Content-Based Image Retrieval Using Transfer Learning and Vector Database

Li Shuo, Lilly Suriani Affendey, Fatimah Sidi
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia

***Abstract*—Content-based image retrieval (CBIR) systems are essential for efficiently searching large image datasets using image features instead of text annotations. Major challenges include extracting effective feature representations to improve accuracy, as well as indexing them to improve the retrieval speed. The use of pre-trained deep learning models to extract features has elicited interest from researchers. In addition, the emergence of open-source vector databases allows efficient vector indexing which significantly increases the speed of similarity search. This paper introduces a novel CBIR system that combines transfer learning with vector databases to improve retrieval speed and accuracy. Using a pre-trained VGG-16 model, we extract high-dimensional feature vectors from images, which are stored and retrieved using the Milvus vector database. Our approach significantly reduces retrieval time, achieving real-time responses while maintaining high precision and recall. Experiments conducted on ImageClef, ImageNet, and Corel-1k datasets demonstrate the system's effectiveness in large-scale image retrieval tasks, outperforming traditional methods in both speed and accuracy.**

***Keywords*—Content-based image retrieval (CBIR); image retrieval; transfer learning; convolutional neural networks; VGG-16; vector database; milvus; feature extraction; high-dimensional vectors; real-time image search**

I. Introduction

In the Internet era, massive amounts of multimedia data (including text, images, audio, video, etc.) are continuously generated and stored. How to effectively retrieve relevant information from a huge image dataset has become an urgent problem to be solved. Traditional text-based image retrieval methods can no longer meet this demand, while content-based image retrieval (CBIR) technology retrieves through image features, significantly improving the accuracy and efficiency of retrieval. Practical applications of CBIR include e-commerce, web search, and medical image analysis.

The core of CBIR technology includes two key steps: image feature extraction (indexing stage) and similarity matching (retrieval stage). Image feature extraction is to convert the original image content into feature vectors, while similarity matching is to compare the feature vectors of the query image with the image in the dataset, calculate their similarity, and identify similar images. Research on CBIR technology has a long history, nevertheless most work mainly focus on model training and evaluation of image classification tasks, measuring the accuracy, and not considering the time consumption for indexing or retrieval.

This paper proposes a new CBIR retrieval system that combines transfer learning technology and vector database. Image features are extracted through convolutional neural networks (CNNs), and high-dimensional vectors are stored in vector databases to achieve efficient image retrieval. We evaluated the performance of the indexing and retrieval stages and verified the feasibility and effectiveness of the scheme. This study aims to explore how to build a high-performance CBIR system to meet the real-time retrieval requirements in large-scale image datasets.

This paper is organized as follows: Section II discusses related research in the field of CBIR and the techniques used in this study; Section III details the methodology adopted in this study; Section IV describes the proposed CBIR system utilizing transfer learning and vector database; Section V presents the experimental results, where we evaluate the performance of our system on various benchmark datasets. In Section VI, we conclude the paper by summarizing the findings, discussing the implications of the proposed system, and suggesting potential directions for future work.

II. Related Work

The theory of image retrieval has been around for a long time. Initially, text annotation was used to store image descriptive text in a database for retrieval. However, this method is manual and subjective as well as inefficient and inaccurate when faced with large amounts of data [1]. Subsequently, researchers began to study CBIR technology, using low-level features such as color, shape, and texture to represent images [2]. This method is automated and efficient, but prone to errors and has low accuracy in complex image recognition.

Modern CBIR technology extracts features from images through deep learning, uses convolutional neural networks to vectorize images, and performs matching through similarity calculations [3]. Currently, commonly used similarity matching algorithms include Euclidean distance, cosine similarity, etc.

Based on these technologies, many CBIR applications have emerged, mainly focusing on the construction, training, and evaluation of models in classification tasks, and evaluating their performance by classifying images from different data sets [4] Thus, to improve the performance of CBIR systems, numerous research on feature extraction, similarity matching, as well as storage architecture have been conducted by different researchers.

Traina et al. [5] proposed the SIFT (Scale-Invariant Feature Transform) machine vision algorithm for feature extraction and used K-means to calculate similarity distances. Simran et al. [6] introduced the use of deep learning techniques for extracting image features. Kumar et al. [7] proposed using DarkNet-19 and

DarkNet-53 models for feature extraction, along with PCA for dimensionality reduction. Sikandar et al. [8] proposed the use of ResNet50 and VGG16 for feature extraction, employing KNN for similarity calculations.

Similarity calculation is also very important. Alsmadi [11] used the memetic algorithm method to calculate similarity. Sikandar et al. [8] used the KNN method to calculate similarity. Niu et al. [12] used the Residual Vector Product Quantization for approximate nearest neighbor method to calculate.

Retrieval performance is an important indicator of CBIR system, which is usually evaluated by accuracy. Some relevant studies have been collected for retrieval performance. Chughtai et al. [13] used transfer learning to call VGG16, VGG19, EfficientNetB0, ResNet50 and other models for CBIR, with a retrieval accuracy of up to 96%. Mohammed et al. [14][14] used two pre-trained deep learning models ResNet50 and VGG16 and a machine learning model KNN implementation to achieve a maximum accuracy of 100%. Gautam et al. [15] used VGG16 and ResNet-50 architectures to obtain a maximum accuracy of 90.18%. Sadiq et al. [16] combined NASNetMobile, DenseNet121 and VGG16 models to achieve a maximum accuracy of 98%. Thanikachalam et al. [17] proposed Tokens-to-Token Vision Transformer (T2T-ViT), a novel CBIR method with an accuracy of up to 99.42%.

