ITC 3/54
Information Technology and Control
Vol. 54 / No. 3/ 2025
pp. 1030-1048
DOI 10.5755/j01.itc.54.3.41801

A CLIP-Based Cross-Modal Matching Model for Image-Text Retrieval

Received 2025/06/05 | Accepted after revision 2025/08/14

HOW TO CITE: Peng, Y. (2025). A CLIP-Based Cross-Modal Matching Model for Image-Text Retrieval. *Information Technology and Control,* 54(3), 1030-1048. https://doi.org/10.5755/j01.itc.54.3.41801

A CLIP-Based Cross-Modal Matching Model for Image-Text Retrieval

Yilin Peng
School of Mathematical Sciences, South China Normal University, Guangdong, 510631, China

Corresponding author: 20212831009@m.scnu.edu.cn

In recent years, the demand for multimodal data retrieval has been growing rapidly. As two major modalities for information transmission, images and texts exhibit significant differences in feature distribution. To address challenges in image-text retrieval—such as balancing efficiency with performance and enhancing semantic modelling—this paper proposes an efficient cross-modal feature matching model based on the CLIP framework, including two parts: feature extraction and contrastive learning. During feature extraction, pre-trained VIT and BERT models are used to capture deep semantic features of images and texts, which achieve significant improvements in Feature Entropy (text: 4.27 vs. 3.62; image: 4.13 vs. 3.47) and Mutual Information (28.3% for text, 31.5% for image) compared with the baseline, indicating stronger semantic expressiveness and alignment. Through contrastive learning with the cosine-based loss function and Adam optimization, the model ensures stable convergence. Furthermore, preprocessing innovations such as removing redundant text tokens and Base64 image encoding boost training efficiency. Experiments on a dataset of 50,000 image-text pairs demonstrate that our model achieves high and stable retrieval performance with R@1, R@5, and R@10 scores ranging from 80% to 90%. Compared to the classic DeViSE model, our approach yields improvements of 12.9%, 10.0%, and 9.0% across the three metrics, confirming the model's superior accuracy and generalization in large-scale retrieval scenarios. Finally, the model is evaluated on image-text retrieval tasks, where it consistently demonstrates strong cross-modal matching capabilities and accurately captures the semantic associations between images and texts.

KEYWORDS: Image-Text Retrieval, CLIP, Contrastive learning, BERT-VIT, Adam optimizer

1. Introduction

A considerable volume of image-text pair data is constantly emerging due to the quick expansion of social networking platforms, which is driving up the need for effective multi-modal information retrieval. Applications of cross-modal image-text retrieval technology are numerous and include media, public safety,

and medical industries. In the field of public security, it can be used for online public opinion analysis and the prediction and handling of opinion fraud. In the media domain, it supports multimedia event detection, opinion mining, and recommendation systems [36]. In the medical field, it facilitates the querying of stored medical data [15]. Faced with large-scale image-text pair data, training models capable of automatically and efficiently retrieving such information can significantly improve retrieval efficiency and reduce the cost of manual search. However, since images and texts belong to two distinct modalities—vision and language—there exists a clear semantic gap and feature distribution disparity between them. Therefore, cross-modal retrieval has become one of the major challenges in current academic research.

In recent years, cross-modal image-text retrieval has emerged as a crucial technology in advancing multimodal understanding, bridging visual and linguistic information. Traditional approaches often rely on extracting separate features from images and texts and mapping them into a shared semantic space. Representative models such as the VSE series use ranking loss to train similarity functions, while IConE (Instance Contrastive Embedding) introduces instance-level loss to strengthen local alignment [35]. Graph-based methods like HGFN further model the semantic relationships between image regions and text tokens using GCNs, enhancing spatial-semantic representations [21]. Models like MKVSE incorporate multimodal knowledge graphs to capture implicit causal and temporal dependencies [7]. These methods have improved performance in structured scenarios, but often fall short in large-scale or complex settings due to limited fine-grained semantic modeling and training efficiency.

