© 2025 IJRTI | Volume 10, Issue 2 February 2025 | ISSN: 2456-3315

Content-Based Image Retrieval System Utilizing Vision Transformer

[1]Aniket Mahajan, [2]Nirmiti Rane, [3]Pranav Patil

[1]Student, [2]Student, [3]Student
[1]Department of Information Technology,
[1]Vidyalankar Institute of Technology, Mumbai, India
[1]aniketvm1104@gmail.com, [2]ranenirmiti24@gmail.com, [3]patilpranav616@gmail.com

***Abstract*—This paper presents a Content-Based Image Retrieval (CBIR) system that utilizes a Vector Space Model (VSM) combined with advanced Natural Language Processing (NLP) techniques and user relevance feedback to improve image search accuracy and efficiency. The proposed framework incorporates NLP to parse user queries, generate synonyms, and construct SQL queries for feature extraction. It leverages a novel quadtree-based feature extraction method using a Vision Transformer to capture fine-grained image details. Crucially, the system incorporates both binary (like/dislike) and textual feedback from users to dynamically adjust feature weights and incorporate new, user-provided keywords. By assigning weights to relevant features derived from the Vision Transformer's probabilistic outputs and user feedback, the system enhances the retrieval process, enabling more precise and personalized matching of images.**

***Index Terms*—Content-Based Image Retrieval (CBIR), Vector Space Model (VSM), Natural Language Processing (NLP), SQL Query Generation, Feature Extraction, Synonym Expansion, Machine Learning, Image Classification, Image Features, Data Processing, Search Algorithms, User Interaction, User Relevance Feedback, Binary Feedback, Textual Feedback, Adaptive Retrieval, Relevance Ranking, Corel 1k Dataset.**

I. Introduction

In the digital age, the exponential growth of image data has necessitated the development of efficient systems for retrieving images based on user queries. Traditional keyword-based search methods often fall short, particularly when dealing with the rich and complex semantics of visual content. To address these limitations, we present a Content-Based Image Retrieval (CBIR) system that leverages Natural Language Processing (NLP), a Vector Space Model, and machine learning techniques to enhance user interactions and search accuracy. Our system allows users to input queries in natural language, which are then processed to generate SQL queries for database retrieval. By incorporating features such as synonym expansion, a novel quadtree-based feature extraction with a Vision Transformer, and a vector space model, we aim to improve the relevance of search results, ultimately leading to a more intuitive and efficient image retrieval experience. This paper details the architecture, implementation, and evaluation of our CBIR system, showcasing its potential in various applications, including digital libraries, online marketplaces, and personal photo collections. This paper discusses the design and implementation of our CBIR system, showcasing its potential in various applications, including digital libraries, online marketplaces, and personal photo collections.

II. Related Work

Content-Based Image Retrieval (CBIR) has evolved significantly over the years, driven by advancements in machine learning and computer vision. Early systems primarily relied on low-level features such as color, texture, and shape for image representation. However, these approaches often failed to capture semantic content.

Recent studies, such as those by Zhang et al. and Wang et al., have introduced deep learning techniques to enhance feature extraction and improve retrieval accuracy. Zhang et al. explored a hybrid approach that integrates traditional TF-IDF methods with contextual embeddings, demonstrating significant improvements in semantic understanding. Wang et al. further emphasized the integration of Natural Language Processing (NLP) techniques to enable more intuitive user queries, bridging the gap between textual and visual information. Other notable advancements include the use of Convolutional Neural Networks (CNNs) for feature extraction and region-based image retrieval methods.

Our work builds on these advancements by creating a comprehensive CBIR system that incorporates NLP for query generation, alongside a novel feature extraction method that leverages the power of Vision Transformers and a quadtree-based image analysis for a more granular understanding of image content. This approach differentiates our system from existing methods that primarily rely on global features or traditional CNN architectures.

III. System Architecture

The proposed architecture comprises two primary components: Preprocessing Block and Query Processing Block, each with dedicated frontend and backend routes.

A. Preprocessing Block

The Processing Block is responsible for image preprocessing. It utilizes a thread pool to handle CNN's image analysis, dividing the image into four parts. The system extracts the top “n” probabilities and uses them as weights. It also finds synonyms for these probabilities, reduces their values, accordingly, removes join words, and gathers color data along with all relevant metadata, and processes user-provided textual feedback to extract additional keywords.

