un sistema CBIR se compone de tres etapas principales:

- Extracción de características: las imágenes se codifican en un vector numérico que representa sus propiedades visuales.
- Indexación: los vectores se almacenan en una estructura de datos optimizada para búsquedas de alta velocidad.
- Medida de similitud y recuperación: el vector de la imagen de consulta se compara con los vectores indexados mediante una métrica de distancia, recuperando las imágenes más cercanas.

El principal objetivo del CBIR es reducir el gap semántico, es decir, la brecha entre las características visuales de bajo nivel extraídas por la máquina y el significado semántico de alto nivel percibido por el humano, para mejorar la precisión en la recuperación.

B. Técnicas

La calidad del sistema CBIR depende intrínsecamente de la capacidad del vector de características para representar fielmente el contenido visual. La investigación moderna se ha desplazado hacia el deep learning para abordar esta tarea en imágenes complejas.

Las características visuales de bajo nivel han sido, de forma histórica, la base de los sistemas CBIR, incluyendo el color, la forma, e incluso la textura. Algoritmos como los Patrones Binarios Locales (LBP) o las Matrices de Co-Ocurrencia de Niveles de Gris (GLCM) son esenciales para capturar la granularidad y la uniformidad de los patrones de nubes y frentes atmosféricos.

Para el reconocimiento de patrones visuales complejos, como los presentes en imágenes satelitales y de radar, la técnica de Transfer Learning utilizando Redes Neuronales Convolucionales pre-entrenadas ha demostrado ser superior.

Una CNN se utiliza como un extractor de características genérico. El vector de deep features se obtiene típicamente de una capa de pooling o de la capa fully-connected anterior a la capa de clasificación. Estos embeddings de alta dimensión codifican información semántica y estructural superior, siendo más robustos a las variaciones de iluminación y escala que afectan a las imágenes de clima, lo que mejora significativamente el rendimiento del CBIR.

Para abordar la complejidad de las imágenes meteorológicas, se ha optado por un enfoque híbrido que combina características de bajo nivel con descriptores profundos. La concatenación de estos descriptores permitirá al sistema capturar una gama más amplia de propiedades visuales, desde la textura y el espectro de color, hasta los patrones semánticos complejos.

C. Algoritmos

La búsqueda de similitud se logra mediante una función de distancia aplicada a los vectores de características. La similitud del coseno es particularmente adecuada cuando se usan deep features de CNNs, ya que mide la similitud angular entre los vectores, independientemente de su magnitud.

Para la búsqueda eficiente de datos extensos, se requiere la Búsqueda del Vecino Más Cercano Aproximado (ANN), que optimiza la velocidad a costa de una pequeña pérdida controlada en la precisión.

En nuestro caso, para implementar esta búsqueda se va a utilizar FAISS, una librería de código abierto reconocida por su optimización en CPU y GPU para la indexación y la búsqueda de vectores densos.

FAISS permite la implementación de índices complejos como el Índice de Archivo Invertido (IVF) para la división del espacio de búsqueda en clusters, y la Cuantización de Producto (PQ) que reduce de forma drástica el espacio de memoria requerido por los vectores, permitiendo así una búsqueda más rápida.

III. METODOLOGÍA IMPLEMENTADA

A. Adquisición de datos

Para el desarrollo y la evaluación del sistema CBIR enfocado en el Reconocimiento de Imágenes Meteorológicas, se utilizó un conjunto de datos de dominio público obtenido de la plataforma Kaggle.

El conjunto de datos original estaba compuesto por un total de 6862 imágenes distribuidas en 11 clases que representan diferentes fenómenos atmosféricos, incluyendo: dew, fogsmog, frost, glaze, hail, lightning, rain, rainbow, rime, sandstorm y snow.