The above literature has conducted in-depth research on CBIR, with different retrieval methods and good performance. However, most work do not measure the time consumption for indexing, nor do they report the retrieval time, which is also an important indicator for measuring the performance of CBIR systems.

Mezzoudj et al. [18] on the other hand used a big data solution to retrieve and index CBIR, and reported the time for the indexing and retrieval stages of CBIR using the ImageClef and ImageNet datasets. Their solution greatly improved the efficiency of CBIR. However, they did not report the retrieval accuracy.

Recently, Stata et al. [9][9] and Singla et al. [10] focused on utilizing vector databases for vector extraction and evaluated the performance of these databases.

From the literature, we attempt to implement CBIR that uses transfer learning feature extraction, and utilize a vector database to store the high-dimensional vectors in, so that the vector data volume undertakes the retrieval task and improves the retrieval efficiency.

III. METHODOLOGY

This section describes the experimental design process in detail, including experimental steps, application architecture design, software and hardware configuration, data set description, experimental evaluation indicators, and experimental verification methods.

A. Application Architecture Design

Fig. 1 is the application architecture design diagram. First, the pre-trained VGG model is used to process each image in the dataset, converting these images into high-dimensional vectors. This step is called vectorization, which converts the pixel information of the image into feature vectors to facilitate subsequent similarity calculations. The generated high-dimensional vectors are then stored in the Milvus vector database for efficient storage and retrieval operations. Milvus is a high-performance distributed vector database that can store high-dimensional vectors and quickly retrieve massive amounts of vector data. VGG16 is a classic convolutional neural network (CNN) model that efficiently extracts image features. Transfer learning technology allows the VGG model to be used directly for vectorizing images, reducing the model training process and improving work efficiency.

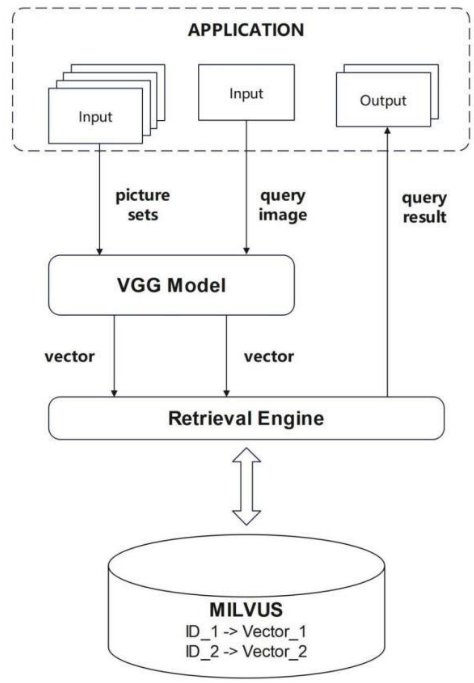

Fig. 1. Application architecture design diagram.

When a user submits an image for retrieval and matching, the input image is processed in the same manner. Specifically, the input image will also be vectorized through the VGG model to generate its corresponding high-dimensional feature vector. This feature vector is then used to query the Milvus database to find the vector that is most similar to it.

The Milvus database calculates vectors that are similar to the query vector and returns a list of results sorted according to the similarity score. The similarity score represents the similarity distance between the query vector and the vectors stored in the Milvus database. The lower the similarity score, the more similar the two images are. Finally, the system retrieves the images corresponding to these vectors, sorts them by similarity, and displays them to the user. In this way, users can easily find other images similar to their input image to meet the needs of image retrieval and matching.

B. Hardware

This experiment uses hardware resources of Intel Quad Core i5-4210U CPU, 1.70 GHz, and 8 GB memory. Table I shows the hardware required for this project.

TABLE I. HARDWARE INFORMATION

Hardware	Specification
Central Processing Unit (CPU)	Intel Core i5 4210 @ 1.7 GHz
Primary Memory (Random Access Memory)	8 GB
Secondary Memory (Hard Disc Drive)	1024 GB Solid State Drive

C. *Software and Frameworks*

Python was chosen as the development language. The reason is that it contains many mature function libraries and frameworks. Table II shows the software and framework required for this project.