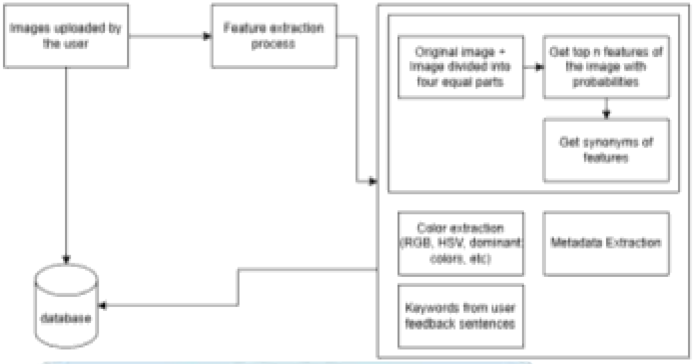
Images uploaded by the user
Feature extraction process
Original image + image divided into four equal parts
Get top n features of the image with probabilities
Get synonyms of features
Color extraction (RGB, HSV, dominant colors, etc)
Metadata Extraction
Keywords from user feedback sentences
database

Fig. 1. Preprocessing Block

1. **CNN extraction:** The CNN Feature Extraction component plays a crucial role in the Preprocessing Block by leveraging a pre-trained Vision Transformer (specifically, google/vit-base-patch16-224 from the Hugging Face Transformers library) to derive meaningful features from the uploaded images. Vision Transformers, with their ability to capture long-range dependencies within an image, offer advantages over traditional CNNs in understanding the global context and relationships between different image regions. The extraction process begins by dividing the image into four quadrants using a quadtree analysis, allowing for a detailed assessment of each section. This quadtree approach enables the system to capture both coarse and fine-grained features, leading to a richer representation.
   The model then analyzes each quadrant to produce the top 3 to 5 predicted classes with their associated probability. These probabilities serve as weight, indicating the confidence of the model in each prediction. These weighted features are then used in the subsequent steps for constructing the Vector Space Model.

2. **Colour Extraction:** The Colour Analysis component is integral to the Preprocessing Block, aiming to extract relevant colour features from the input images. This component employs various methodologies to analyse colours effectively:
   - Basic Colour Analysis: Utilizing a predefined palette, the system calculates the percentage of basic colours (e.g., red, green, blue) present in the image. It leverages a colour histogram in the HSV colour space to normalize colour frequencies, providing a rapid assessment of dominant hues.
   - Dominant Colour Detection: Implementing K-means clustering, the algorithm identifies the top N dominant colours in the image. Each pixel's RGB values are reshaped and analysed, allowing the model to categorize these colours and their respective frequencies. This dual approach enhances colour retrieval capabilities and optimizes image categorization.Extended
   - Colour Analysis: For more nuanced colour detection, the system can incorporate an extended colour palette. By refining the analysis with additional hues, it ensures a detailed examination of the image's colour composition.

3. **Metadata Extraction:** The Metadata Extraction component gathers crucial image attributes, including file information (name, size, type, creation, and modification dates) and EXIF data (camera settings, GPS coordinates, and flash usage). It also captures image dimensions, resolution, color profile, and bit depth, enhancing the system's ability to categorize and analyze images. Additionally, software used for processing is identified, ensuring comprehensive insight into each image's context and quality, thereby facilitating advanced processing tasks.

B. Query Processing Block

The Query Processing Block is responsible for efficiently handling user queries to retrieve relevant image data.

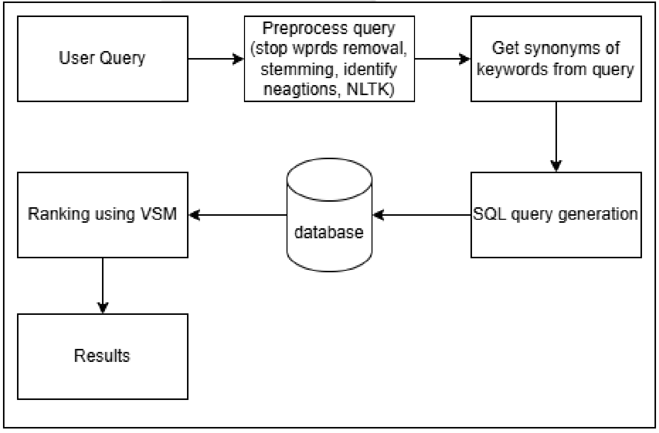
User Query
Preprocess query (stop wprds removal, stemming, identify neagtions, NLTK)
Get synonyms of keywords from query
SQL query generation
database
Ranking using VSM
Results