Figure 1: Imagen de la clase dew



Figure 2: Imagen de la clase fogsmog



Figure 3: Imagen de la clase frost



Figure 4: Imagen de la clase glaze



Figure 7: Imagen de la clase rain



Figure 5: Imagen de la clase hail



Figure 8: Imagen de la clase rainbow



Figure 6: Imagen de la clase lightning



Figure 9: Imagen de la clase rime



Figure 10: Imagen de la clase sandstorm



Figure 11: Imagen de la clase snow

Con el objetivo de garantizar una base de datos balanceada que optimice el tiempo de indexación, se aplicó un proceso de reducción sobre el conjunto de datos original, manteniendo todas las clases para asegurar la diversidad de patrones climáticos.

Se mantuvieron las 11 clases originales para asegurar la máxima representación de los fenómenos. Sin embargo, para estandarizar el tamaño de los subconjuntos y optimizar los recursos computacionales, se seleccionaron de forma aleatoria un total de 100 imágenes por clase.

El dataset final utilizado consta de un total de 1210 imágenes, 110 por clase. El conjunto fue dividido de manera que se tenían 1100 imágenes en el subconjunto utilizado para el entrenamiento y la construcción del índice FAISS, mientras que las restantes 110 imágenes se reservaron para la evaluación del rendimiento del sistema.

B. Preprocesamiento de las imágenes

El preprocesamiento nos permite estandarizar el conjunto de datos y garantizar que las imágenes sean compatibles con la arquitectura de la Red Neuronal Convolutacional seleccionada para la extracción de características.

Como las imágenes originales pueden tener diferentes resoluciones y proporciones, se aplicaron los siguientes pasos:

- Redimensionamiento: todas las imágenes fueron redimensionadas a una resolución cuadrada de 224x224.
- Conversión al espacio de color: las imágenes se convirtieron al formato RGB, asegurando así que todas tuvieran tres canales de color.

La normalización de valores de píxeles es un paso crucial para el Deep Learning, ya que asegura que la distribución de los datos de entrada sea consistente y evita que los valores de píxeles grandes dominen la función de coste durante el entrenamiento. Por ello, se llevaron a cabo las siguientes transformaciones:

También se llevaron a cabo verificaciones para identificar y eliminar imágenes con artefactos, ruido excesivo, o posibles

duplicados, garantizando la calidad y singularidad de la base de datos.

- Escalado: los valores de los píxeles se escalaron al rango [0, 1].
- Estandarización por media y desviación típica: para una compatibilidad óptima con los pesos pre-entrenados de la CNN, se estandarizaron los píxeles utilizando la media y la desviación estándar del dataset. Esto transforma la distribución de los píxeles para que se centre alrededor de cero, optimizando así el rendimiento del extractor de características pre-entrenado.

También se llevaron a cabo verificaciones para identificar y eliminar imágenes con artefactos, ruido excesivo, o posibles duplicados, garantizando la calidad y singularidad de la base de datos.

C. Extracción de características

La Extracción de Características se trata del componente principal del sistema CBIR, ya que la calidad del vector numérico resultante determina la capacidad del sistema para medir la similitud semántica y visual entre los patrones climáticos.

Para garantizar una evaluación exhaustiva del rendimiento del modelo en el dominio de las imágenes meteorológicas, se implementaron y compararon tres estrategias de extracción de características distintas. Todo esto con el objetivo de determinar qué representación vectorial ofrece un mejor equilibrio entre robustez y capacidad de discriminación para los patrones climáticos complejos.

Cada modelo utilizado combina un vector de características profundas extraído de una CNN con descriptores de bajo nivel relevantes para los patrones climáticos, mediante la concatenación de vectores.

a. Extractor profundo

Como modelo base se utilizó la arquitectura ResNet-50, pre-entrenada con pesos de ImageNet ya que contiene una gran diversidad de patrones visuales, convirtiéndolo en un extractor genérico y robusto. Además, la capa de clasificación fue eliminada. El vector de características se extrajo de la capa de Global Average Pooling, que reduce la dimensionalidad espacial de los mapas de características a un solo vector, lo que resulta en un vector de 2048 dimensiones por imagen.