TABLE II. SOFTWARE INFORMATION

Software	Version	Purpose
Python	3.6.0	Provide a basic development environment for the project
Tensorflow	1.15.4	An open-source deep learning framework that integrates many algorithms and models to facilitate model training.
Keras	2.3.1	A machine learning framework repackaged based on TensorFlow, an open-source high-level API neural network framework. Simplifies use and facilitates development.
Numpy	1.16.5	Integrated function library that supports a large number of dimensional array and matrix operations.
Matplotlib	3.7.2	A two-dimensional drawing library developed in Python language.
Pillow	7.1.0	Get the image based on the image path and
Flask	2.0.3	Start a service to accept and process request requests
Milvus	1.0.0	Vector database stores high-dimensional vectors after vectorization of images. And used to query and retrieve similar vectors.
Docker	19.03	Install and run Milvus and the CBIR project

D. *ImageClef Dataset*

ImageClef is a series of events, and a different dataset is released each year, so the size of the dataset will vary with the event and year. The ImageCleF dataset contains images and related annotation information, and researchers can conduct research on various image understanding tasks, such as image classification, image annotation, image retrieval, etc. The dataset has a total of 20,000 images, which are numerically classified into folders. This is a photo book, taken from all over the world. These datasets usually cover different subject areas and multiple languages, providing a rich research foundation.

E. *ImageNet Dataset*

The ImageNet dataset is a computer vision dataset. It is a large image dataset established to promote the development of computer image recognition technology. Many well-known models have been trained on this dataset, such as VGG-16, VGG-19, Restnet-50, etc. The images in the ImageNet dataset cover most of the image categories seen in daily life. This is a dataset with more than one million images.

F. *Corel-1k Dataset*

The Corel-1000 dataset is widely used in the CBIR field, as many researchers use this dataset to evaluate the quality of image retrieval tasks. It contains a total of 1000 images in 10 categories, namely: Africa, Beach, Building, Bus, Dinosaur, Elephant, Flower, Food, Horse, and Monument. There are 100 images in each category. This dataset was used to evaluate the quality of retrieval for CBIR applications.

G. *Experimental Evaluation Metrics*

To evaluate the indexing performance, the time consumption for indexing of images of different sizes in the imageClef data were measured.

To evaluate the retrieval performance, the retrieval time for images of different sizes from the imageClef dataset were taken.

In addition, the time consumption of indexing and retrieval using the ImageNet dataset was measured to evaluate the performance in a large dataset.

Finally, the retrieval quality was evaluated using the precision and recall metrics on the Corel-1k dataset to measure the retrieval accuracy.

IV. CBIR USING TRANSFER LEARNING AND VDBMS

In this section, the implementation of the CBIR experiment will be discussed in detail. First, the relevant environment is deployed, the VGG-16 model is built, and the dataset is preprocessed. Subsequently, the execution time of the indexing and retrieval stages with different data amounts is recorded on the imageClef dataset, and the accuracy of the retrieval stage is tested on the Corel-1k dataset. Finally, the experiment is repeated using the ImageNet dataset and the relevant data is recorded.

A. *Data Preprocessing*

Since the images are stored in various folders and are nested, it is necessary to extract the images from various directories and then merge them into the same directory folder. This makes it easier to process the images. At the same time, filter out non-image files. Finally, the dataset is compressed packaged and uploaded to the specified path on the server for subsequent use.

B. *Experimental Environment Setup*

In this step, the operating environment needs to be prepared. All software and the related dependent libraries as shown in Table II are installed. The project code can be found at https://github.com/LI-SHUO-lee/CBIR.git.

C. *Initialize VGG Model*

In this work, transfer learning is utilized to directly employ the VGG model for feature extraction. Initially, the VGG model needs to be defined. The VGG16 model requires the input image size to be 224x224 pixels with three channels. Pre-trained weights on the ImageNet dataset are utilized here. These pre-trained weights facilitate faster convergence and generally enhance the model's performance. Max pooling is employed, which selects the maximum value in each region as the output. The parameter includes top=False indicating that the top layer (fully connected layer) of the model is excluded, as a custom output layer is intended to be added on top. An array of zeros with shape (1, 224, 224, 3) is created using np.zeros((1, 224, 224, 3)) as input, and the model's prediction method is called to make a prediction. This is typically performed to ensure that the model is loaded correctly and can make a normal prediction.

D. CBIR Indexing Phase

The indexing stage refers to the process of directly using existing models and parameters to extract features from images through transfer learning technology and storing these high-dimensional vectors describing image features in the vector database so that subsequent retrieval processes can be carried out effectively.

Fig. 2 shows the process of image indexing. All images are preprocessed to ensure that their suffix is jpg or png, and at the same time unify the size of the images to ensure that they are 224*224*3. Then the features of the image are extracted through the VGG model. In order to reduce the pressure on the vector database Milvus, high-dimensional vectors are temporarily stored in the server cache. After waiting for the image batch vectorization to be completed, the vectors in the server cache are flushed into the vector database Milvus to ensure that the database is only requested once. This ensures the function and improves efficiency.

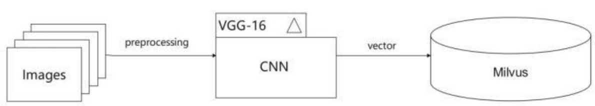

Fig. 2. Indexing process.

Table III shows the time taken for indexing different image sizes using the ImageClef dataset.

TABLE III. THE TIME CONSUMPTION OF INDEXING

Image size	2011	5952	10000	11963	17953	20000
Time consumption	446 s	1066 s	1689 s	1991 s	3089 s	3412 s

E. CBIR Retrieval Phase

Conducting an experimental study on the retrieval stage of the CBIR application is essential, as it is the core component of the project. The retrieval phase's success directly impacts the overall project outcome. Therefore, an in-depth experimental exploration is necessary to address its challenges.