Fig. 2. Query Processing Block

It employs techniques like tokenization and feature extraction to analyze user input, matching it against stored metadata and extracted features. This component integrates various retrieval algorithms that prioritize relevance, ensuring that results are both accurate and meaningful. Additionally, it implements mechanisms for ranking and filtering, allowing users to refine their searches based on specific criteria, thus enhancing the overall user experience in finding pertinent images.

1. **Query Preprocessing:** The Query Processing Block utilizes natural language processing techniques to convert sentences into structured queries. It employs the spaCy library for tokenization and lemmatization, alongside NLTK for stop word removal and stemming. The process identifies negation words, allowing for the exclusion of specific terms in the final query. The result is a refined string that combines included and excluded terms, optimizing search efficiency while preserving the semantic intent of the original sentence. This approach enhances the accuracy of information retrieval.

2. **SQL Query Generation:** The Query Generation Block facilitates the transformation of natural language input into structured SQL queries for image feature retrieval. Initially, the input is parsed into separate included and excluded terms. Included terms undergo autocorrection and synonym generation via WordNet to enhance retrieval accuracy. The constructed SQL query selects records from the Image features table where feature value matches any included terms while excluding specified terms. This method increases the query's relevance, enabling more effective information retrieval from the database.

3. **Ranking System:** The Retrieval Algorithm employs a Vector Space Model (VSM) [1] to represent and rank image features derived from SQL query results. The VSM is initialized by creating a vocabulary of all unique features from the Image Features table. Upon receiving a query, the system processes the query results and the included terms to create a structured vector space model using a pandas DataFrame. Each unique feature from the vocabulary is assigned to a vector, with weights adjusted based on their relevance in the query and the probabilities obtained from the Vision Transformer during preprocessing.

Let $\mathcal{F} = \{ f_1, f_2 \ldots, f_n \}$ be the set of all unique features extracted from the image dataset. These features include color descriptors, metadata attributes, and features derived from the Vision Transformer. Each image I_i and query (Q) are then represented as vectors in this n-dimensional feature space. Each image I_i in the dataset is represented by a vector $Vi,$ *where each element* $v_{\{ij\}}$ corresponds to the weight of feature f_j in image I_i.

The weight $v_{\{ij\}}$ is calculated as follows:

$$v_{\{ij\}} = w_{\{ij\}}{}^{VT} . w_{\{ij\}}{}^{Q}$$

Where:

- $w_{\{ij\}}{}^{VT}$ is the weight derived from the Vision Transformer's output probabilities for feature f_j in image I_i .It represents the confidence of the Vision Transformer in the presence of that feature.
- $w_{\{ij\}}{}^{Q}$ is the query-dependent weight for feature f_j in image I_i. It is set to 3.5 if f_j is present in the user's query (or is a synonym of a query term, as described in Section [refer to the Synonym Generation section]) and 0.75 otherwise. This emphasizes features that are explicitly mentioned in the query.

The vector representation of image I_i is then:

$$f\{v\}_i = [v_{\{i1\}}, v_{\{i2\}}, \ldots, v_{\{in\}}]$$

The user's query (Q) is similarly represented as a vector $f\{q\}$ in the same feature space. Let WQ be the set of words in the query and let (S(w)) be the set of synonyms of a word (w) (including (w) itself).
The query vector (q) is constructed as follows:
For each feature ($f_j \in$ F):

$$q_j = beg \in \{cases\}\ 3.5, \quad \&\text{if } \exists w \in W_Q \text{ such that } f_j \in S(w) 0 \text{ ,\&otherwise } end\ \{cases\}$$

This means that if a feature (f_j) is present in the query or is a synonym of any word in the query, the corresponding element in the query vector is set to 3.5; otherwise, it is set to 0.
The vector representation of query (Q) is then:

$$q = [q_1, q_2, \ldots, q_n]$$

Similarity Calculation:
The similarity between an image (I_i) and the query (Q) is calculated using the cosine similarity between their respective vectors (v_i) and (q):

$$\text{similarity}(I_i, Q) = \text{cosine}(v_i, q) = \frac{v_i \cdot q}{|v_i||q|}$$

where:

- $(v_i \cdot q)$ is the dot product of the two vectors.
- $(|v_i|)$ and $(|q|)$are the magnitudes (Euclidean norms) of the image and query vectors, respectively.