Dentro de la función `get_cnn_features`, se aplica la función `preprocess_input` de Keras/TensorFlow, que realiza la estandarización de los píxeles requerida por el modelo de CNN entrenado en ImageNet, lo que garantiza la compatibilidad con los pesos pre-entrenados.

b. Descriptores de bajo nivel

Para complementar las deep features y abordar las propiedades visuales, se implementaron dos descriptores de bajo nivel.

En primer lugar, se usó el Histograma de Color, ya que el color es un indicador crítico en las imágenes meteorológicas.

Se definió la función `get_hsv_features` de manera que utilizara el espacio de color HSV en lugar del RGB porque el HSV separa la información de la iluminación de la cromática, ofreciendo una mayor robustez ante variaciones lumínicas que pueden afectar a las imágenes meteorológicas. Esta función calcula un histograma 3D con 8x8x8 bins, que da lugar a un vector de 512 dimensiones y se normaliza antes de ser aplanado.

Por otra parte, se contemplaron los Patrones Binarios Locales porque la textura es fundamental para capturar la granularidad y uniformidad de los patrones de nubes y frentes atmosféricos. La función `get_lbp_features` convierte la imagen a escala de grises y aplica la matriz LBP con un radio de 3 y un total de 24 puntos.

Así mismo, se utiliza el método uniform, que agrupa patrones poco comunes en una única bin, dando lugar a un histograma normalizado que genera un vector de 26 dimensiones.

c. Concatenación final

Se definió la función `generate_matrix` para gestionar la concatenación de los vectores de los extractores seleccionados. El embedding final de cada imagen se genera mediante la concatenación de los vectores de tipo float32.

Se contemplan un total de tres estrategias de concatenación: CNN + HSV; CNN + LBP; y CNN + HSV + LBP.

Además, antes de ser indexado, el vector final se somete a una normalización L2, que asegura que todos los vectores tengan una longitud unitaria. Esto es fundamental para que la similitud del coseno funcione correctamente como métrica de similitud en FAISS.

D. FAISS

El último punto se centra en la indexación eficiente de los tres conjuntos de vectores de características generados mediante la librería FAISS.

FAISS permite la implementación de la Búsqueda del Vecino Más Cercano Aproximado, lo cual es imprescindible para mantener tiempos de consulta bajos en la base de datos de 1100 imágenes.

Con el fin de determinar el mejor balance entre precisión de la recuperación y velocidad de la consulta, se implementaron y compararon dos tipos de índices de FAISS: el Flat Index, y el IVF-PQ.

a. Métrica de distancia y normalización

El proceso de extracción de características finalizó con la normalización L2 de cada vector. Esta normalización garantiza que, al utilizar la distancia euclídea como métrica de similitud en FAISS, se obtengan resultados idénticos a los de la similitud del coseno, que es la métrica preferida de los deep features.

b. Configuración del índice IndexFlatL2

Se implementó este índice como línea base de rendimiento. El índice plano realiza una búsqueda exhaustiva sobre todos los vectores de la base de datos.

Al calcular la distancia a cada elemento, garantiza la máxima precisión en la recuperación de los k vecinos más cercanos.

No obstante, pese a tener un rendimiento óptimo en precisión, su velocidad de búsqueda es lineal respecto al tamaño de la base de datos, motivo por el que no se considera una opción escalable.

El vector de consulta se inyecta directamente en el índice. Este índice calcula la distancia euclídea entre el vector de consulta y cada uno de los 1100 vectores almacenados en la base de datos.

Este cálculo se realiza con la función `index.search`, que devuelve las distancias y los índices de los k vectores más cercanos.

Como se revisa de forma exhaustiva cada vector, se garantiza la recuperación de los verdaderos vecinos más cercanos, lo que establece el rendimiento de precisión máximo que el sistema es capaz de alcanzar.

c. Configuración del índice IVF-PQ

En segundo lugar se probó el Index Inverted File with Product Quantization (IVF-PQ), ya que ofrece un equilibrio entre la velocidad de búsqueda, precisión y reducción del uso de memoria, propiedades cruciales para el manejo de vectores de alta dimensión.