Firstly, we will evaluate retrieval time, a crucial performance metric. Assessing retrieval time helps understand the system's efficiency in handling a large volume of images, reflecting architectural improvements.

Secondly, accuracy evaluation is vital in CBIR. We will compare search results with standard dataset to assess the accuracy.

1) Retrieval time consumption evaluation: Fig. 3 illustrates the image-retrieving process. When the data set is fully vectorized, the database stores the feature vector of each image in the data set. The user only needs to input a query image, and then use the same model and algorithm to vectorize and extract features of the image to obtain a feature vector. Only the feature vector and the number of data pieces needing fuzzy matching need to be communicated to Milvus. Milvus will automatically search from its own library through the vector, and then return the closest Top k vectors and vector IDs. Next, the table is consulted to find the correspondence between the vector ID and the actual image, enabling the retrieval of the fuzzy-matched image to be returned to the user. This enables users to search for images by image. The time consumed in this stage is used as a key indicator for evaluating the CBIR application.

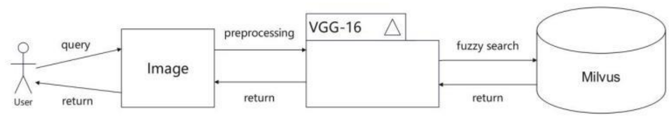

Fig. 3. Image retrieval.

First, the ImageClef dataset is used to perform the retrieval process according to the image size in Table IV. Then the consumed time is recorded respectively to evaluate the performance of the retrieval process of the CBIR application.

TABLE IV. THE TIME CONSUMPTION OF RETRIEVAL

Image size	2011	5952	10000	11963	17953	20000
Time consumption	0.27 s	0.26 s	0.33 s	0.23 s	0.20 s	0.30 s

2) Retrieval accuracy evaluation: Retrieval accuracy is also an important indicator of retrieval performance. Since the vector database is an approximate search, the search results may not be 100% correct. However, accuracy and speed are contradictory indicators. We cannot blindly pursue speed and ignore the accuracy of CBIR. Therefore, at this stage, we will evaluate and record the retrieval accuracy on the Corel-1k dataset, which will also be used to evaluate the performance of CBIR applications. Fig. 4 illustrates the process of evaluating the accuracy.

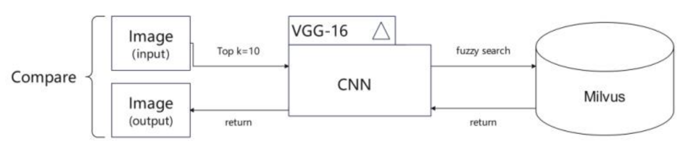

Fig. 4. The accuracy process.

In this experiment, precision and recall will be used to evaluate the retrieval quality. The Corel-1k dataset will be used because many papers use this dataset to calculate precision and recall for evaluating CBIR applications. There are 10 categories in Corel-1k, each with 100 images, for a total of 1,000 images. One image is randomly selected from each category for query testing, and the remaining 99 are used for training. Therefore, 990 images need to be vectorized, converted into vectors through the VGG-16 model, and stored in Milvus. The Euclidean distance (L2) is used to calculate the distance between vectors. Setting Top k = 10, which means that 10 images will be returned for each query. For the returned images, we extract the classification label of the image and then compare it with the classification label of the query image. If the classification label is consistent, the retrieval is considered correct, and the application has successfully detected the image of this class. Otherwise, it is an error. Record the number of correctly classified images and compare them with the returned images and all images of this class in the database to calculate the precision and recall. Finally, the process is repeated for each image in each category, and the precision and recall rate of each category are calculated to verify the retrieval performance of the application.

Precision formula:

$$\text{Precision} = \frac{\text{number of similar images retrieved}}{\text{total number of images retrieved}} \tag{1}$$

Recall formula:

$$\text{Recall} = \frac{\text{number of similar images retrieved}}{\text{total number of similar images in the database}} \tag{2}$$

Table V shows the calculation results of precision and recall through the experiment.

TABLE V. THE ACCURACY FOR DIFFERENT IMAGE CATEGORIES

	Precision	Recall
africans	0.869	0.711
beaches	0.847	0.592
buildings	0.898	0.666
buses	1.000	1.000
dinosaurs	1.000	0.956
elephants	1.000	0.853
flowers	0.994	0.804
foods	0.983	0.805
horses	1.000	0.828
monuments	0.935	0.716

F. *Time Consumption in the ImageNet Dataset*

To verify time consumption on the ImageNet dataset, we will use the same method. Specifically, we will record the time taken for both the indexing and retrieval phases. Repeating the experiment on different datasets will help verify the robustness and wide applicability of the proposed method. Table VI shows the experiment results.

TABLE VI. THE RESEARCH RESULTS IN THE IMAGENET DATASET

ImageNet size	Time for indexing	Time for retrieval
1,461,406	249,400 s	0.5 s

V. RESULTS AND DISCUSSION

A. *Introduction*

This section will provide a detailed analysis of the project experiment data. We will collect, organize, visualize, and statistically analyze the experimental data. Using the reference paper [18] as a benchmark, we will compare and discuss various aspects such as architecture design, algorithm comparison, development language, index performance, and retrieval performance to demonstrate the superiority of our architecture.