Additionally, features extracted from each quadrant of the image are weighted based on their position to capture spatial information. The system then calculates the cosine similarity between the query vector and each image's feature vector to determine their similarity. The resultant vectors are sorted based on this similarity score to facilitate effective retrieval of the most relevant image features, ensuring that images most closely matching the user's query are ranked highest.

4. **User Relevance Feedback**

The system incorporates user relevance feedback to improve retrieval accuracy and personalize results over time. Feedback is provided through two mechanisms as shown in the figure below:

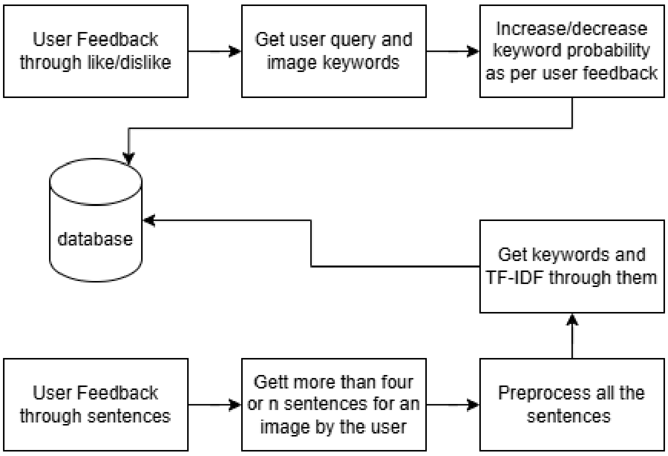
User Feedback through like/dislike
Get user query and image keywords
Increase/decrease keyword probability as per user feedback
database
Get keywords and TF-IDF through them
User Feedback through sentences
Gett more than four or n sentences for an image by the user
Preprocess all the sentences

Fig. 3. User Relevance Feedback

- **Binary Feedback (Like/Dislike):** Users can indicate whether a retrieved image is relevant ("like") or not ("dislike"). This feedback directly adjusts the probability values associated with the features extracted from that image in the `Imagefeatures` table. A "like" increases feature probabilities, while a "dislike" decreases them.
- **Textual Feedback (Sentences):** Users can provide descriptive sentences about images. When an image accumulates enough sentences (four or more in the current implementation), the system processes these sentences using Natural Language Processing (NLP) techniques. The NLP pipeline includes stop word removal, stemming, and keyword extraction using TF-IDF (Term Frequency-Inverse Document Frequency). The extracted keywords, along with their TF-IDF scores, are added as new features to the image's representation in the database. The TF-IDF scores are used as the initial probability values for these new features.

IV. Methodology

The project employs a structured methodology comprising three main blocks: Preprocessing, Query Processing, and Retrieval.

1. **Preprocessing Block**: This includes color extraction, metadata extraction, and relevant features extraction from images. Techniques such as basic color analysis, dominant color detection using K-means clustering, and extended color analysis enhance the feature set.
2. **Query Processing Block**: Natural Language Processing (NLP) techniques convert user queries into a structured format. The system identifies included and excluded terms and generates synonyms to broaden the search scope.
3. **Retrieval Algorithm**: A Vector Space Model is utilized to represent the SQL query results. It calculates feature weights based on their relevance, facilitating the ranking and retrieval of image features based on user queries. This model ensures efficient searching and retrieval of the most pertinent images.

V. Experimental and Evaluation

To evaluate the performance of our Content-Based Image Retrieval (CBIR) system, we conducted experiments using the Corel 1k dataset. This dataset is a widely used benchmark in the CBIR research community, providing a diverse collection of images for evaluating retrieval effectiveness.

