High-Dimensional Image Retrieval via Adaptive Subspace Dimension Product Quantization

Lei Lu[1], Yu Li[2*], Huamin Liu[3]
[1]Information Engineering College, Jiangxi College of Applied Technology, Ganzhou 341000, China
[2]Public Basics Department, Ganzhou Polytechnic, Ganzhou 341000, China
[3]Information Engineering College, Ganzhou Polytechnic, Ganzhou 341000, China
E-mail: 15879799310@163.com, yutou6094@163.com, liu13767715@163.com
[*]Corresponding author

Keywords: product quantization, self-adaption, high-dimensional data, image retrieval, neighbor search, subspace

Received: June 3, 2025

To address low search accuracy and long search time in product quantization (PQ) algorithms for large-scale datasets, the Adaptive Subspace Dimension Product Quantization (ASDPQ) algorithm is proposed. It optimizes subspace partitioning by adaptively choosing the number of sub-spaces based on quantization error comparison, speeding up the search. During training, it uses two encoding patterns and selects the optimal one for efficient quantization. A high-dimensional data image retrieval model is developed. In experiments on SIFT and GIST ANN search datasets, ASDPQ outperforms OPQ and PQ algorithms, with recall rates of 0.84 and 0.97, and search times of 3.135ms and 5.374ms respectively. It also reduces addition computation by 4.54% and 6.96% compared to PQ. When integrated into an image retrieval system, it achieves a similarity rate of over 80% and an average shortest retrieval time of 2.63ms, demonstrating its effectiveness and reliability in high - dimensional data image retrieval.

Povzetek: Za iskanje po visokodimenzionalnih slikovnih podatkih je razvit algoritem Adaptive Subspace Dimension Product Quantization (ASDPQ), ki dinamično prilagaja delitev v podprostore glede na napako kvantizacije in s tem izboljša hitrost iskanja. ASDPQ uporablja dva vzorca kodiranja ter samodejno izbere optimalnega, kar omogoča učinkovitejše kvantiziranje in iskanje najbližjih sosedov v velikih slikovnih zbirkah.

1 Introduction

In the context of rapid progress in information technology, cloud computing, big data, artificial intelligence, and other emerging technologies have been widely applied, which has led to the generation of massive types of high-dimensional information in the Internet, including text, video, images, audio, and sensor data. These data not only have huge volumes, but also complex structuralfeatures, manifested as significant characteristics such as high dimensionality, nonlinearity, and heterogeneity [1-3]. However, these data usually have high-dimensional and large-scale characteristics, limited storage space, and users' increasing demand for information retrieval speed [4-5]. Therefore, developing efficient and accurate high-dimensional data image search algorithms has become a key research direction in the academic community.

To tackle low accuracy in traditional high-dimensional data image retrieval, Yan et al. introduced multi-view deep neural networks into hash learning, developing a supervised multi-view hash model with higher retrieval performance [6]. For security improvement, Feng et al. proposed a privacy-preserving image retrieval scheme based on image encryption, showing excellent encryption and retrieval performance with higher accuracy [7]. Johnson et al. proposed Product Quantization (PQ) search methods to reduce computing resources in image retrieval, achieving 55% of theoretical peak performance and an 8.5-fold speed increase [8]. Feng et al. proposed a PQ adversarial generation method to address deep PQ network shortcomings, creating adversarial samples to improve retrieval performance [9]. In graph indexing, Wang et al. proposed a connection-based graph native query difficulty measurement method, defining Steiner difficulty, which correlated better with actual query workload across datasets [10]. In water resource management, Alawsi et al. combined data preprocessing, artificial neural networks, particle swarm optimization, shrinkage coefficient, and chaotic gravity search to construct a hybrid algorithm, outperforming comparison algorithms in statistical indicators [11].

Since the 21st century, the Approximate Nearest Neighbor (ANN) search technique has received widespread attention. Among them, vector quantization-based algorithms have been broadly utilized in fast image retrieval due to their effectiveness in encoding high-dimensional visual features. For example, Yu et al. proposed PQ networks, residual PQ networks, and temporal PQ networks, which have achieved state-of-the-art performance in fast image and video retrieval [12]. Faced with issues such as local sensitive hashing and limited accuracy in handling item quantities using index-based methods, Lian et al. proposed a PQ collaborative filtering method. The results denoted that this method was significantly superior to state-of-the-art comparison

algorithms, improving recommendation performance [13]. Considering the problem of low search efficiency of traditional algorithms when performing similarity searches on large-scale time series datasets, Zhang et al. proposed a dynamic time warping method based on PQ. The results showed that this method achieved the best trade-off between query efficiency and retrieval accuracy compared to traditional methods [14]. To bridge the semantic gap between open vocabulary and visual content, Fakhfakh R et al. developed a personalized image retrieval system framework. This framework selected the most relevant images based on specified queries, user interests, and semantic interpretations. The results showed that this method achieved an average precision of 0.675, surpassing other works using the same database under similar conditions [15]. The number of codewords in a codebook is determined by experience, leading to an imbalance in the representation capabilities of codewords, which results in redundancy or insufficiency and reduces retrieval performance. To address this issue, Gu L et al. introduced an entropy-optimized deep weighted PQ method. The results showed that this method not only improved retrieval performance but also enhanced the representation capabilities of codewords and balanced their allocation [16]. Currently, deep learning-based hashing and quantization methods heavily rely on expensive label information in large-scale image retrieval, failing to fully utilize data resources. To tackle this problem, Zhao X et al. proposed a self-supervised method that does not require label information, specifically the contrast self-supervised weak orthogonal PQ. The results indicated that this method achieved better performance on the CIFAR-10, NUS-WIDE, and FLICKR25K datasets [17]. The summary results of the existing studies are shown in Table 1.

Table 1: Summary of relevant work.

Author's name	Algorithm name	Method description	Evaluation indicators	Methodological flaws
Yan C et al. [6]	Supervise the multi-view hash model	The multi-view deep neural network is introduced into the field of hash learning	Image retrieval performance	Low accuracy
Feng Q et al. [7]	Image retrieval scheme for privacy protection based on image encryption	Develop a privacy protection image retrieval scheme based on image encryption	Encryption and retrieval performance, retrieval accuracy	Security is not sufficient, especially in the process of data transmission and storage vulnerable to attacks
Johnson J et al. [8]	Vigorous search, approximate search and compressed domain search methods for PQ	The methods of violent search, approximate search and compressed domain search are proposed for PQ	The nearest neighbor implementation runs at speed	The consumption of computing resources is large, which makes it difficult to meet the real-time retrieval requirements of large-scale data
Feng Y et al. [9]	PQ adversarial generation method	A PQ adversarial generation method is proposed to mislead the target product quantitative retrieval model	High dimensional data image retrieval performance	Deep PQ network has defects in fast image retrieval and is vulnerable to adversarial sample attacks
Wang Z et al [10]	A difficulty measurement method for connected graph native queries	A new connection-based graph native query difficulty measurement method is proposed, and the Steiner difficulty is defined	The correlation between Steiner difficulty and actual query workload	The response performance of the traditional map index varies greatly for different queries, resulting in unstable service quality
Alawsi M A et al. [11]	Hybrid algorithm (combining data preprocessing, artificial neural network, particle swarm optimization based on contraction coefficient and chaotic gravity search algorithm)	The data preprocessing and artificial neural network are combined with particle swarm optimization based on contraction coefficient and chaotic gravity search algorithm	Performance of various statistical indicators	The algorithm is complex, the calculation cost is high, and it is mainly aimed at a specific field (water resources management)
Yu T et al. [12]	PQ network, residual PQ network and time PQ network	PQ network, residual PQ network and time PQ network are proposed	Fast image and video retrieval performance	/
Lian D et al. [13]	PQ collaborative filtering method	The PQ collaborative filtering method is proposed	Recommended performance	Local sensitive hashing and index-based methods deal with a limited number of items and low accuracy
Zhang H et al [14]	Dynamic time warping method based on PQ	A dynamic time warping method based on PQ is proposed	The trade-off between query efficiency and retrieval accuracy	Traditional algorithms are inefficient in searching large-scale time series data sets