B. *Architecture Comparison*

The architecture has the following advantages over Mezzoudj [18].

First, the VCBIR architecture is simpler, requiring no big data platform; it can be deployed and maintained on a single server, facilitating horizontal scaling.

Second, image vectorization uses the VGG-16 model, which provides more accurate feature extraction while maintaining high efficiency. If higher accuracy or a lighter model is needed, the vector model can be easily replaced without altering other components.

Finally, vector storage uses Milvus instead of Hadoop, eliminating the need for big data components like Spark for vector calculations, thus simplifying the architecture. Milvus supports horizontal scaling, allowing for upgrades to a distributed cluster in case of storage bottlenecks, without impacting upper-layer applications. Table VII summarizes the architectural differences.

TABLE VII. ARCHITECTURE COMPARISON TABLE

	Architecture from reference paper [18]	VCBIR Architecture
Vector model	CS-LBP extracts image features based on color, texture, etc., but it has low accuracy.	VGG extracts image features based on neural networks and has more layers with high accuracy.
Operating environment	Rely on the big data platform.	The ordinary operating system is sufficient.
Resource consumption	High resource consumption.	Low resource consumption.
Deployment	Hard to deploy.	Easy to deploy.
Maintain	The entire big data platform needs to be maintained. High maintenance costs.	Only Milvus and programs need to be maintained. Low maintenance costs.
Expand	Easy to extend	Easy to extend

C. *Performance of the Indexing Module*

1) Data analysis: The experimental data was collected from the CBIR indexing phase and compared with the study [18]. Table VIII shows the comparison of retrieval time results.

TABLE VIII. TIME CONSUMPTION OF THE INDEXING FOR DIFFERENT DATASET SIZES AND METHODS

Images size	HDFS	Tachyon	VCBIR
2011	240 s	120 s	446 s
5952	660 s	420 s	1066 s
11963	1260 s	720 s	1991s
17953	1380 s	1140 s	3089 s
20000	1500 s	1200 s	3412 s

Fig. 5 visualizes the graph for comparison. The asterisk line graph represents the time consumed by the method (VCBIR) in the indexing phase. As can be seen from the figure, the performance of this method in the indexing phase is lower than that of the method in study [18] and presents a linear distribution. The reason will be analyzed in detail later.

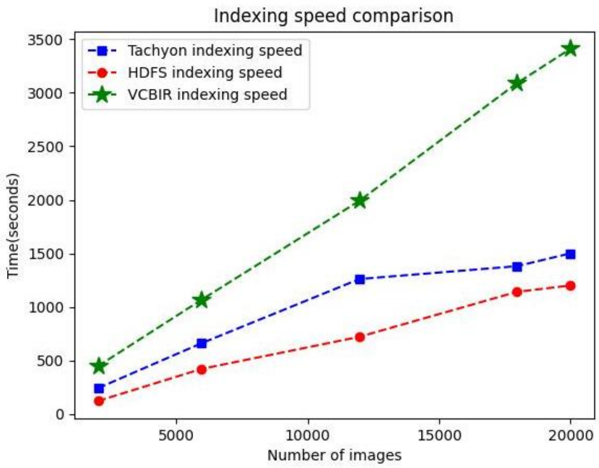

Fig. 5. Indexing speed for different methods.

The research on indexing performance from relevant literature was reviewed and compared. Table IX presents the index performance from different studies and Fig. 6 visualizes the comparison.

TABLE IX. PERFORMANCE COMPARISON OF DIFFERENT INDEXING APPROACHES FOR 10,000 IMAGES

Approach	Description	Time(s)
Centralize method [20]	Sequential method on 1 node	600,000
DIRS method [20]	HBase system+MapReduce on 9 Hadoop nodes	200,000
Luca C. et R. method [21]	HDFS+MapReduce on 1 Hadoop node	2,820
VCBIR method	Milvus +VGG16 on 1 node	1689
Reference paper method [18]	Tachyon+MapReduce on 1 Spark node	460

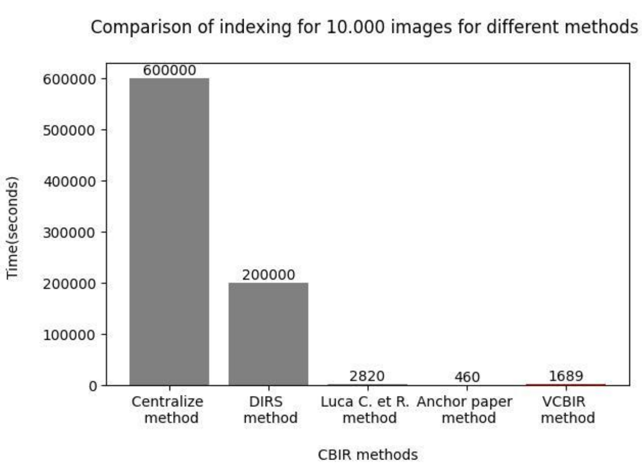

Fig. 6. The time cost visualization for different methods.

From Fig. 6. and Table IX, it can be observed that although the CBIR index performance based on the vector database is a little worse, the performance is still greatly improved in this field. It validates the feasibility of the approach.