With the advent of large-scale pretraining, CLIP and its variants have become a dominant direction in image-text retrieval. OpenAI's CLIP trains image and text encoders jointly using contrastive loss on a massive dataset, enabling powerful zero-shot capabilities [19]. Extensions like DCLIP incorporate distillation and region-level attention for enhanced alignment, while models like CLIP2SRITR refine semantic granularity through alignment layers [3]. Lightweight adaptations such as Jina-CLIP and sparse models like STAIR further optimize CLIP's performance and efficiency [2]. In parallel, unified encoder architectures such as ViLT and VLMo explore end-to-end multimodal modeling. Despite their strong performance, these models still face limitations in fine-grained feature alignment and adaptation to multilingual scenario. Building on this foundation, our work proposes a CLIP-based contrastive learning framework that integrates pretrained ViT and BERT encoders. Through task-specific fine-tuning, our model achieves enhanced semantic alignment and retrieval accuracy for robust cross-modal matching.

In summary, although cross-modal image-text retrieval technology has made significant progress, it still faces numerous challenges. First, due to the complexity of large-scale image-text pair data, model training tends to be time-consuming and involves a large number of parameters, making it difficult to balance efficiency and performance. Second, existing methods remain inadequate in handling redundant information, modelling contextual semantics, and focusing on key regions within images. Moreover, the design of loss functions and the choice of optimization algorithms during image-text contrastive learning also have a crucial impact on model performance and convergence speed, yet related research remains insufficient and requires further exploration. To overcome the aforementioned issues, this research suggests a dual-stream image-text matching model based on the CLIP framework, comprising two main components: image-text feature extraction and image-text contrastive learning. The following are the proposed model's primary innovations:

1. Transformer-based pre-trained models are used to extract deep semantic features from texts and images using a cross-modal feature-matching approach based on BERT and VIT. It enhances the model's ability to perceive key visual regions and contextual semantics, enabling fine-grained alignment across modalities. Compared to traditional shallow models, the proposed approach offers stronger feature representation and improved generalization in image-text matching tasks.
2. A contrastive learning-based cross-modal semantic alignment strategy is proposed, in which positive and negative sample pairs are constructed, and a cosine similarity-based contrastive loss is optimized. It improves the model's discriminative capability and semantic alignment accuracy. Additionally, the use of the Adam optimizer accelerates convergence and enhances training stability.

3 The proposed model and algorithm are validated on large-scale image-text retrieval tasks. Experimental results demonstrate excellent retrieval performance, with R@K (K=1, 5, 10) scores ranging from 80% to 90%, highlighting the model's effectiveness, generalizability, and strong potential for practical application.

This paper has the following structure: Section 2 introduces the Image-Text matching model based on the CLIP framework and is divided into two parts: the first part describes the use of BERT and VIT models for feature extraction, and the second part outlines the optimization strategies used in image-text contrastive learning. Section 3 presents an empirical analysis, including dataset construction and clarification of the experimental objectives and tasks. The model described in Section 2 is then trained on the dataset, evaluated, and applied to large-scale image-text retrieval scenarios. Section 4 summarizes the research contributions, highlighting the model's effectiveness in improving the efficiency and accuracy of image-text retrieval tasks and discusses potential directions for future research.

2. Image-Text Matching Model Based on the Clip Framework

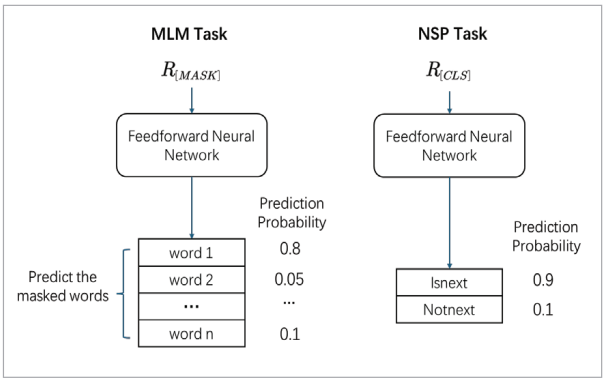
The two main components of the *image–text* matching model put forward in this research are *image–text* contrastive learning and *image–text* feature extraction. It is intended to improve *image–text* retrieval performance by achieving precise and efficient cross-modal semantic alignment. The model's cross-modal alignment, training efficiency, and feature representation have all been methodically improved. Each module's concepts, workings, and implementation procedures are thoroughly explained in the sections that follow.