Fakhfakh R et al. [15]	Personalized image retrieval system framework	The framework of personalized image retrieval system is proposed to select images according to query, user interest and semantic interpretation	Average mean accuracy	It is difficult to bridge the semantic gap between open vocabulary and visual content
Gu L et al. [16]	Energy optimization deep weighted PQ method	An entropy optimization deep weighted PQ method is proposed	Search performance, code word representation capability, and code word allocation balance	The number of code words in a codebook depends on experience, and the representation capacity of code words is unbalanced
Zhao X et al. [17]	Compare the self-supervised weak orthogonal PQ	A self-supervised method is proposed that does not require labeled information	Function	Hashing and quantization methods based on deep learning rely too much on expensive tag information

In summary, despite progress in high-dimensional data image retrieval with intelligent algorithms, there are application drawbacks. High-dimensional data's sparsity and complexity lead to high computational costs, limiting retrieval speed and real-time performance. The "curse of dimensionality" increases noise, affecting retrieval accuracy. Besides, the algorithm's insufficient understanding of image semantics makes it hard to capture user intent accurately, causing retrieval results to deviate from expectations.

Applying PQ to high-dimensional data image retrieval faces low accuracy (due to fixed sub-space partitioning not adapting to complex data structure, causing large quantization errors) and long retrieval times (from unnecessary computation in large-scale data processing). The proposed ASDPQ algorithm dynamically adjusts sub-space partitioning by adaptively selecting dimensions based on data's local characteristics, using finer dimensions in dense areas for accuracy and coarser ones in sparse areas to reduce complexity and time, thus enhancing image retrieval performance.

Based on this, the following clear research objectives are proposed: (1) Reduce retrieval time: Improvement of PQ algorithm by adaptive subspace selection reduces computational complexity and speeds up high-dimensional data retrieval. (2) Improve recall rate: The process of subspace division and quantization coding is optimized to improve the accuracy of image retrieval and ensure that the target image is accurately found in large-scale datasets. (3) Minimize computational overhead: By improving the design of the algorithm, unnecessary computations are reduced and the efficiency of the algorithm is improved to make it more suitable for resource-limited environments.

To achieve these objectives, the research first improves the Adaptive Subspace Dimension Product Quantization (ASDPQ) algorithm by adaptively selecting dimensions of sub-spaces. Then, by combining content-based image retrieval methods, a new image retrieval model suitable for high-dimensional data is proposed. The ASDPQ algorithm's core innovation is its adaptive subspace selection mechanism, setting it apart from existing methods. Traditional PQ algorithms have fixed sub-spaces and dimensions, lacking flexibility. ASDPQ dynamically adjusts subspace dimensions based on data characteristics, comparing quantization errors to select the best dimension combination. For complex high-dimensional image data, it chooses different dimensions for sub-spaces according to local features, improving search accuracy, reducing computational load, and enhancing overall performance. This study aims to solve the efficiency problem in high-dimensional image data retrieval, improve the performance and practicality of image retrieval, provide new ideas and methods for the current image retrieval field, and promote the development and application of image retrieval technology.

2 Methods and materials

2.1 ASDPQ algorithm design

The PQ algorithm is crucial for high-dimensional data image retrieval, quantifying and decomposing high -dimensional vectors into low-dimensional sub-spaces to reduce storage and computational complexity, improving retrieval speed while maintaining accuracy [18-19]. However, traditional PQ algorithms have naive spatial partitioning, sub-optimal subspace partitioning, and lack candidate set selection based on quantization errors during ANN search, limiting accuracy. They also perform poorly in anomaly detection when normal and abnormal data are mixed [20]. In response to the above issues, the research combined with the characteristics of the vector distribution of the dataset optimized the query distance table stage in the algorithm and proposed an improved PQ algorithm. This algorithm adaptively selected the dimension of sub-spaces by comparing quantization errors, reduced the number of vector sub-spaces in the dataset, and accelerated the search speed. Compared with the standard PQ method, it has fewer sub-spaces, which can accelerate the search speed during the search process, as shown in Figure 1.

In Figure 1, based on the comparison of quantization errors between Mode 1 and Mode 2, ASDPQ adaptively selects the appropriate subspace dimension for coding. The algorithm adopts two encoding modes, quantizes the raw data through PQ, and saves the parameters. The input data is divided into different subvectors, encoded using different codebooks, and recorded for replacement. Choosing a mode with small quantization error may adjust the number of sub-spaces. The algorithm trains two PQ codebooks to adaptively partition and encode the dataset vectors into sub-spaces. In indexing and retrieval, the

Euclidean distance between the query vector and the dataset vector is calculated using the ADC method to achieve similarity matching. The implementation process of the ASDPQ algorithm in this design is shown in Figure 2.

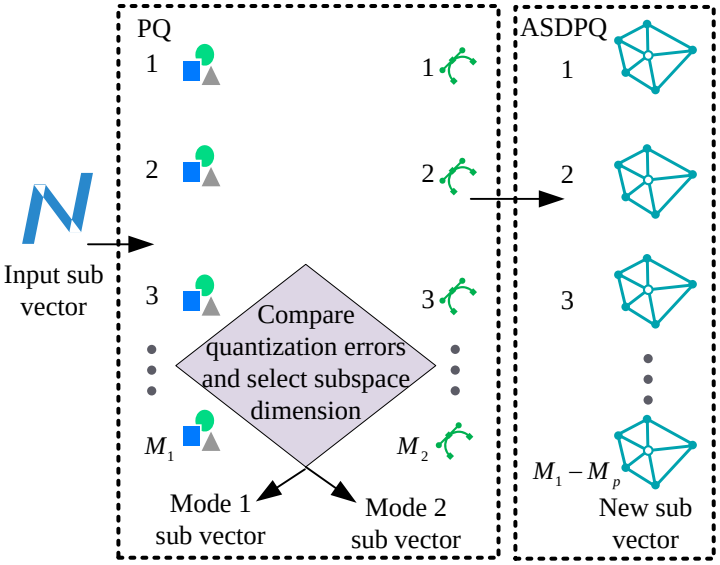

Figure 1: Schematic diagram of sub-vectors.

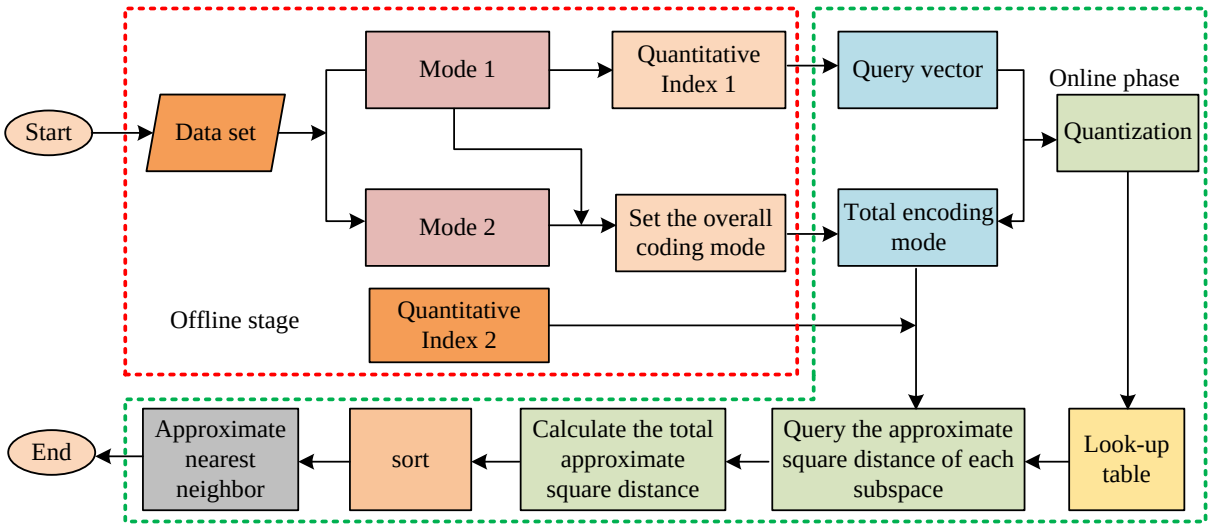

Figure 2: Flowchart of ASDPQ algorithm index establishment and retrieval process.