2) Result analysis: The reasons why this solution is weaker than the big data solution in study [18] at the indexing stage are as follows:

- Since the literature [18] uses a big data solution, it is easy to implement distributed applications based on the big data platform, so efficiency can be greatly improved. However, the VCBIR is developed using Python. Global Interpreter Lock (GIL) is a mechanism in the Python interpreter. This lock greatly limits the multi-threading performance of the program, which is the main reason why the performance of Python programs is weaker than that of Java. The VGG model relies on tensorflow, and it does not support multi-process execution. Therefore, the entire program can only be executed in a single thread and a single process, and the machine resources cannot be fully utilized. Leading to inefficiency.
- Since the VGG-16 in the indexing stage is used, the model has 16 layers, which is more time consumption than the ordinary texture-based feature extraction algorithm, sacrificing time for quality. Therefore, it is not as efficient as the indexing efficiency in the reference paper [18]. However, the extracted features better characterize the original image. Although the indexing efficiency of this solution is lower than that of the study [18], it is significantly higher than other studies in the same field. It is completely acceptable. Moreover, tasks of this stage can be run in batches or asynchronously at night without disrupting normal operations. There are many solutions to this problem.

D. Performance of the Retrieval Module

1) Data analysis: The retrieval phase of CBIR is the focus of this project and the core problem that needs to be solved. In order to verify its performance, experimental data from the CBIR retrieval phase is collected. Table X lists the experimental results.

TABLE X. COMPARISON OF THE AVERAGE COMPUTING TIME IN THE SEARCHING MODULE

Image dataset size	Parallel k-NN without cache	Parallel k-NN with cache	VCBIR
2011	180 s	120 s	0.27 s
5952	540 s	240 s	0.26 s
11963	1140 s	540 s	0.23 s
17953	1740 s	840 s	0.20 s
20000	1980 s	960 s	0.30 s

For ease of comparison, the chart was visualized, as depicted in Fig. 7. The asterisk line shows the time consumed by the method (VCBIR) for different amounts of data in the retrieval phase. The other two show the time consumed by the solutions in study [18] for different amounts of data in the retrieval phase.

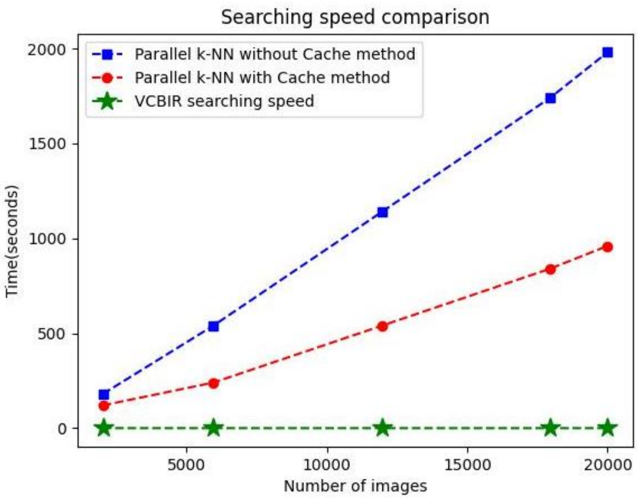

Fig. 7. Searching speed for different methods.

From the figure, it is evident that the performance of the VCBIR solution in the retrieval stage is very outstanding, far higher than the big data solution, and it can almost achieve a response in seconds and real-time return. Moreover, as the amount of data increases, the performance is almost not affected.

As shown in Table XI, the retrieval performance of existing literature was organized to facilitate a better comparison of the current research status in this field.

TABLE XI. PERFORMANCE COMPARISON OF DIFFERENT RETRIEVAL APPROACHES FOR 20,000 IMAGES

Approach	Description	Time (s)
Centralize method [20]	Sequential method on 1 node	25,000
DIRS method [20]	HBase system + MapReduce on 9 Hadoop nodes	15,000
Sakr et al. Method [22]	Parallel retrieval on 1 node Hadoop	1200
reference paper method [18][18] without cache	Parallel k-NN on 1 node Spark without cache	1980
reference paper method [18] with cache	Parallel k-NN on 1 node Spark with cache	960
VCBIR method	Milvus +VGG16 on 1 node	0.3

Again, to visualize the performance improvements, a histogram was generated. Fig. 8 depicts the results of research from various literature sources in this field. The VCBIR in the figure represents the time consumption of the solution in the retrieval field. It is evident that the performance improvement is substantial.

2) Result analysis: From the above results, it is evident that the performance improvement of CBIR applications based on vector databases in the field of retrieval is very huge, even several orders of magnitude higher than the performance of the previous solution. And as the number of images increases, the performance remains basically unchanged. It can be said that real-time response is achieved. Leading the way in performance research in this field.