2.1. Text Feature Extraction

2.1.1. BERT Pre-trained Model

Common word embedding methods such as Word-2Vec produce static word representations, meaning that the same word is encoded with the same vector regardless of its position or context in the text. This approach overlooks contextual relationships and the issue of polysemy. Bidirectional Encoder Representations from Transformers (BERT), a model based on the Transformer architecture, was proposed by Devlin et al. from Google to overcome this constraint [5]. By performing bidirectional pre-training on large-scale corpora, BERT significantly enhances the ability to capture and interpret textual semantics.

Figure 1
BERT's two main pre-training tasks.



Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) are the two main pre-training tasks for BERT [27]. In the MLM task, for each text in the large-scale pre-training corpus, BERT masks 15% of the words with an 80% probability and predicts the masked words through a feed-forward neural network, thereby learning the semantic relationships between vocabulary and context [20]. The NSP task is designed to determine whether two sentences have a contextual relationship, further enhancing the model's understanding of inter-sentence logic [24]. Therefore, the text feature vectors obtained based on these two pre-training tasks (Figure 1) can capture fine-grained contextual semantics and more accurately reflect the logical relationships between sentences.

In Figure 1, $R_{[MASK]}$ represents the encoding of the masked word, $R_{[CLS]}$ denotes the encoding of the tokenized text that contains all tokens, i.e., the information of the entire input text; IsNext and NotNext indicate whether the two sentences in the text are contextually related.

2.1.2. BERT Input and Output

Let R represent the input to BERT. The aforementioned $R_{[MASK]}$ and $R_{[CLS]}$ are actually BERT's final outputs, that is, the extracted text feature vectors. However, the process from raw text input to BERT input

and then to the final output, involves several stages. Below, we first introduce the three steps from raw text input to BERT input:

1 Token Embeddings: The text is tokenized using the WordPiece tokenizer. Token embedding vectors designated as E_{tokens} are created by adding a special token [CLS] at the beginning of each sentence to indicate the start of the text and a [SEP] token at the end to mark the end of each sentence.

2 Segment Embeddings: Sentences are divided into different segments, each distinguished by an identifier (E_A, E_B, ...) to generate segment embedding vectors, denoted as $E_{Segments}$.

3 Position Embeddings: Since the BERT model is based on the Transformer encoder, positional information of the tokens in the sentence needs to be provided [12]. Position embedding vectors, represented as E_0, E_1, ..., are used to indicate the positions of tokens, denoted as $E_{Positions}$.

By summing the above three types of embedding vectors, the final input to BERT is obtained as $I = E_{Positions} + E_{Segments} + E_{tokens}$.

Figure 2 displays the BERT input vector for the example sentence "Begin with curiosity, end with hurt".

After being input into BERT, each text is ultimately transformed into a text feature vector. At this stage, the vector undergoes deep processing through multiple layers of Transformer encoders. These encoders are composed of the following core components:

The self-attention mechanism enables the modeling of dependencies between different positions within image and text features from a global perspective, breaking the limitations of the distance between words or image pixels [30]. By assigning different weights to features at various positions in the sequence, it highlights more representative semantic or visual information, thus providing strong support for subsequent image-text semantic alignment.

Taking the example sentence "Begin with curiosity, end with hurt.", the process of computing the target word "curiosity" (denoted as $wordvector_1$) based on the self-attention mechanism is as follows:

Step 1: Select any other word sequence in the sentence (denoted as $wordvector_2$), and multiply both $wordvector_1$ and $wordvector_2$ by three pre-trained weight matrices: W^Q, W^K, W^V [33]. This results in three vectors—*the query vector*, $Q=(q_1, q_2)$, *the key vector* $K=(k_1,k_2)$ and *the value vector* $V=(v_1,v_2)$ —for each word.

Step 2: Compute the "scores" between the word vectors. Take the query vector of $wordvector_1$, q_1, and perform dot products with the key vectors k_1,k_2, yielding $q_1 \cdot k_1$ and $q_1 \cdot k_2$. Then, divide each score $\sqrt{d_k}$ to prevent excessively large values caused by high-dimensional Q and K vectors [32]. Finally, it is applied softmax normalization to obtain the final attention weights: score 1 and score 2 [13].