In Figure 2, the process adopts two encoding modes with different numbers of sub-spaces and clusters. In the training of the algorithm, the dataset is first trained in two different modes, namely mode one and mode two. These two modes have differences in the setting of the number of sub-spaces and cluster classes k_1 / k_2 to adapt to different data features and quantization requirements. After the training is completed, the system will save the generated index, codebook, and quantization error information. The index is used for fast data retrieval, while the codebook stores the cluster centers of each subspace. The quantization error records the size of the error introduced during the quantization. The D-dimensional dataset is decomposed into M sub-vectors using the K-means algorithm, and the subspace codebooks $\mathbf{C}_{\mathbf{ml}}$ and $\mathbf{C}_{\mathbf{m2}}$ are trained and encoded. Finally, the sub-codebooks are integrated to obtain the PQ total codebook, as shown in formula (1).

$$\begin{cases} X^m = \left\{ x_{i,\frac{D(m-1)}{M}+1}, x_{i,\frac{D(m-1)}{M}+2}, \ldots, x_{i,\frac{Dm}{M}} \right\}^T, \\ m = 1,2,\ldots,M \\ \mathbf{C}_{\mathbf{ml}} = \mathbf{C}_{\mathbf{ml}}{}^1 \times \mathbf{C}_{\mathbf{ml}}{}^2 \times \ldots \times \mathbf{C}_{\mathbf{ml}}{}^{M_1} \\ \mathbf{C}_{\mathbf{m2}} = \mathbf{C}_{\mathbf{m2}}{}^1 \times \mathbf{C}_{\mathbf{m2}}{}^2 \times \ldots \times \mathbf{C}_{\mathbf{m2}}{}^{M_2} \end{cases} \quad (1)$$

In formula (1), $\mathbf{C}_{\mathbf{ml}}$ is composed of M_1 subcodebooks $\mathrm{C}_{\mathrm{ml}}^{\mathrm{i}}\left(\mathrm{i}=1,2,\cdots,\mathrm{M}_1\right)$, each containing k_1 codewords; $\mathbf{C}_{\mathbf{m2}}$ is composed of M_2 subcodebooks $\mathrm{C}_{\mathrm{m2}}^{\mathrm{j}}$, each containing k_2 codewords. From this, it can be seen

that $\mathbf{C_{m2}}$ contains M_2 times k_2 codewords, and the same applies to $\mathbf{C_{ml}}$. M_1 represents the number of sub-spaces in pattern one, while M_2 represents the number of sub-spaces in pattern two. k_1 is the number of codewords in the sub-codebook of pattern one (256), and k_2 is the number of codewords in the sub-codebook of pattern two (512), as shown in formula (2).

$$\begin{cases} \mathbf{C_{ml}}^{m} = \left\{ \mathbf{C_{ml1}}^{m}, \mathbf{C_{m12}}^{m}, \ldots, \mathbf{C_{mlk_1}}^{m} \right\} \\ \mathbf{C_{m2}}^{m} = \left\{ \mathbf{C_{m21}}^{m}, \mathbf{C_{m22}}^{m}, \ldots, \mathbf{C_{m2k_2}}^{m} \right\} \end{cases} \quad (2)$$

In the training process based on ASDPQ algorithm, two encoding modes are first trained on the dataset to optimize quantization performance and improve retrieval efficiency. Specifically, in pattern one, each dataset vector x_i will be quantized into M_1 indices for representation. This quantization process divides the high-dimensional vector space into multiple sub-spaces and quantizes them separately in each subspace, thereby representing the original vector as a combination of multiple subspace indices, as shown in formula (3).

$$X_i' = C_{mli}^{h_l}, \quad i = 1, 2, \cdots, n \quad (3)$$

In formula (3), X_i' is the i th component of the quantized vector X', h_l is the codeword index closest to the component X_i of the original vector obtained by searching in the sub-codebook $\mathbf{C_{ml}}$, and $C_{mli}^{h_l}$ is the i th component of the codeword indexed as h_l in the sub-codebook $\mathbf{C_{ml}}$. The research will select an appropriate encoding mode based on the size of the quantization error, then perform quantization processing on the query vector, and finally calculate the query distance table. Using these tables, the squared distances between the query vector and the vectors in the data set are approximated, summed to obtain the total approximate squared distance, and sorted to determine the nearest neighbours, thus completing the approximate nearest-neighbour search to obtain the total approximate squared distance, $Dis(q, X_i)$, as shown in in formula (4).

$$Dis(q, X_i) = \sum_{m=1}^{M_2} Dis_m(q, X_i) \quad (4)$$

In formula (4), X_i indicates the ANN sought, and q denotes the query vector. The ASDPQ algorithm obtains the minimum value by sorting the total approximate square distance, completing the index establishment. In ANN search, the ASDPQ code is used to calculate the Euclidean distance between the query vector and the dataset vector, ensuring the same ranking without calculating the square root. In the retrieval process of ASDPQ algorithm, the entire calculation process is divided into two parts: constructing the distance table and searching for the calculated distance. Firstly, the dataset q is divided into M_1 or M_2 sub-vectors, and the squared Euclidean distance between the query vector sub-vectors and the corresponding codewords in the codebook is calculated and stored in the query distance table, as shown in formula (5).

$$D_{m2}(i, j) = \left\| q_i - C_{m2}^{j} \right\|^2, \quad i = 1, 2, \cdots, M_2, \quad j = 1, 2, \cdots, k_2 \quad (5)$$

In formula (5), $D_{m2}(i, j)$ represents the distance between the i sub-vector q_i of the query vector and the j codeword in the pattern two sub-codebook C_{m2}. k_2 is the number of codewords in the subcode book C_{m2}. The study utilizes a pre-constructed lookup table to obtain the nearest neighbor distance through calculation and sorting. Through the above design, the pseudo-code of ASDPQ algorithm can be obtained, as shown in Figure 3.

The ASDPQ algorithm first divides the dataset into two modes during the training phase, and then constructs sub-codebooks for each mode to form a complete codebook. In the retrieval stage, the query vector is broken down into sub-vectors, and the nearest codeword is identified from the corresponding sub-codebook. The approximate distance between the query vector and each vector in the dataset is calculated, and the results are sorted to obtain the nearest neighbor.

```
Algorithm ASDPQ
Input: Dataset X, query vector q
Output: Nearest neighbors of q

// Training phase
1. Split X into two modes: Mode1 and Mode2
2. For Mode1:
   - Cluster into M1 sub - codebooks C_ml^i, i = 1..M1, each with k1 codewords
3. For Mode2:
   - Cluster into M2 sub - codebooks C_m2^j, j = 1..M2, each with k2 codewords
4. Combine sub - codebooks to form C_ml and C_m2

// Search phase
5. For each sub - vector x_i of query q:
   - Find nearest codeword in corresponding sub - codebook
6. Calculate approximate distances D(q, X_i)
7. Sort and return nearest neighbors
```

Figure 3: Pseudo-code of ASDPQ algorithm (Image source: Author's own drawing).