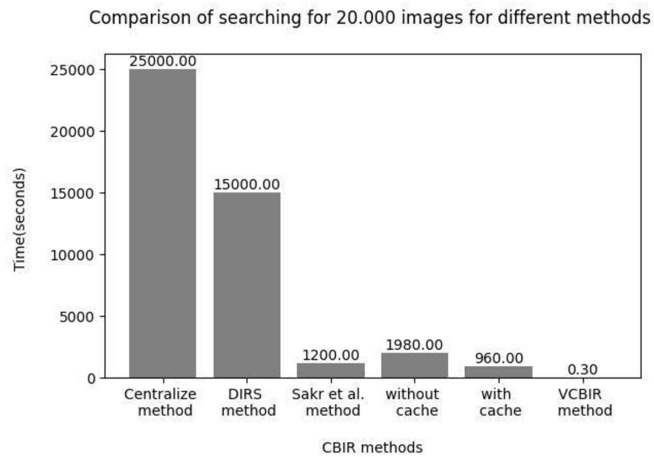

Fig. 8. Searching speed visualization for different methods.

E. Performance Comparison Between Different Datasets

Similarly, the indexing and retrieval times consumption between the ImageClef and ImageNet dataset also were recorded and compared. CBIR applications based on vector databases are comprehensively evaluated. Table XII and Table XIII are the performance comparisons of indexing and retrieval.

TABLE XII. INDEX PERFORMANCE ON TWO DATASETS

Dataset	Nbr of images	Using 1 node	Using 5 nodes	VDBMS
ImageClef	20,000	1200 s	720 s	3412 s
ImageNet	1,461,406	51,283 s	32,000 s	249,400 s

TABLE XIII. RETRIEVAL PERFORMANCE ON TWO DATASETS

Dataset	Sequential k-NN Maillo et al.2015	Parallel k-NN using 1 node	Parallel k-NN using 5 nodes	VDBMS
ImageClef	/	960 s	790 s	0.3 s
ImageNet	107,735 s	42,250 s	34,265 s	0.5 s

F. Performance of the Retrieval Accuracy

However, accuracy is also an important indicator of retrieval performance. In the vector database of this project, the IVF_FLAT index type is utilized, which considers both performance and accuracy. The retrieval function of this application was tested using the Corel-1k dataset, and its evaluation was based on precision and recall. The experimental results have been documented in Table V of Section IV. Fig. 9 shows the precision and recall results.

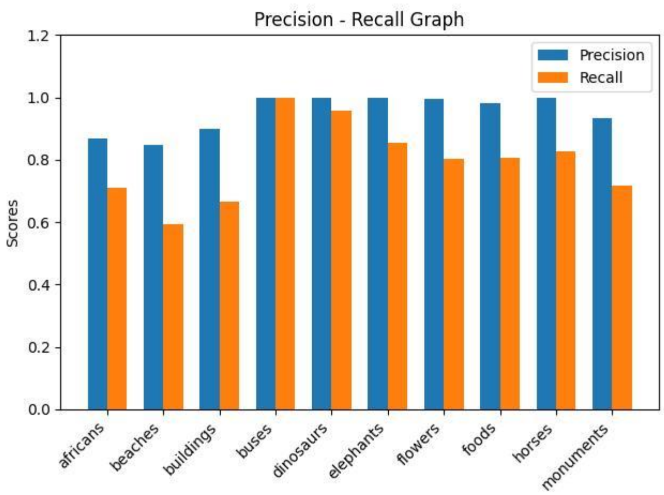

Fig. 9. The precision and recall in Corel-1k for different categories.

From the above figure, it is evident that the retrieval precision of this application is very high, even reaching 100% for some categories, and the lowest is still 60%. The recall is also very well. For the buses category, all images of these image categories can be retrieved based on the query image. It can be concluded that the retrieval accuracy of this application is relatively high and can meet the needs of most daily applications.

VI. Conclusion and Future Research

A. Introduction

Based on the design, a CBIR application utilizing transfer learning, and a vector database was developed. The performance of the application was evaluated by recording the time and accuracy of the indexing and retrieval stages. Experimental results indicated that the performance for the indexing stage did not outperform the big data solution, however for the retrieval stage, the performance of the VCBIR application is much better than that of the reference paper [18] and other solutions in this field. Furthermore since indexing is normally done offline, the performance is still acceptable. Whether it is the time consumption or accuracy of the retrieval stage, the performance is very outstanding, especially the retrieval time is reduced to about 1s, which is improved by several orders of magnitude and can achieve real-time response.

B. Contribution

The introduction of the vector database significantly improves the performance of the CBIR retrieval system, not only increasing the retrieval speed but also ensuring the accuracy. This new technology provides a powerful high-dimensional vector management tool for image retrieval, effectively solving the problems of low retrieval performance and accuracy, thereby bringing faster and more accurate query responses. This improvement improves the user experience and enhances the reliability and trust of the system, indicating that the integration of the vector database and the CBIR system has opened up new space for the development of image retrieval technology.

C. Future Work

- This project currently uses CPU for experiments but considering that GPU has higher performance and efficiency in the field of image processing, we plan to change the program to a GPU version in the future. In this way, we expect to significantly improve the performance and response speed of CBIR applications, allowing them to process large-scale image data faster and more accurately. This improvement will bring more opportunities and advantages to the project, provide users with a better image retrieval experience, and promote the development and application of CBIR technology in practical applications.
- Currently, the project is limited to implementation in a single-machine environment, but there are plans to study how to transform it into a distributed project in the future. This improvement aims to better utilize the advantages of distributed systems and improve the performance and efficiency of CBIR. Additionally, the distributed architecture can enhance the stability and reliability of the system, making it better able to cope with the challenges of large-scale data processing and high concurrent access. This step will bring broader development opportunities to the project, provide users with faster and more reliable image retrieval services, and promote the application and development of CBIR technology in distributed environments.