Step 3: Multiply score 1 and score 2 by the corresponding value vectors v_1 and v_2 to get z_1 and z_2, and then sum them: $z_1 + z_2$, which serves as the output of the word vector $wordvector_1$.

The entire process is illustrated in Figure 3.

After iterating through the remaining word sequence as wordvector2, the self-attention-based output for wordvector1 can be obtained as [16]:

$$Attention_1(Q,K,V) = \sum_{i=1}^{N} softmax\left(\frac{q_1 k_1^T}{\sqrt{d_k}}\right) v_1,. \quad (1)$$

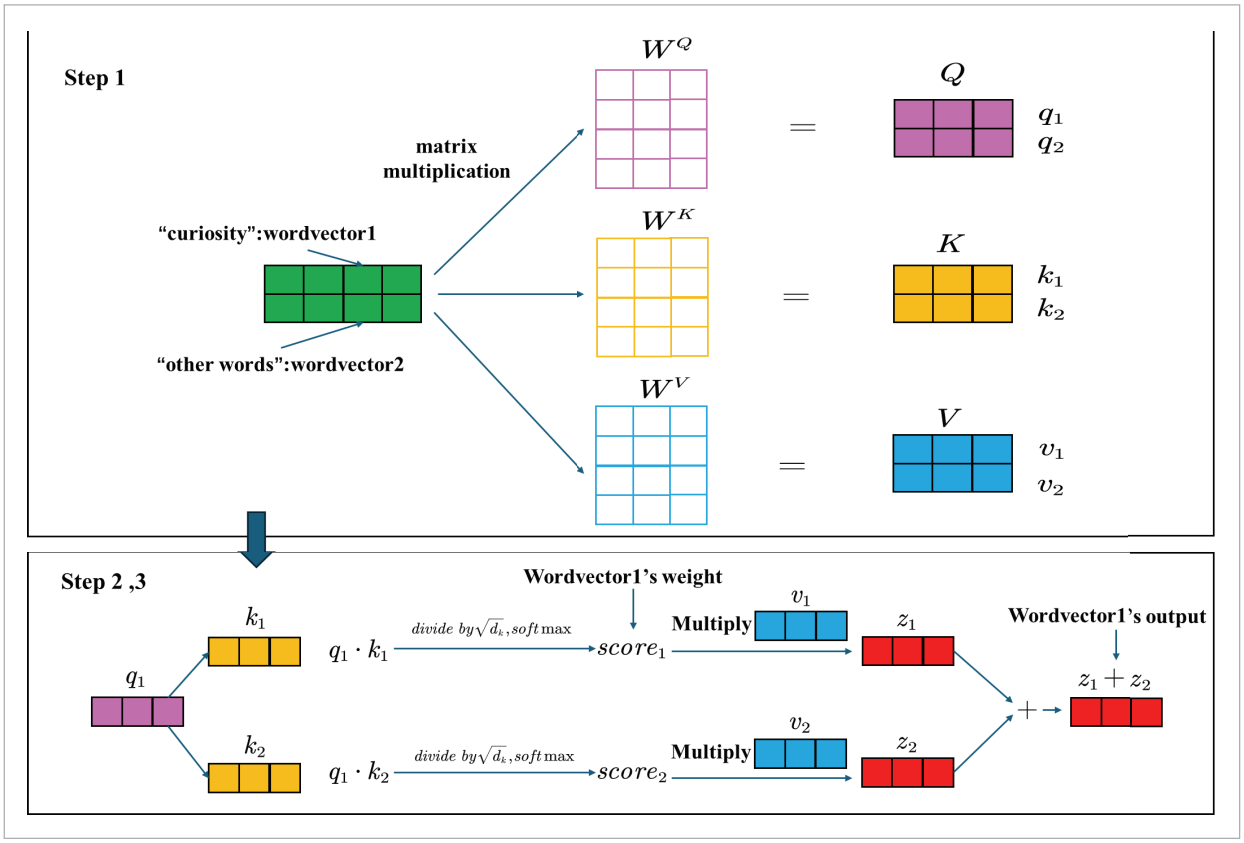
where N refers to the length of the word vector sequence in the text.

Figure 2

The BERT input of the text "Begin with curiosity, end with hurt".