Sub-space dimensions significantly impact the ASDPQ algorithm's retrieval performance. A low-dimension reduces computational complexity and time but causes feature loss, increasing quantization errors and lowering precision (recall, mAP) as key features are hard to retain. A high-dimension preserves features well but increases computational load, storage costs, time, and may harm stability due to overfitting. ASDPQ avoids traditional dimensionality reduction. Its adaptive mechanism analyzes data and dynamically sets dimensions by comparing quantization errors under different sub-space divisions.

2.2 Construction of high-dimensional data image retrieval model based on ASDPQ algorithm

After designing the ASDPQ algorithm, a high-dimensional data image retrieval model based on it is proposed to boost image retrieval performance. Image retrieval technology has evolved from text-based (TBIR) to content-based. TBIR uses text descriptions for image search, combining natural language processing and computer vision, but has drawbacks like time-consuming manual annotation and diverse Chinese descriptions causing inefficiency. Content-based retrieval extracts and compares image features with database images [21]. Based on this, a content-based image retrieval method is studied, combined with the designed ASDPQ algorithm, to propose an image retrieval model suitable for high-dimensional data. The overall architecture of the model is denoted in Figure 4.

Figure 4 shows the model's two-stage processing: offline and online. The offline stage builds the dataset feature library and index codebook. The online stage uploads user query images and returns similar results via feature matching. For image retrieval, the model extracts HSV, SIFT, and GIST features. SIFT feature points, local extrema detected in different scale spaces, are scale - and rotation-invariant and stable against lighting and noise, crucial for image matching and object recognition. The implementation of SIFT algorithm is denoted in Figure 5.

In Figure 5, after inputting the image to be retrieved, the model extracts its feature points and establishes a feature point set. Feature matching is achieved by calculating the distance between feature point sets, where the smaller the distance, the higher the similarity. Matching must reach the set image registration rate threshold to be considered successful, otherwise it will be retrieved again. GIST feature extraction first constructs a two-dimensional Gabor filter bank, which utilizes its good frequency and directional selectivity to filter the image and extract local texture and edge information. Next, by performing scale and rotation transformations on Gabor filters, multi-scale and multi-directional filter banks are generated to capture the features of the image at different scales and directions, as denoted in formula (6).

$$\begin{cases} g_{mn}(x,y) = \alpha^{-m}(x',y'), \alpha \succ 1 \\ x' = \alpha^{-m}(x\cos\theta, y\cos\theta) \\ y' = \alpha^{-m}(-x\sin\theta, y\sin\theta) \\ \theta = n\pi/(n+1) \end{cases} \tag{6}$$

In formula (6), θ means the rotation angle, α^{-m} denotes the dilation scale factor, and m and n represent the scale and direction of the Gabor filter, respectively. After the above processing, the image is divided into k small blocks, each of which is filtered using Gabor filter banks and converted into 4 × 8 data. The entire image is then converted into 4 × 8 × k data. In HSV feature extraction, the image is converted to the HSV color space and the hue, saturation, and brightness feature values of each pixel are extracted.

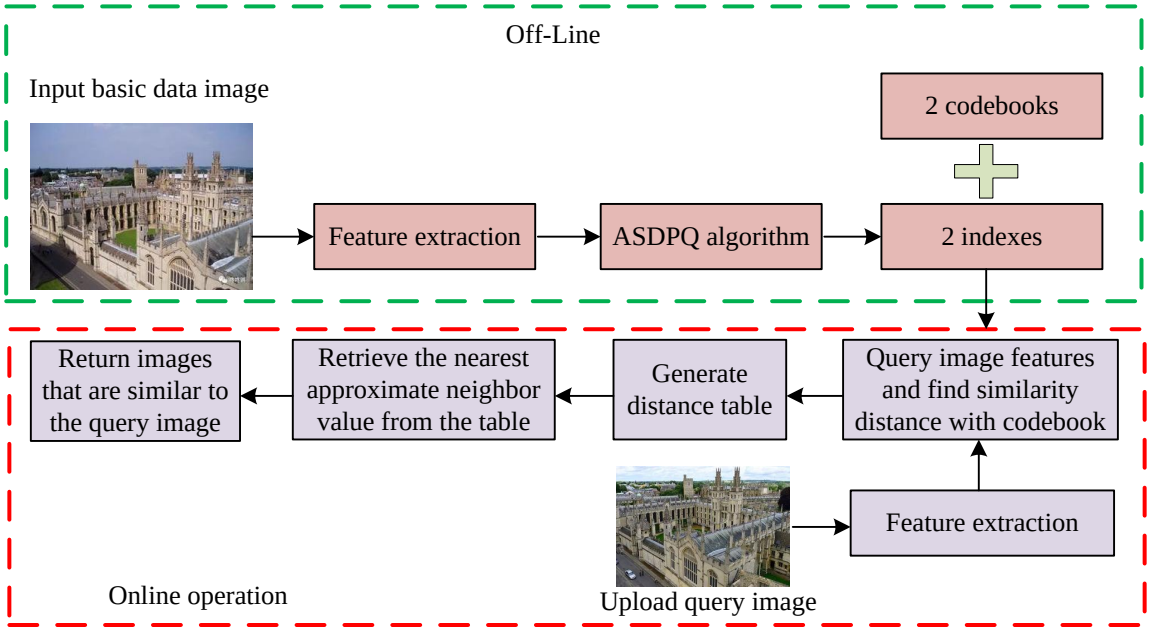

Figure 4: Overall architecture diagram of high-dimensional data image retrieval model based on ASDPQ algorithm.

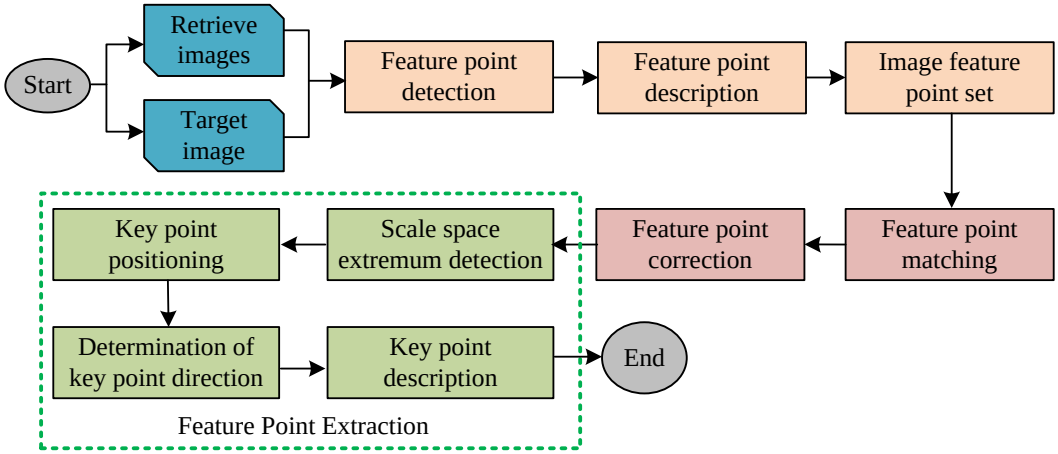

Figure 5: Implementation process of SIFT algorithm.

The high-dimensional data image retrieval model based on ASDPQ algorithm achieves efficient retrieval through two-stage offline and online operations. The model first performs feature extraction on the image dataset to obtain high-dimensional feature representations for each image. Then, the ASDPQ algorithm is used to quantify the features and generate indexes and codebooks for the image dataset. This process compresses high-dimensional data into low-dimensional representations while preserving key feature information, providing efficient data support for subsequent retrieval. After the user uploads the query image, the model first performs HSV color, SIFT, and GIST feature extraction on the query image to obtain its high-dimensional feature representation. The study sequentially connects vectors of different features (e.g., 128 - D SIFT, 180-D HSV, and 960-D GIST) to form a long final image feature vector (1268-D in the example). Each feature vector is normalized using Z-score before fusion. The feature combination is chosen for their complementary nature in expressing different image aspects (local details, color, and global structure). This multi-feature fusion comprehensively describes image content, improving image retrieval accuracy.