El proceso de configuración se replicó para cada modelo, variando únicamente la dimensión de entrada. Además, esta configuración se realizó en dos fases principales:

Para la cuantificación de producto se aplicó una comprensión de vectores. El vector de D dimensiones se dividió en M segmentos de k bits cada uno. Se seleccionó un número de segmentos M=32 y un número de bits por segmento k=8. Esta configuración comprime de forma significativa el espacio de memoria requerido por los vectores, facilitando la indexación y acelerando los cálculos de distancia durante la búsqueda.

Para la indexación de archivo invertido, se implementó una capa de clustering para la búsqueda aproximada. Se utilizó un número de clusters Nlist=100. La indexación en FAISS requiere una fase de entrenamiento donde el índice aprende los centroides de los Nlist clusters utilizando los datos del dataset, que en nuestro caso contiene 1100 vectores.

Una vez entrenado y poblado el índice con los 1100 vectores de cada modelo, la consulta se realiza especificando el parámetro nprobe, que representa el número de listas invertidas que se explorarán. Para la evaluación se utilizó nprobe=5, por lo que solo se buscan los 5 clusters más cercanos a la consulta, lo que optimiza la velocidad de recuperación.

El índice devuelve la identificación de las N imágenes más cercanas al vector de consulta, que se utilizarán para la fase de evaluación del rendimiento.

E. Interfaz de Usuario

Para facilitar la interacción con el sistema CBIR y permitir una validación cualitativa de los resultados, se implementó una interfaz gráfica de usuario (UI) vía web. Esta herramienta abstrae la complejidad de los cálculos vectoriales subyacentes, ofreciendo una experiencia de búsqueda intuitiva y accesible.

a. Tecnología utilizada

La interfaz fue desarrollada utilizando el framework Streamlit de Python, una librería diseñada para el desarrollo de aplicaciones interactivas de datos. Se seleccionó esta tecnología por su capacidad para prototipar rápidamente aplicaciones de ciencia de datos y su integración nativa con las librerías utilizadas en el sistema, como TensorFlow, Keras, OpenCV y FAISS.

b. Arquitectura y Funcionamiento

La aplicación actúa como una capa de presentación que conecta directamente con los índices generados en la fase de entrenamiento.

Al arrancar, el sistema carga en memoria el modelo ResNet50 pre-entrenado para actuar como extractor de características en tiempo real. En el código su invocación va acompañada de un decorador. Esto se hace porque la biblioteca Streamlit ejecuta el código completo cada vez que cambia algo en la interfaz. Con el decorador, se indica al código que guarde esa sección en memoria RAM, evitando la descarga repetida del modelo.

El usuario dispone de un menú de configuración lateral donde puede seleccionar qué índice FAISS quiere utilizar para realizar la búsqueda, así como el número de imágenes que quiere recuperar, donde puede elegir en un rango de 1 a 10 imágenes.

Al subir una imagen de *query*, el sistema aplica las mismas transformaciones de preprocessamiento

descritas en la sección III.B y extrae su vector de características según la estrategia seleccionada.

El vector resultante se utiliza para consultar el índice FAISS correspondiente. La interfaz muestra la imagen original junto con las K imágenes más similares recuperadas del dataset, indicando la distancia L2 calculada a cada una.

c. Visualización

En la Figure 1 se muestra una captura de pantalla de la aplicación en funcionamiento, donde se observa una consulta exitosa utilizando el descriptor combinado CNN + HSV.



Figure 12: Interfaz visual

Esta implementación permite al usuario aplicar el proyecto a sus propias fotos, además de permitir evaluar el comportamiento del sistema ante imágenes no vistas previamente. Así, se cumple también con el requisito de adaptar una interfaz de usuario para la visualización de resultados.