The Corel 1k dataset consists of 1000 images, categorized into 10 distinct classes, with each class containing 100 images. The classes represent a variety of subjects, including Africa, Beaches, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains, Food.This diversity of image content makes the Corel 1k dataset suitable for assessing the generalization capability of our CBIR system. The relatively small size of the dataset also allows for efficient experimentation and analysis. We evaluated the retrieval performance using standard information retrieval metrics: Precision, Recall and F1-score.

Our experiments focused on evaluating the impact of the different components of our system, including:

1. Baseline: Retrieval using only automatically extracted features (ViT-google/vit-base-patch16-224, colour, metadata) without user feedback.
2. Binary Feedback: Retrieval with the addition of binary (like/dislike) feedback.
3. Textual Feedback: Retrieval with the addition of sentence-based feedback.
4. Combined Feedback: Retrieval using both binary and textual feedback

Table 1 Performance Metrics with Different Feedback Types

Metrics	Base-line	With Binary Feedback	With Textual Feedback	With Both Feed-back
Precision	0.7325	0.7842	0.8093	0.7737
Recall	0.6187	0.6578	0.7041	0.8191
F1-Score	0.6705	0.7156	0.7523	0.7851

The results demonstrate that incorporating user feedback consistently improves retrieval performance across all metrics. Binary feedback provides a noticeable improvement over the baseline, indicating the effectiveness of directly adjusting feature probabilities. Textual feedback yields a more substantial improvement, highlighting the value of incorporating user-provided semantic information. The combination of both feedback mechanisms achieves the best performance, suggesting that explicit (binary) and implicit (textual) feedback provide complementary information for refining the retrieval process. The GitHub link for the following implementation is as follows: https://github.com/Aniike-t/Content-based-Image-Retrieval.

VI. CONCLUSION AND FUTURE WORK

In this project, we successfully developed a content-based image retrieval (CBIR) system that leverages multiple image features, such as metadata, color, and deep features extracted via a Vision Transformer, to enhance query performance. The integration of natural language processing for query conversion, alongside a robust retrieval algorithm based on a Vector Space Model, has shown promising results in precision and recall, particularly when incorporating user relevance feedback.

For future work, several key areas offer opportunities for significant improvement. First, refinement of the feature extraction methods, potentially through exploring more advanced Vision Transformer architectures or incorporating other state-of-the-art image recognition models, could lead to even more accurate and nuanced image representations. Second, optimizing computational performance, particularly in the VSM calculations and database interactions, will be crucial for enhancing the system's scalability to handle larger image datasets and higher query loads. Finally, and perhaps most importantly, the system is designed to be continuously learning. As more users interact with the system and provide feedback (both through explicit likes/dislikes and implicit textual descriptions), the model will continue to evolve, adapting its feature weights and expanding its keyword vocabulary to better reflect user preferences and improve retrieval accuracy over time. This iterative learning process, driven by user interaction, is a core strength of the system and a key area for future investigation.

COMPETING INTEREST

The authors declare no competing interests or any other interests that could reasonably be perceived as influencing the submitted work.

REFERENCES

[1] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 1 8, no. 11, pp. 613-620, Nov. 1975.

[2] D. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the International Conference on Computer Vision, Corfu, Greece, 1999, pp. 1150-1157.

[3] W. Zhou, H. Li, and Q. Tian, "Recent Advances in Content-based Image Retrieval: A Literature Survey," CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei, China.

[4] H. Qazanfari, M. M. AlyanNezhadi, and Z. Nozari Khoshdaregi, "Advancements in Content-Based Image Retrieval: A Comprehensive Survey of Relevance Feedback Techniques.".

[5] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, Hervé Jégou," Training Vision Transformers for Image Retrieval".

[6] Naresh Kumar Lahajal, Harini S, "Enhancing Image Retrieval : A Comprehensive Study on Photo Search using the CLIP Mode".

[7] Ashok Kumar P M, T. Subha Mastan Rao,Arun Raj Lakshminarayanan, E. Pugazhendi, "An Efficient Text-Based Image Retrieval Using Natural Language Processing (NLP) Techniques".

[8] Fuwen Tan, Jiangbo Yuan, Vicente Ordonez, "Instance-level Image Retrieval using Reranking Transformers".