Finally, utilizing advanced ASDPQ algorithm, the query features submitted by users are accurately matched with features pre-stored in a large dataset. Through this matching process, the similarity between the query and the features of each dataset is calculated and sorted in descending order of similarity, ultimately returning the image result that is most similar to the user's query. This model is suitable for large-scale high-dimensional data scenarios and has high practicality and efficiency in the field of image retrieval.

High-dimensional data image retrieval is widely used, with data often containing sensitive info like biometric features and medical images, and leaks having serious effects. The ASDPQ algorithm can boost privacy

protection. In sensitive image retrieval, it encrypts original images during preprocessing, then performs adaptive subspace operations on encrypted data to create an encrypted index. During retrieval, the query image is encrypted and matched with the index, keeping the process encrypted to prevent plain-text exposure. Its adaptive feature adjusts subspace dimensions based on data, precisely segmenting and quantizing large-scale sensitive data, reducing privacy breach risks while maintaining efficiency and privacy.

3 Results

3.1 Experimental setup and ASDPQ algorithm training

To validate the ASDPQ algorithm, SIFT and GIST ANN search datasets were used. SIFT has high-dimensional (128-512) feature vectors for precise, computation-intensive tasks, with a large training set and dataset library (millions/billions of vectors). The SIFT dataset has 1 million 128-dimensional descriptors (100,000 for training, 10,000 for testing). GIST has lower-dimensional (up to 960) vectors focusing on global image description, with 500,000 training and 1,000 query descriptors. The SIFT dataset (feature vector dimensions: 128 - 512) was chosen for precise, computation-intensive applications and to validate the algorithm in high-dimensional scenarios. The GIST dataset, with relatively lower feature vector dimensions focusing on global image description, was selected to contrast with SIFT and comprehensively evaluate the ASDPQ algorithm's performance across different high-dimensional data types.

In data preprocessing, features were normalized using Z-score to have a mean of 0 and standard deviation of 1 before training and testing. The dataset was split into an 80% training set and 20% test set, randomly repeated 10 times to assess performance stability. For recall calculation, ground truth was the 100 nearest neighbors of the query point found by brute-force search under Euclidean distance. All experiments were repeated 10 times, and average and standard deviation results were reported for performance stability. The experimental operating environment and parameter design are denoted in Table 2.

Table 2: Experimental operating environment and parameter design.

Experimental environment		Setting items
Operating environment	CUDA version	11.4.0
	Central processing unit	NVIDIARTX4090Ti
	Memory	16.00GB
	Batch size	8
	Initial value of learning rate	0.001
	Optimizer	Adam
	Software environment	MatlabR2018a
Parameter settings	Mode 1	The subspace is 8 and the number of clusters is 256
	Mode 2	The subspace is 4 and the number of clusters is 512

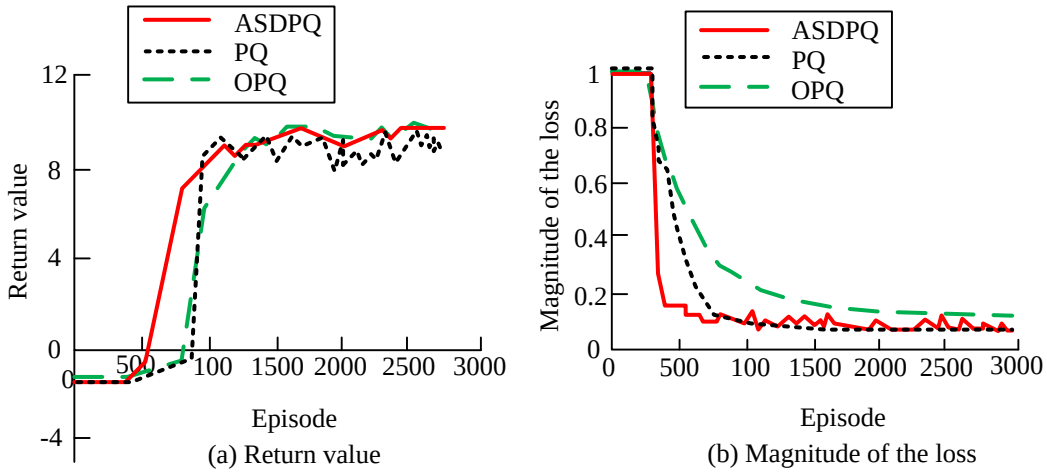

Figure 6: ASDPQ algorithm training results.

The comparison methods chosen for this test were PQ and Optimized Product Quantization (OPQ). Specifically, the study utilized the PQ and OPQ implementations provided by the Faiss library. The Faiss library, developed by Facebook AI Research, is designed for efficient similarity search and dense vector clustering. For the PQ algorithm, the study selected 128 sub-spaces, each containing 256 cluster centers. For the OPQ algorithm, the same subspace dimensions and number of clusters as PQ were used, and the optimization steps of OPQ were applied. All algorithms' hyperparameters were optimized through a grid search to ensure fair comparisons under identical conditions.

In the experimental setup, grid search was used for hyperparameter tuning of the ASDPQ algorithm. Key hyperparameters like subspace dimensions and

quantization coding parameters were set with possible value ranges, and candidate values were determined based on data and experience. Grid search explored these combinations to find the optimal ones for the algorithm's performance on the validation set. To ensure fairness, all comparison algorithms' hyperparameters were also tuned using grid search for an objective comparison under identical conditions. The evaluation indicators selected wee the recall rate for evaluating search accuracy, as well as the search time and computational savings for evaluating search speed. For the ASDPQ algorithm, the training results are shown in Figure 6.

From Figure 6 (a), overall, both OPQ and PQ algorithms reached their maximum return values after 1,000 episodes. The ASDPQ algorithm reached its maximum return value after 500 episodes, and the speed of obtaining return value was faster. Among them, compared with PQ and OPQ algorithms, ASDPQ had more stable data changes, indicating that the proposed algorithm performed better. From Figure 6 (b), both OPQ and PQ algorithms gradually stabilized after 1,000 episodes, and there was ineffective path learning during the learning. ASDPQ learned along the path with the highest value and gradually stabilized after 327 episodes, indicating that the ASDPQ algorithm had better search performance and experimental results at the same episode.

To assess the ASDPQ algorithm's scalability, the study examined its computational complexity (training: $O(d \cdot n + k \cdot d \cdot \log n)$, search: $O(d \cdot \log k)$, with k subspaces) across datasets of size n and dimension d. The PQ algorithm has training complexity $O(d \cdot n)$ and search complexity $O(d \cdot k)$, while the OPQ algorithm has similar complexities with an extra optimization step. ASDPQ maintains high search accuracy with comparable complexity to PQ and OPQ, demonstrating good scalability.

3.2 ASDPQ algorithm performance testing

After completing algorithm training, a comparative analysis was carried out on the recall rates of ASDPQ algorithm with OPQ and PQ algorithms to test the search accuracy of the raised algorithm. The recall tests of each algorithm on SIFT and GIST datasets are shown in Figure 7.