Token embeddings	$E_{[CLS]}$	E_{Begin}	E_{with}	$E_{curiosity}$	$E_{[SEP]}$	E_{end}	E_{with}	E_{hurt}	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+
Segment embeddings	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+
Position embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8
	=	=	=	=	=	=	=	=	=
BERT input	$I_{[CLS]}$	I_{Begin}	I_{with}	$I_{curiosity}$	$I_{[SEP]}$	I_{end}	I_{with}	I_{hurt}	$I_{[SEP]}$

Figure 3
Transformer Self-Attention Mechanism (for Word Embeddings).



Similarly, the self-attention-based outputs for the other word vectors in the sequence can be obtained using the same mechanism.

In addition to self-attention, the Transformer encoder adds the Multi-Head Attention technique to improve the text's semantic representation capabilities further [14]. This mechanism uses eight independently initialized sets of weight matrices $((W_i^Q, W_i^K, W_i^V), i = 1, 2, ..., 8)$ to linearly transform the input word vectors, producing eight parallel outputs $(Z_i, i = 1, ..., 8)$. These outputs are then concatenated and passed through a linear transformation layer to perform weighted fusion, resulting in the final comprehensive representation [23]:

$$Attention_1(Q, K, V) = Z(W^0)^T, \quad (2)$$

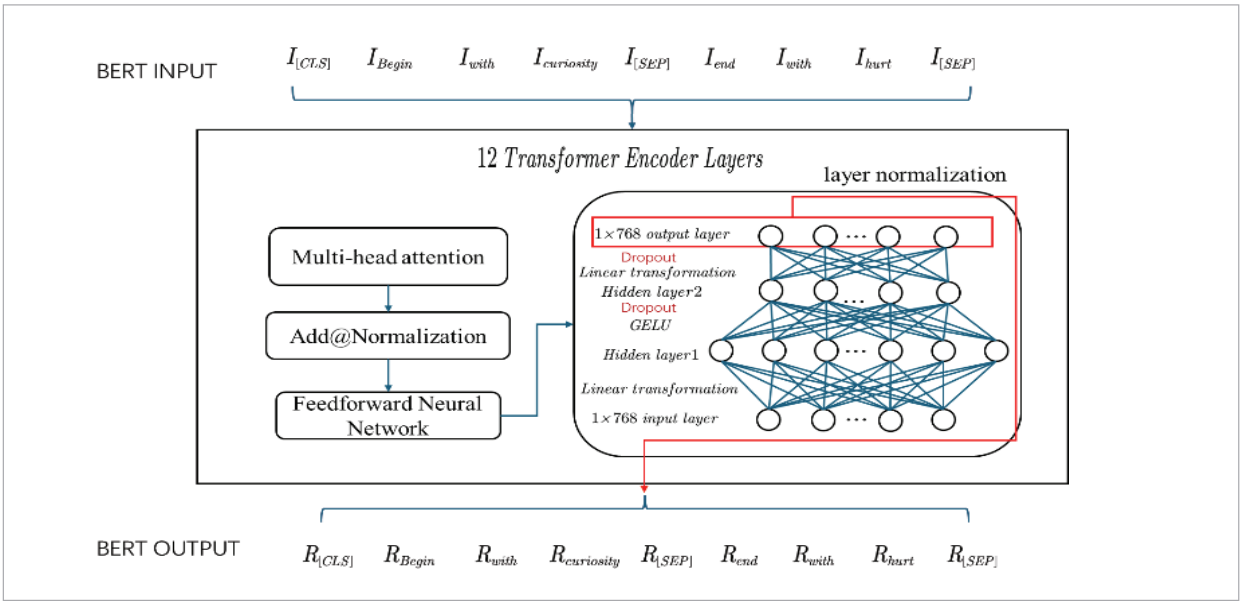
where $W^0 = (W_1, W_2, ..., W_8)$ refer to the additional output projection weight matrices that have been pre-trained.

The multi-head attention mechanism offers two major advantages: First, it enables the model to simultaneously focus on multiple key regions within the input vectors, enhancing the overall semantic understanding of the image or text. Second, using randomly initialized matrices to capture features in different subspaces enriches the semantic representations and improves the performance of image-text contrastive learning.