IV. EVALUACIÓN DEL RENDIMIENTO

A. Configuración de la evaluación

Para evitar el problema de data leakage u overfitting, como se comentó al hablar del dataset se dividió el conjunto de datos en un conjunto de train y uno de test.

El conjunto de entrenamiento se utilizó para extraer los vectores de características y para la construcción y entrenamiento de los índices FAISS.

Por otra parte, el conjunto de prueba se reservó exclusivamente para lanzar las consultas y medir la precisión del sistema.

B. Métrica de rendimiento

La métrica principal seleccionada para medir la efectividad del sistema fue la precisión en K, con un valor de K = 5.

Esta métrica mide la proporción de imágenes correctamente clasificadas, es decir, pertenecientes a la misma clase que la imagen de consulta, dentro de los primeros 5 resultados obtenidos por el sistema.

$$precision@K = \frac{\# \text{ imágenes correctas en el top } K}{K}$$

Equation 1: Precisión en K

C. Comparación de estrategias de extracción de características

Se compararon tres estrategias de extracción de características con el objetivo de determinar qué representación vectorial ofrecía el mejor equilibrio entre robustez y capacidad de discriminación para los patrones climáticos complejos.

Cada estrategia consistió en un enfoque híbrido que combinaba un vector de características profundas extraído de una Red Neuronal Convolutacional pre-entrenada, con descriptores de bajo nivel relevantes para el análisis del clima, todo esto mediante la concatenación de vectores.

a. Forma + color, CNN + HSV

Esta estrategia combina las características semánticas y de forma de la CNN con información de color de bajo nivel.

Su importancia recae en la clasificación de aquellas clases donde el color es el principal discriminador semántico, ya que permite diferenciar fenómenos visuales como arcoíris o tormentas de arena de fenómenos de color neutro como pueden ser la nieve o la lluvia.

b. Forma + textura, CNN + LBP

En este caso se combinan las Deep Features de ResNet-50 con los Patrones Binarios Locales.

Con su uso se refuerza la distinción de clases con estructuras visuales complejas, pero colores neutros.

Sobre todo, permite diferenciar entre imágenes de lluvia o nieve, que se caracterizan por tener texturas superficiales y patrones sutiles, de otros fenómenos contemplados.

c. Forma + color + textura, CNN + HSV + LBP

Por último, se contempló un vector híbrido completo que combina las características de la CNN con los dos descriptores de bajo nivel implementados.

Esta combinación permite conseguir una representación semántica mucho más rica al capturar la máxima información visual.

Además, permite evaluar si la combinación de todos los descriptores de bajo nivel aporta un beneficio que supere a las combinaciones por separado.

D. Análisis de resultados

La siguiente tabla resume los resultados obtenidos para cada una de las 11 clases del conjunto de prueba y la precisión total de las tres estrategias de extracción de características implementadas.

TABLE I. PRECISIÓN @ K

Clase	Precisión por Método		
	CNN + HSV	CNN + LBP	CNN + HSV + LBP
Dew	0.54	0.54	0.54

Clase	Precisión por Método		
	CNN + HSV	CNN + LBP	CNN + HSV + LBP
Fogsmog	0.76	0.76	0.76
Frost	0.58	0.58	0.58
Glaze	0.58	0.58	0.58
Hail	0.76	0.76	0.76
Lightning	0.98	0.98	0.98
Rain	0.70	0.70	0.70
Rainbow	0.92	0.92	0.92
Rime	0.78	0.8	0.78
Sandstorm	0.88	0.88	0.88
Snow	0.74	0.74	0.74
PRECISIÓN TOTAL	74.74%	74.93%	74.74%

El rendimiento general de las tres estrategias de extracción de características fue similar. La combinación de CNN + LBP obtuvo la precisión total más alta con un 74.93% de precisión, mientras que las otras dos combinaciones obtuvieron una precisión del 74.74%.