From Figure 7 (a), in the SIFT dataset, overall, as the number of iterations increased, the recall rates of ASDPQ algorithm and other compared algorithms gradually increased and tended to stabilize. The maximum recall rates of ASDPQ algorithm and traditional OPQ and PQ algorithms were 0.84, 0.78, and 076, respectively. Among them, the ASDPQ algorithm had the best recall performance and converged the fastest, with a 7.7% improvement compared to the traditional OPQ algorithm. By calculating the 95% confidence interval, it can be confirmed that the recall rate of ASDPQ algorithm is significantly higher than that of PQ and OPQ algorithms (p<0.05). From Figure 7 (b), on the GIST dataset, overall, as the number of iterations increased, the recall rates of ASDPQ algorithm and other compared algorithms gradually increased and tended to stabilize. The max recall rates of ASDPQ algorithm and traditional OPQ and PQ algorithms were 0.97, 0.94, and 0.88, respectively. Among them, the ASDPQ algorithm had the best recall performance, which was 3.2% higher than the traditional OPQ algorithm. Similarly, by calculating the 95% confidence interval, it can be confirmed that the recall rate of ASDPQ algorithm is significantly higher than that of PQ and OPQ algorithms (p<0.05). The above results indicate that the ASDPQ algorithm has higher search accuracy and better performance than other compared algorithms, and can be effectively applied in high-dimensional data image retrieval. The search time test results of ANN search with different algorithms on SIFT and GIST datasets are denoted in Table 3.

According to Table 3, the search time of ASDPQ algorithm on SIFT and GIST datasets was 3.135ms and 5.374ms, respectively, and the performance was better than the comparison algorithms OPQ and PQ. Among them, in the SIFT dataset, the ASDPQ algorithm performed the best, improving by 20.4% and 21.3% respectively compared to traditional OPQ and PQ algorithms. Although the ASDPQ took a longer time in the 'computing distance table/ms' phase, this was due to the more complex adaptive subspace partitioning and encoding operations performed by ASDPQ to adapt to the characteristics of the dataset vectors. While these operations increased the time cost during the computation of the distance table, they reduced unnecessary computations in subsequent queries and searches due to more precise subspace partitioning and encoding, thus shortening the overall search time. The results denote that the ASDPQ algorithm has superior performance in search speed and can achieve fast retrieval of high-dimensional data images. Next, the study analyzed the computational savings of different algorithms, as shown in Figure 8.

Figures 8 (a) and 8 (b) show the comparison of computational complexity on the SIFT dataset and GIST dataset, respectively. From Figure 8 (a), on the SIFT dataset, both OPQ and ASDPQ algorithms saved addition computation compared to PQ algorithm. Among them, the ASDPQ algorithm saved 4.54% of the addition computation and had the best effect. From Figure 8 (b), on the GIST dataset, similar to the PQ algorithm, both OPQ and ASDPQ algorithms saved the amount of addition computation. Among them, the ASDPQ algorithm saved 6.96% of the addition computation and had the best effect. The above results show that the ASDPQ algorithm designed by the research performs better than the comparative algorithm in terms of search speed performance, and can be effectively applied to high-dimensional data image retrieval. At the same time, the ASDPQ algorithm achieves a good balance between recall rate and computing saving, which not only improves the search accuracy, but also reduces the computing cost.

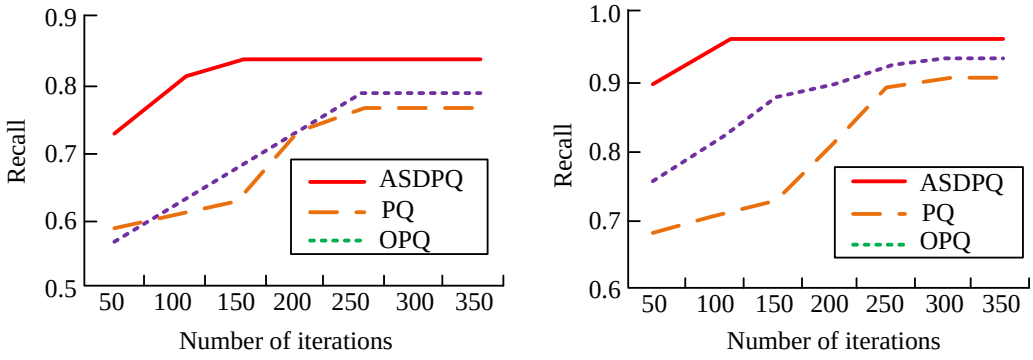

Figure 7: Comparison results of different algorithms.

Table 3: Search time and acceleration ratio test results of different algorithms on SIFT and GIST datasets.

Data set	SIFT			GIST		
Methods	PQ	OPQ	ASDPQ	PQ	OPQ	ASDPQ
Distance calculation table/ms	0.355	0.423	0.555	0.574	1.477	1.794
Distance query table/ms	3.625	2.536	2.580	5.02	3.66	3.58
Search time/ms	3.981	3.959	3.135	5.594	5.437	5.374
Compared to the acceleration ratio of PQ		0.55%	21.3%		2.8%	3.9%

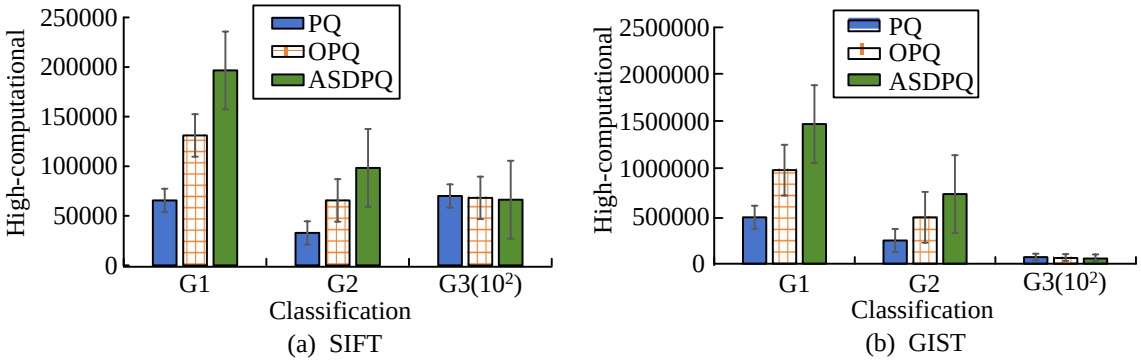

Figure 8: Comparison of computational savings between different algorithms.

Table 4: Comparison results of ASDPQ algorithm and mainstream high-dimensional data retrieval algorithms.

Data set/algorithms	SIFT			GIST		
	Precision	F1 score	mAP	Precision	F1 score	mAP
ASDPQ	0.84	0.82	0.80	0.97	0.96	0.95
DQN	0.78	0.76	0.75	0.94	0.93	0.92
GCNH	0.80	0.78	0.77	0.95	0.94	0.93
SSDPQ	0.81	0.79	0.78	0.96	0.95	0.94

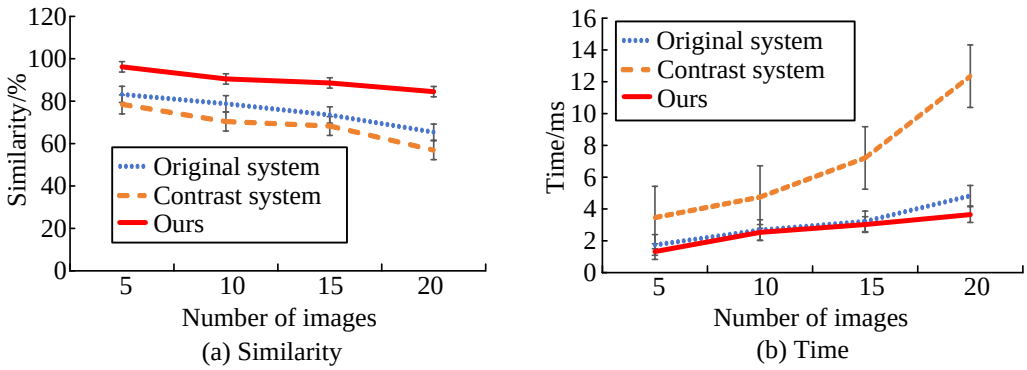

Figure 9: Comparison results of image similarity and retrieval time returned by each system.