Acknowledgment

The authors wish to thank The Faculty of Computer Science and Information Technology, Universiti Putra Malaysia for supporting this research and funding this publication.

References

[1] Min, H., & Shuangyuan, Y. (2010, August). Overview of content-based image retrieval with high-level semantics. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 6 , pp. V6-312). IEEE.

[2] Humeau-Heurtier, A. (2019). Texture feature extraction methods: A survey. IEEE access, 7, 8975-9000.

[3] Singh, P., Hrisheekesha, P. N., & Singh, V. K. (2021). CBIR-CNN: content-based image retrieval on celebrity data using deep convolution neural network. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(1), 257-272.

[4] Kapadia, M. R., & Paunwala, C. N. (2018, July). Improved CBIR system using Multilayer CNN. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 840-845). IEEE.

[5] Traina, A. J., Brinis, S., Pedrosa, G. V., Avalhais, L. P., & Traina Jr, C. (2019). Querying on large and complex databases by content: Challenges on variety and veracity regarding real applications. Information Systems, 86, 10-27.

[6] Simran, A., Kumar, P.S., and Bachu, S. (March 2021). Content-based image retrieval using deep learning convolutional neural networks. IOP Conference Series: Materials Science and Engineering (Vol. 1084, No. 1, pp. 012026). IOP Publishing.

[7] Kumar, S., Singh, MK, and Mishra, MK (August 2022). Improving content-based image retrieval using deep learning models. In Journal of Physics: Conference Series (Vol. 2327, No. 1, pp. 012028). IOP Publishing.

[8] Sikandar, S., Mahum, R. and Alsalman, A. (2023). A novel hybrid approach for content-based image retrieval using feature fusion. Applied Sciences, 13(7), 4581.

[9] Stata, R., Bharat, K. and Maghoul, F. (2000). Term vector database: Fast access to web index terms. Computer Networks, 33(1-6), 247-255.

[10] Singla, S., Eldawy, A., Diao, T., Mukhopadhyay, A., and Scudiero, E. (2021, April). Experimental study of large raster and vector database systems. In 2021 IEEE 37th International Conference on Data Engineering (ICDE) (pp. 2243-2248). IEEE.

[11] Alsmadi, M. K. (2017). An efficient similarity measure for content based image retrieval using memetic algorithm. Egyptian journal of basic and applied sciences, 4(2), 112-122.

[12] Niu L., Xu Z., Zhao L., He D., Ji J., Yuan X., Xue M. (2023). Residual vector product quantization for approximate nearest neighbor search. Expert Systems with Applications, 120832.

[13] Chughtai, I. T., Naseer, A., Tamoor, M., Asif, S., Jabbar, M., & Shahid, R. (2023). Content-based image retrieval via transfer learning. Journal of Intelligent & Fuzzy Systems, 44(5), 8193-8218.

[14] Mohammed, M. A., Oraibi, Z. A., & Hussain, M. A. (2023, August). Content based Image Retrieval using Fine-tuned Deep Features with Transfer Learning. In 2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE) (pp. 108-113). IEEE.

[15] Gautam, G., & Khanna, A. (2024). Content Based Image Retrieval System Using CNN based Deep Learning Models. Procedia Computer Science, 235, 3131-3141.

[16] Sadiq, S. S. (2024). Improving CBIR Techniques with Deep Learning Approach: An Ensemble Method Using NASNetMobile, DenseNet121, and VGG12. Journal of Robotics and Control (JRC), 5(3), 863-874.

[17] Thanikachalam, R., Thavasimuthu, R., Arulkumar, V., Prabin, S. M., Saranya, N., & Devi, R. (2024). T2T-ViT: A Novel Semantic Image Mining Approach for Improving CBIR Using Vision Transformer.

[18] Mezzoudj, S., Behloul, A., Seghir, R., & Saadna, Y. (2021). A parallel content-based image retrieval system using spark and tachyon frameworks. Journal of King Saud University-Computer and Information Sciences, 33(2), 141-149.

[19] Traina, AJ, Brinis, S., Pedrosa, GV, Avalhais, LP and Traina Jr, C. (2019). Querying large complex databases by content: Diversity of practical applications and challenges in accuracy. Information Systems, 86, 10-27.

[20] Zhang, J., Liu, X., Luo, J., & Lang, B. (2010, December). Dirs: Distributed image retrieval system based on mapreduce. In 5th International Conference on Pervasive Computing and Applications (pp. 93-98). IEEE.

[21] Costantini, L., & Nicolussi, R. (2015). Performances evaluation of a novel Hadoop and Spark based system of image retrieval for huge collections. Advances in Multimedia, 2015, 11-11.

[22] Sakr, N. A., ELdesouky, A. I., & Arafat, H. (2016). An efficient fast-response content-based image retrieval framework for big data. Computers & Electrical Engineering, 54, 522-538.