Esto sugiere que las Deep Features extraídas de ResNet-50 son el componente dominante en la capacidad predictiva del sistema. Además, aunque la diferencia sea mínima, la ligera ventaja de CNN + LBP sugiere que la información de textura es algo más efectiva que la adición de color o la combinación completa para la distinción de patrones.

Si hacemos un análisis detallado por clase, los resultados obtenidos por cada clase revelan una clara distinción entre los fenómenos meteorológicos más sencillos y aquellos que presentan patrones mucho más similares a los de otras clases.

Como era de esperar, las clases con mejor clasificación son aquellas que poseen características visuales muy distintivas o únicas, lo que facilita su separación semántica.

Por ejemplo, en las clases lightning, rainbow y sandstorm se consigue superar el 80% de precisión.

En el caso de lightning, los rayos, caracterizados por ser líneas brillantes y ramificadas sobre un fondo oscuro, representan un patrón estructural tan distintivo que el vector de características lo codifica de manera aislada. Esto resulta en una correcta clasificación porque no hay otros fenómenos en el dataset que comparten esta estructura geométrica o contraste lumínico.



Figure 13: Imagen de la clase lightning

En el caso de rainbow, la precisión está asociada a la discriminación cromática de las imágenes de esta clase. El arcoíris se define por sus colores espectrales, por lo que al contemplar el descriptor HSV se consigue separar la cromaticidad de la iluminación, permitiendo al sistema identificar de forma consistente el patrón de color circular de esta clase.

La alta precisión en rainbow resalta la importancia de añadir información de color, crítica para diferenciar estados climáticos con tonalidades específicas.

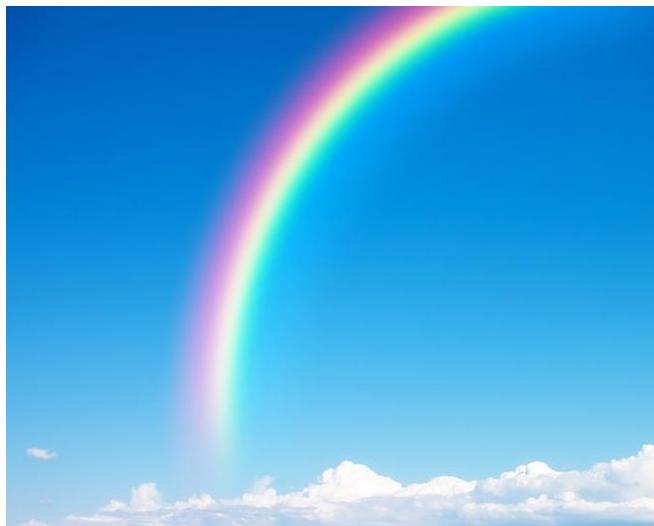


Figure 14: Imagen de la clase rainbow

El fenómeno de sandstorm se caracteriza por una textura granulada y un dominio de tonos ocre y rojizos. La combinación de la forma de la nube de polvo por la CNN y la información de textura extraída con LBP permitió una alta distinción de esta clase.



Figure 15: Imagen de la clase sandstorm

Por otra parte, las clases que demostraron tener un peor rendimiento fueron aquellas con alta similitud visual y baja varianza cromática, lo que sin duda alguna representa el mayor desafío para el sistema. Entre estas clases, podemos encontrar dew, frost o glaze, todas con valores de precisión inferiores al 60%.

El motivo de estos pobres resultados es la alta similitud intrínseca que hay entre imágenes de estas clases, ya que todas representan fenómenos con pocos colores en las imágenes y estructuras muy parecidas.

Para obtener algo más de contexto y poder comprender mejor los errores, se analizó la matriz de confusión del modelo que obtuvo el mejor desempeño global.