Finally, the study selected the Deep Quantization Network (DQN), Graph Convolutional Network-Based Hashing Algorithm (GCNH), and Self-Supervised Deep Product Quantization (SSDPQ) as comparison algorithms. The accuracy, mAP, and F1 score were used as evaluation metrics to test the ASDPQ algorithm. The test results are presented in Table 4.

As shown in Table 4, the ASDPQ algorithm outperformed the latest technologies on both the SIFT and GIST datasets in terms of accuracy, F1 score, and mAP. The performance advantage was more pronounced on the SIFT dataset, indicating its better adaptability to high-dimensional data. It also excelled on the GIST dataset, demonstrating its stability and efficiency in searching for data with varying feature dimensions. Overall, the ASDPQ algorithm demonstrates superior performance.

3.3 Application performance testing of high-dimensional data image retrieval model

After completing the algorithm performance testing, to further verify the practicality and reliability of the proposed high-dimensional data image retrieval model based on ASDPQ algorithm, Matlab was used to integrate the proposed model into a certain image retrieval system for system integration testing. In system integration testing, a high-dimensional data image retrieval model based on the ASDPQ algorithm is integrated into an image retrieval system. For comparison, two baseline systems are developed: the Original System, based on the traditional PQ algorithm using Faiss library implementation, and the Contrast System, based on the OPQ algorithm also using Faiss implementation. The PQ algorithm had 128 sub-spaces with 256 cluster centers each, and the OPQ algorithm used the same dimensions and clusters with extra optimization. Hyperparameters of all systems were optimized via grid search for fair comparisons. The retrieval system was designed with seven key components. During testing, the Oxford University building dataset (5,062 detailed photos of Oxford buildings with diverse perspectives, lighting, weather, and seasonal changes, along with rich annotations) was chosen, offering valuable, high - quality data for system development and improvement. The similarity and retrieval time of the returned images between the statistical research model and the original system, as well as the contrast system, are shown in Figure 9.

Table 5: Standard deviation data of different systems in multiple runs.

Number of images	System	Simlarity Std.Dev	Time Std.Dev(ms)
5	Original system	2.1	0.5
5	Contrast system	2.3	0.6
5	Ours	1.8	0.4
10	Original system	2.4	0.7
10	Contrast system	2.6	0.8
10	Ours	2.0	0.5
15	Original system	2.7	0.9
15	Contrast system	2.9	1.0
15	Ours	2.2	0.6
20	Original system	3.0	1.1
20	Contrast system	3.2	1.2
20	Ours	2.4	0.7

Table 6: mAP and precision@k results of different systems

System	mAP	Precision@5	Precision@10
Original system	0.72	0.68	0.65
Contrast system	0.75	0.70	0.68
Ours	0.80	0.75	0.72

(a) Original drawing (b) Research system (c) Original system (d) Contrast system

Figure 10: Visual display of retrieval results for each system.

Figures 9 (a) and 9 (b) show the similarity and retrieval time calculation results of the images returned by each system. From Figure 9 (a), overall, as the number of images increased, the similarity of the images returned by each system gradually decreased. Among them, the image similarity of the research model was the highest, reaching over 80%, which was 19.6% higher than the average similarity of the original system. From Figure 9 (b), overall, as the number of images increased, the retrieval time of each system returning images gradually increased, but the changes in the research system were relatively stable. Among them, the retrieval time of the research

model was the shortest, with an average of 2.63ms, which was 15.5% shorter than that of the original system. The research results indicated that the research model performed the best in returning image similarity and retrieval time, achieving the expected results, proving the practicality and reliability of the high-dimensional data image retrieval model based on ASDPQ algorithm. The standard deviation data of different systems in multiple runs are shown in Table 5.

From Table 5, the standard deviation of the similarity and retrieval time of the research model was relatively small, indicating that its performance was more stable. The mAP and precision@k results of different systems are shown in Table 6.

From Table 6, the research model was better than other systems in mAP, Precision@5, and Precision@10 indicators, which further proved its superiority. To visually display the test results, the study visualized the most similar images retrieved by each system, as shown in Figure 10. In this study, the selection of "the most similar image" was based on the cosine similarity index. Specifically, for each query image, the cosine similarity between the query image and all images in the data set was calculated, and the image with the highest similarity score was selected as the "most similar image".

From Figure 10, compared with the original system and the comparison system, the images returned by the research model had higher similarity with the original images in terms of architectural features, environmental features, colour features, etc. The findings denoted the feasibility of the research raised image retrieval model for high dimensional data based on ASDPQ algorithm to be applied on image retrieval system. The cosine similarity scores of the most similar images returned by different systems are shown in Table 7.

From the cosine similarity scores in Table 7, the similarity scores of the most similar images returned by the research model were generally higher than those of the original system and the comparison system, which further proved its effectiveness.

To assess the ASDPQ algorithm's practicality and scalability, this study applied it to a large e-commerce platform with a vast image database of over 5 million product images across categories like clothing and electronics. Each image has high-dimensional features (up to 512+ dimensions). The ASDPQ algorithm was implemented to improve user experience in searching for product images. GCNH and SSDPQ algorithms were selected as comparison algorithms, and the results are presented in Table 8.

Table 7: Cosine similarity scores of the most similar images returned by different systems.

Query image	Original system (Cosine similarity)	Comparison system(Cosine Similarity)	Ours(Cosine similarity)
Image1	0.75	0.78	0.82
Image2	0.70	0.73	0.79
Image3	0.72	0.75	0.80
Image4	0.68	0.71	0.77
Image5	0.73	0.76	0.81

Table 8: Application effect of ASDPQ algorithm on large-scale data sets.

Algorithm	Retrieval time (ms)	Recall	mAP
ASDPQ	8.5	0.88	0.86
SSDPQ	12.3	0.82	0.79
GCNH	10.1	0.85	0.83

As Table 8 indicates, the ASDPQ algorithm outperformed comparison algorithms in large-scale e-commerce image databases. With a 8.5ms retrieval time, it responded quickly to user image searches. Its recall rate of 0.88 and mAP of 0.86 were higher, showing more accurate product image retrieval. This is thanks to its adaptive mechanism that handles high-dimensional data, reduces complexity, and enhances accuracy, demonstrating excellent applicability and performance.

4 Discussion

This study proposed the ASDPQ algorithm for high-dimensional data image retrieval. Compared with traditional PQ and OPQ algorithms, ASDPQ had significant advantages in search accuracy and speed [22]. The ASDPQ algorithm optimized sub-space partitioning by adaptively selecting the number of sub-spaces using quantization error comparison. When processing high-dimensional data, it effectively reduced the number of vector quantum spaces in the dataset, thereby reducing computational complexity and accelerating search speed. On the SIFT and GIST datasets, ASDPQ exhibited the best recall rate, significantly reducing search time and making it highly efficient in processing large-scale high-dimensional data. This algorithm could adaptively adjust the number of sub-spaces and clustering classes based on the characteristics of the dataset, providing high flexibility and performing well on different types of datasets. For the SIFT dataset, selecting fewer sub-spaces and more clusters could improve recall and search speed. For the GIST dataset, adjusting parameters to better fit global descriptive features can achieve excellent performance. In addition, ASDPQ performed well in reducing computational load, saving 4.54% of addition operations on the SIFT dataset and 6.96% on the GIST dataset compared to the PQ algorithm. This allowed for more efficient use of computing resources when processing large-scale datasets, thereby improving retrieval efficiency [23].

In summary, the ASDPQ algorithm efficiently processes and rapidly retrieves high-dimensional data by adaptively selecting subspace dimensions and cluster categories. Its advantages are evident in handling various

types of datasets, significantly reducing computational costs. Therefore, the ASDPQ algorithm has broad application prospects in the field of high-dimensional data image retrieval.