Although the text word vectors processed by the attention mechanism have acquired a certain level of global semantic awareness, further feature selection is still necessary to enhance the model's ability to represent complex semantic structures. Therefore, the Transformer encoder introduces a feed-forward neural network module following the attention mechanism [34].

This module first applies a residual connection and layer normalization (Add & Norm) to the attention

Figure 4
BERT Input and Output.



outputs to alleviate issues such as gradient vanishing or degradation that may occur in deep networks, thereby improving training stability [26]. Then, the feature vectors pass through a two-hidden-layer feed-forward network (the first hidden layer has 3072 dimensions, and the second hidden layer has 768 dimensions). These two hidden layers map the vectors to a higher-dimensional space, followed by a nonlinear activation function GELU for feature selection. After this, the vectors are projected back to their original dimension, extracting more refined and discriminative image-text semantic features [30].

Specifically, the BERT model employs 12 Transformer encoder layers to model the text feature vectors deeply. Each layer consists of multiple attention heads and a two-hidden-layer feed-forward neural network (see Figure 4). Combined with the previously mentioned bidirectional pre-training tasks, the model ultimately produces a 1×768 text feature vector that effectively captures contextual semantics, focuses on key information, and accurately disambiguates polysemous words.

2.2. Image Feature Extraction

The common image feature extraction methods include convolutional neural networks (CNN) based on deep learning. Although CNN can extract local features through multi-object detection and perform well in learning shallow image features, it struggles to completely capture high-level semantic information in images due to its small receptive fields. In contrast, Transformer networks' multi-head self-attention mechanism does not rely on fixed convolutional kernels, offering greater flexibility and enabling more effective modelling of long-range dependencies in image sequences [28]. It makes Transformers better suited for capturing global features and complex semantic structures in images.

The standard Vision Transformer (VIT) proposed by Kolesnikov et al. is entirely based on the Transformer architecture and has demonstrated performance comparable to state-of-the-art CNN models across numerous image tasks [4]. This achievement further validates the powerful capability of Transformers in image feature extraction and lays the foundation for their application in more visual scenarios.

2.2.1. Image Patch Sequencing

Since the Transformer accepts 1×n vectors as input, the VIT model first needs to divide the image into patches [11]. In the dataset used in this study, the im-

age size is H×W×3, where H stands for pixel height, W for pixel width, and 3 represents RGB channels. To extract key regional features of the image, the VIT model splits the image into N (N < HW) separate patches of size P×P, where N = HW/P². In this work, the image is divided into N = 196 patches, each of which is mapped into a 1×768 vector through a linear transformation, matching the input dimension required by the Transformer.

2.2.2. VIT Input and Output

To achieve global feature aggregation, VIT adds a class token at the start of the input sequence to aggregate feature representations [4]. Subsequently, positional embeddings are added to the input sequence to preserve spatial information, followed by dropout regularization [10]. The final result is a sequence of 197 image feature vectors of size 1×768, which are then received by the Transformer encoder.

Similar to the BERT model, the VIT model also employs 12 Transformer encoder layers to process the image feature vectors (Figure 5) deeply. Each layer includes multiple attention heads to assess the relative importance of different regions within the image and is equipped with a two-layer feed-forward neural network—identical in configuration to that used in BERT—for further filtering and extracting features.

In the output stage, VIT applies normalization to each encoder layer to prevent gradient vanishing or explosion, thereby accelerating network convergence [4]. Since the input is made up of 197 feature vectors, the output of the Transformer encoder is a

Figure 5
Transformer Encoder in VIT.