Matriz de Confusión: CNN_LBP (Filas=Real, Columnas=Predicción)											
Clase Real	dew	fogsmog	frost	glaze	hail	lightning	rain	rainbow	rime	sandstorm	snow
	dew	fogsmog	frost	glaze	hail	lightning	rain	rainbow	rime	sandstorm	snow
dew	6	0	1	0	3	0	0	0	0	0	0
fogsmog	0	9	0	0	0	1	0	0	0	0	0
frost	0	0	7	2	0	0	0	0	1	0	1
glaze	0	0	1	6	0	0	0	0	3	0	0
hail	0	0	0	0	8	0	1	0	1	0	0
lightning	0	0	0	0	0	10	0	0	0	0	0
rain	0	1	0	0	0	0	8	0	0	0	1
rainbow	0	0	0	0	0	0	0	10	0	0	0
sandstorm	0	0	0	0	0	0	0	0	9	0	1
snow	0	0	0	0	0	0	0	0	0	10	0

Figure 16: Matriz de confusión de la estrategia CNN + LBP

La matriz de confusión revela que se clasificaron de forma errónea tres imágenes de la clase dew como imágenes de la clase hail. Este error se atribuye principalmente a la similitud visual esturctural a nivel local entre ambos fenómenos.

La principal causa de la confusión es la similitud de las formas geométricas presentes en las imágenes. Las imágenes de la clase dew contienen gotas de agua circulares, con formas morfológicamente muy similares a los círculos vistos en las representaciones visuales del granizo de la clase hail. A esto hay que sumarle que la CNN utilizada extrae rasgos profundos y estructurales, por lo que al captar la presencia constante de estas estructuras circulares, el vector de características estructurales de la CNN tiende a solaparse entre ambas clases.

De la misma forma, ambos fenómenos se tratan de fenómenos ed color neutro. Esta baja varianza cromática significa que los descriptores de color no ofrecen suficiente información discriminativa para corregir el error estructural capturado por la CNN, haciendo que se priorice la similitud de la forma circular que deriva en una clasificación errónea.



Figure 17: Imagen de la clase dew



Figure 18: Imagen de la clase hail

Así mismo, tres ejemplares de glaze se clasificaron como rime. Esta es probablemente la confusión más esperable, ya que ambos son fenómenos meteorológicos que representan estructuras de hielo muy similares, resultando en una alta similitud visual. La dificultad de distinción se debe a que comparten características visuales clave.

En primer lugar, hay una baja varianza cromática ya que ambas clases se caracterizan por presentar tonalidades blancas o muy claras de forma predominante. Como consecuencia, sus histogramas de color son casi idénticos, limitando la capacidad del sistema para diferenciarlas utilizando el descriptor de color.

Así mismo, ambos fenómenos suelen resultar en imágenes visualmente planas con texturas sutiles, lo que reduce la capacidad de LBP para capturar diferencias significativas en la granularidad o uniformidad de la superficie.



Figure 19: Imagen de la clase glaze



Figure 20: Imagen de la clase rime

La dificultad del sistema para clasificar correctamente los fenómenos se explica por un fallo del modelo para superar el gap semántico que surge de la alta similitud visual intrínseca en algunas de las clases.

La principal causa de la clasificación errónea es que las clases peor clasificadas comparten características visuales clave, como baja varianza cromática o baja frecuencia de textura. En esencia, la superposición de características de bajo nivel implica que, a pesar de usar un vector de Deep Features robusto, el sistema percibe una gran similitud en el espacio vectorial, dificultando la distinción precisa.

La existencia de estas confusiones se debe también a una decisión tomada durante la adquisición de datos. En un inicio, se contempló la posibilidad de eliminar algunas de estas clases por su gran similitud visual. No obstante, se terminó optando por mantener las 11 clases originales para asegurar la máxima representación de los fenómenos y la diversidad de patrones climáticos. Esto permitió exponer las limitaciones inherentes al modelo y al enfoque CBIR en

escenarios donde la diferencia semántica humana se traduce en una similitud visual a nivel de máquina.