5 Conclusion

With the growth of computer technology and Internet applications, retrieving high - dimensional data images quickly and accurately is urgent. To address PQ algorithm's low accuracy and long search time on large-scale datasets, this paper proposed the ASDPQ algorithm and an image retrieval model for high-dimensional data. On the SIFT and GIST datasets, ASDPQ had the highest recall rates (0.84 and 0.97), improving by 7.7% and 3.2% over OPQ. Its search times (3.135ms and 5.374ms) were better than OPQ and PQ. Compared to PQ, ASDPQ saved 4.54% and 6.96% in addition computation. The research model's image similarity was over 80%, 19.6% higher than the original system's average. In addition, the retrieval time of the system was also optimal. The research method proved effective and reliable for high-dimensional data image retrieval. However, the search algorithm design has limitations, not fully considering the correlation between multiple query vectors in a short time, which may involve temporal, spatial, or semantic continuity. Ignoring this can reduce retrieval efficiency or accuracy. Future plans include introducing a dynamic caching system for frequently-occurring query results and using incremental update technology to improve retrieval efficiency for similar queries. Furthermore, as the data dimensions continue to increase and multimodal data (such as the joint retrieval of images, text, and audio) becomes more prevalent, the ASDPQ algorithm faces new challenges. While current algorithms demonstrate certain performance advantages when handling higher-dimensional data, the computational complexity and storage requirements may significantly increase. Future research could explore more efficient dimensionality reduction methods, combined with the ASDPQ algorithm's adaptive subspace dimension selection mechanism, to better handle ultra-high-dimensional data. For multimodal data retrieval, it is essential to study how to effectively integrate the features of different modalities and improve the ASDPQ algorithm to adapt to the characteristics of the multimodal feature space, thereby achieving efficient cross-modal retrieval.

References

[1] Kanchan Rajwar, Kusum Deep, and Swagatam Das. An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. Artificial Intelligence Review, 56(11):13187-13257, 2023. https://doi.org/10.1007/s10462-023-10470-y

[2] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR), 54(4):1-34, 2021. https://doi.org/10.1145/3447582

[3] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte Carlo tree search: A review of recent modifications and applications. Artificial Intelligence Review, 56(3):2497-2562, 2023. https://doi.org/10.1007/s10462-022-10228-y

[4] Pavel Trojovský, and Mohammad Dehghani. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. Sensors, 22(3):855, 2022. https://doi.org/10.3390/s22030855

[5] Farhad Soleimanian Gharehchopogh, Mohammad Namazi, Laya Ebrahimi, and Benyamin Abdollahzadeh. Advances in sparrow search algorithm: a comprehensive survey. Archives of Computational Methods in Engineering, 30(1):427-455, 2023. https://doi.org/10.1007/s11831-022-09804-w

[6] Chenggang Yan, Biao Gong, Yuxuan Wei, and Yue Gao. Deep multi-view enhancement hashing for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(4):1445-1451, 2020. https://doi.org/10.1109/TPAMI.2020.2975798

[7] Qihua Feng, Peiya Li, Zhixun Lu, Chaozhuo Li, Zefan Wang, Zhiquan Liu, Chunhui Duan, Feiran Huang, Jian Weng, and Philip S. Yu. Evit: Privacy-preserving image retrieval via encrypted vision transformer in cloud computing. IEEE Transactions on Circuits and Systems for Video Technology, 34(8):7467-7483, 2024. https://doi.org/10.1109/TCSVT.2024.3370668

[8] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3):535-547, 2019. https://doi.org/10.1109/TBDATA.2019.2921572

[9] Yan Feng, Bin Chen, Tao Dai, and Shu-Tao Xia. Adversarial attack on deep product quantization network for image retrieval. Proceedings of the AAAI conference on Artificial Intelligence, 34(07):10786-10793, 2020. https://doi.org/10.1609/aaai.v34i07.6708

[10] Zeyu Wang, Qitong Wang, Xiaoxing Cheng, Peng Wang, Themis Palpanas, and Wei Wang. Steiner-hardness: A query hardness measure for graph-based ANN indexes. Proceedings of the VLDB Endowment, 17(13):4668-4682, 2024. https://doi.org/10.14778/3704965.3704974

[11] Mustafa A. Alawsi, Salah L. Zubaidi, Nadhir Al-Ansari, Hussein Al-Bugharbee, and Hussein Mohammed Ridha. Tuning ANN hyperparameters by CPSOCGSA, MPA, and SMA for short-term SPI drought forecasting. Atmosphere, 13(9):1436, 2022. https://doi.org/10.3390/atmos13091436

[12] Tan Yu, Jingjing Meng, Chen Fang, Hailin Jin, and Junsong Yuan. Product quantization network for fast visual search. International Journal of Computer Vision, 128(8):2325-2343, 2020. https://doi.org/10.1007/s11263-020-01326-x

[13] Defu Lian, Xing Xie, Enhong Chen, and Hui Xiong. Product quantized collaborative filtering. IEEE

Transactions on Knowledge and Data Engineering, 33(9):3284-3296, 2020. https://doi.org/10.1109/TKDE.2020.2964232

[14] Haowen Zhang, Yabo Dong, Jing Li, and Duanqing Xu. Dynamic time warping under product quantization, with applications to time-series data similarity search. IEEE Internet of Things Journal, 9(14):11814-11826, 2021. https://doi.org/10.1109/JIOT.2021.3132017

[15] Rim Fakhfakh, Ghada Feki, and Chokri Ben Amar. Personalized open-vocabulary image retrieval via semantic and SRSiM-based social features. Informatica, 49(9):34-47, 2025. https://doi.org/10.31449/inf.v49i9.5684

[16] Lingchen Gu, Ju Liu, Xiaoxi Liu, Wenbo Wan, and Jiande Sun. Entropy-optimized deep weighted product quantization for image retrieval. IEEE Transactions on Image Processing, 33(1):1162-1174, 2024. https://doi.org/10.1109/TIP.2024.3359066

[17] Xusheng Zhao, and Jinglei Liu. Joint contrastive self-supervised learning and weak-orthogonal product quantization for fast image retrieval. Knowledge-Based Systems, 304(1):112541-112553, 2024. https://doi.org/10.1016/j.knosys.2024.112541

[18] Tahar Gherbi, Ahmed Zeggari, Zianou Ahmed Seghir, and Fella Hachouf. Entropy-guided assessment of image retrieval systems: Advancing grouped precision as an evaluation measure for relevant retrievability. Informatica, 47(7):159-172, 2023. https://doi.org/10.31449/inf.v47i7.4661

[19] Runhui Wang, and Dong Deng. DeltaPQ: lossless product quantization code compression for high dimensional similarity search. Proceedings of the VLDB Endowment, 13(13):3603-3616, 2020. https://doi.org/10.14778/3424573.3424580

[20] Jiaji Wang, Shuihua, Wang, and Yudong, Zhang. Deep learning on medical image analysis. CAAI Transactions on Intelligence Technology, 10(1):1-35, 2025. https://doi.org/10.1049/cit2.12356

[21] Ze-bin Wu, and Jun-qing Yu. Vector quantization: A review. Frontiers of Information Technology & Electronic Engineering, 20(4):507-524, 2019. https://doi.org/10.1631/FITEE.1700833

[22] Xinyan Dai, Xiao Yan, Kelvin K. W. Ng, Jiu Liu, and James Cheng. Norm-explicit quantization: Improving vector quantization for maximum inner product search. Proceedings of the AAAI Conference on Artificial Intelligence. 34(01):51-58, 2020. https://doi.org/10.1609/aaai.v34i01.5333

[23] Kavita Bhosle, and Vijaya Musande. Evaluation of deep learning CNN model for recognition of devanagari digit. Artificial Intelligence and Applications, 1(2):114-119, 2023. https://doi.org/10.47852/bonviewAIA3202441