V. CONCLUSIONES

El desarrollo e implementación del sistema de Recuperación de Imágenes Basado en Contenido (CBIR), ha proporcionado conclusiones fundamentales sobre la viabilidad y los desafíos de aplicar técnicas avanzadas de búsqueda visual en dominios complejos como la meteorología.

El proyecto ha demostrado con éxito la viabilidad de la búsqueda y recuperación de imágenes sin depender de metadatos textuales o etiquetas manuales en colecciones de datos visuales climáticos.

La adición de información de color utilizando el espacio HSV resultó ser crítica para diferenciar estados climáticos con tonalidades específicas. Esto fue evidente en el alto rendimiento de clases como rainbow, donde las características cromáticas son determinantes.

La implementación de la indexación mediante la librería FAISS permitió alcanzar tiempos de búsqueda del orden de los milisegundos. Esto valida la escalabilidad operativa del sistema para manejar de forma eficiente datasets más grandes al utilizado en la evaluación.

Además, el sistema logró mantener una precisión total aceptable en la mayoría de las clases analizadas.

El principal desafío del sistema radica en escenarios con baja varianza visual. En estos casos, la alta similitud visual a nivel de forma, textura y color, hace que los vectores de características se solapen y por tanto no se consiga hacer una distinción tan precisa.

El análisis de la matriz de confusión mostró que el modelo tiene dificultades para diferenciar entre fenómenos físicamente similares y fenómenos que comparten estructuras locales, lo que indica que el gap semántico no se ha cerrado por completo en esos escenarios.

De cara a mejorar el rendimiento y las limitaciones identificadas, sobre todo en las clases de baja varianza, se proponen las siguientes líneas de trabajo:

- Especialización del extractor de características: se pueden re-entrenar las últimas capas de la CNN, ya que esta técnica de fine-tuning podría permitir

obtener embeddings más especializados y con mayor poder discriminativo para los patrones meteorológicos, superando la naturaleza genérica de los pesos de ImageNet.

- Ponderación dinámica de vectores: asignar diferentes pesos a los vectores de color y textura antes de la concatenación permitiría priorizar la información más relevante según el contexto de la búsqueda.
- Refinamiento de búsqueda: permitir al usuario seleccionar varias imágenes de entrada daría pie a la posibilidad de que el sistema promedio los embeddings o aplique técnicas de realimentación de relevancia para refinar la búsqueda.

REFERENCES

- [1] A. Di Salvo and M. Valdivielso, "CBIR Weather Project," GitHub Repository, 2025. [Online]. Available: https://github.com/AlbertoDiSalvo/CBIR_Weather_Project
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). **Deep Residual Learning for Image Recognition**. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- [3] Johnson, J., Douze, M., & Jégou, H. (2019). **Billion-scale similarity search with GPUs**. *IEEE Transactions on Big Data*, 7(3), 535-547.
- [4] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971-987.
- [5] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). **Content-based image retrieval at the end of the early years**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1349-1380.
- [6] Gonzalez, R. C., & Woods, R. E. (2018). **Digital Image Processing** (4th Edition). Pearson.
- [7] Abadi, M., et al. (2015). **TensorFlow: Large-scale machine learning on heterogeneous systems**. Software available from tensorflow.org.
- [8] Meta Research. (2024). **Faiss: A library for efficient similarity search and clustering of dense vectors**. Disponible en: <https://github.com/facebookresearch/faiss>
- [9] Bradski, G. (2000). **The OpenCV Library**. *Dr. Dobb's Journal of Software Tools*.
- [10] Van der Walt, S., et al. (2014). **scikit-image: image processing in Python**. *PeerJ*, 2, e453.
- [11] Streamlit Inc. (2024). **Streamlit: The fastest way to build and share data apps**. Disponible en: <https://streamlit.io/>
- [12] Kaggle. (2024). **Weather Image Recognition Dataset**. Disponible en: <https://www.kaggle.com/datasets/jehanhathena/weather-dataset>