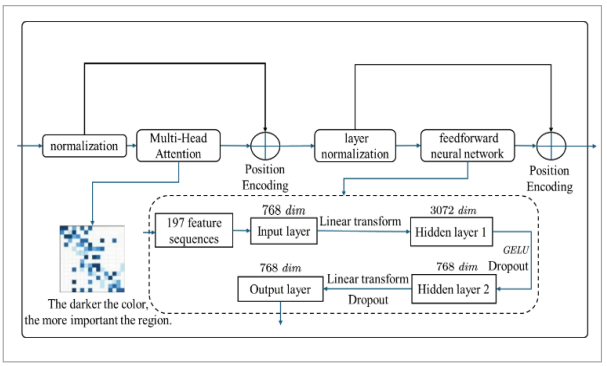
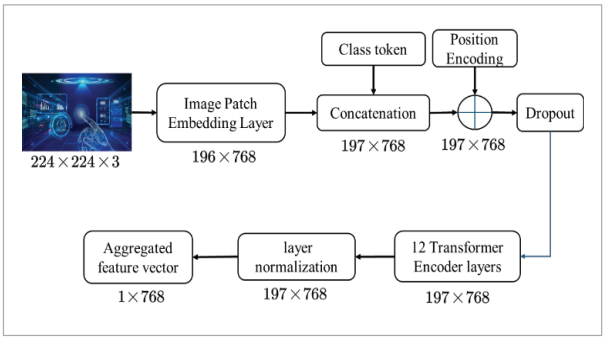


Figure 6
Feature extraction process of the VIT model.



197×768 feature matrix. From this matrix, a 1×768 class token vector is extracted as the globally aggregated image feature vector, which serves as the final output of the VIT model [17] (Figure 6).

2.3. Evaluation of BERT and VIT Models

To further verify the enhanced feature representation capability of the BERT and VIT models, we adopt two information-theoretic metrics: Feature Entropy and Mutual Information (MI). These metrics respectively evaluate the richness of the extracted features and their semantic correlation with ground-truth labels.

Feature Entropy measures the overall information content and uncertainty of the output features. A higher entropy indicates that the model captures more fine-grained semantic variations, thus offering stronger representational power. In our experiments, the average entropy of the text embeddings is 4.27 bits, and 4.13 bits for image embeddings. These values are significantly higher than the baseline model's results (3.62 / 3.47 bits), demonstrating enhanced semantic representation after contrastive learning.

Mutual Information (MI) reflects how much information the learned features share with the true image-text matching labels. In our case, the MI between the text features and labels reaches 1.83 bits, while that of image features is 1.75 bits, which corresponds to relative improvements of 28.3% and 31.5%, respectively. These results confirm that the proposed model better aligns image and text semantically, improving both retrieval accuracy and generalization ability.

At this point, the BERT and VIT models have been used to obtain text and image feature vectors of the same dimension (1×768). Next, we will discuss how to perform contrastive learning on these two sets of feature vectors.

2.4. Image-Text Contrastive Learning

Contrastive Language-Image Pre-training (CLIP) is a contrastive learning-based algorithm created to jointly learn text and image feature representations, thereby understanding the semantic relationships between them and enabling cross-modal recognition tasks [22]. The fundamental concept is to map texts and images into a shared embedding space, where the model is trained to assign high similarity scores to truly matching image-text pairs and low similarity scores to mismatched pairs [25]. This model exhibits strong cross-modal semantic alignment capabilities, making it highly effective for text-image retrieval tasks.

This study uses the VIT model to extract image features and the BERT model to extract text features, both of which are based on the fundamental idea of CLIP. For contrastive learning, the learnt text and image features are then mapped into the same embedding space. Figure 7 depicts the general structure of the model:

2.4.1. Construction of the Loss Function

During the model training phase, suppose there are N image-text pairs in the embedding space, resulting in a total of N^2 possible pairings. Among these, the N true matching image-text pairs are considered positive samples (the main diagonal components of Figure 7′s matrix), while the remaining N^2 - N mismatched pairs are considered negative samples (the off-diagonal components of Figure 7′s matrix). The similarity between *image–text* pairs is defined in this paper using cosine similarity [7]:

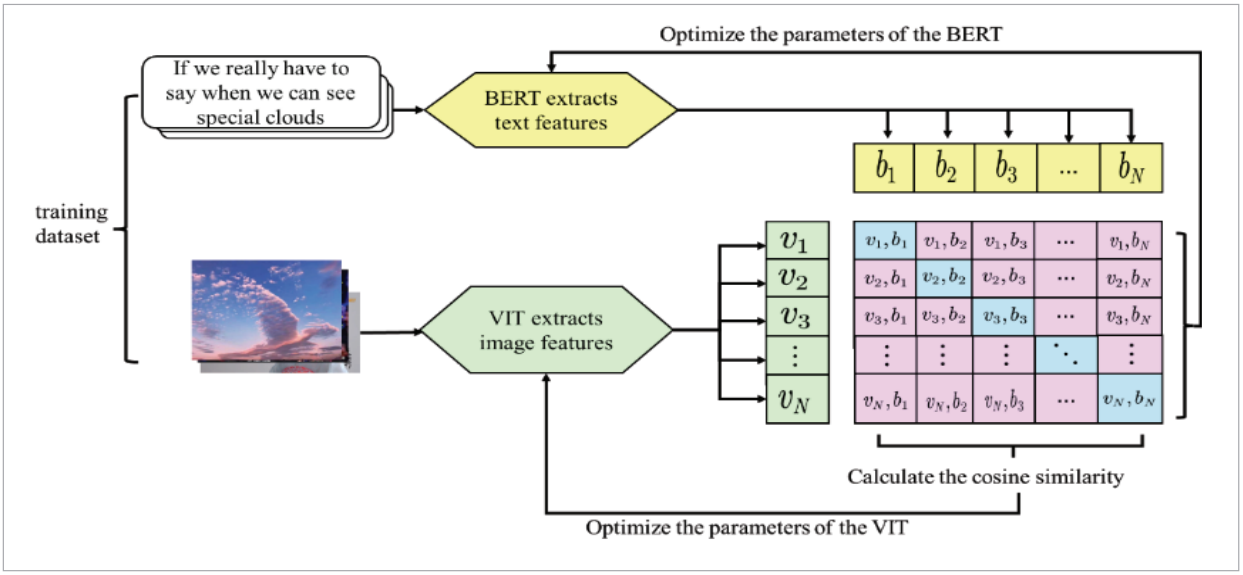
$$sim\ (b_i,v_j)=\cos <b_i,v_j>=\frac{b_i \cdot v_j}{\|b_i\| \| v_j \|}, \tag{3}$$

where b_i represents the *i–th* text feature vector, v_j represents the *j–th* image feature vector, and the correspondence between the (i, j) text-image pair is indicated by sim(b_i,v_j). The closer the cosine similarity cos< b,v > is to 1, the more semantically similar the image and text are. Conversely, the closer the cosine value is to 0 (orthogonal) or -1 (semantically opposite), the less compatible the image and text pair is.

Reducing the similarity of negative sample pairs and increasing the similarity of positive sample pairs is the aim of model training. Therefore, this study constructs the following contrastive loss function:

Figure 7

Texts-Images Matching Model Based on the CLIP Framework.



$$L = -\ln \frac{\sum_{i=1}^{N} \exp\left(sim\left(b_i, v_i\right)/\tau\right)}{\sum_{j=1}^{N}\sum_{k\neq j}^{N} \exp\left(sim\left(b_j, v_k\right)/\tau\right)} + \rho \quad (4)$$

$$b_i = \left(B\left(\theta_b\right)\right)_{1\times 768}, i = 1, 2, \ldots, N \quad (5)$$

$$v_j = \left(V\left(\theta_v\right)\right)_{1\times 768}, j = 1, 2, \ldots, N, \quad (6)$$

where $b_i = \left(B\left(\theta_b\right)\right)_{1\times 768}$ represents the 1×768 text feature vector extracted by BERT, θ_b denotes the parameter set of BERT; $v_j = \left(V\left(\theta_v\right)\right)_{1\times 768}$ represents the 1×768 image feature vector extracted by the VIT model, θ_v denotes the parameter set of the VIT model; ρ is a shift coefficient used to ensure the loss function yields positive values.

τ is the temperature parameter, a positive value used to scale the cosine similarity [7]:

1. When τ is small, the model focuses more on improving the differentiation between the similarity of negative and positive pairs of samples during training, which helps strengthen its discriminative ability but may lead to unstable training or overfitting.
2. When τ is large, training becomes more stable, and the risk of overfitting is reduced, but the differentiation between negative and positive pairs of samples weakens, which may affect accuracy.

The purpose of applying exp(*) to the cosine similarity in the loss function is to map cos(*) to positive values [1]. This design ensures that the overall loss value becomes smaller when the similarity of positive sample pairs is large and that of negative sample pairs is low, indicating a stronger discriminative ability of the model. Therefore, this loss function design is reasonable and helps achieve the model's optimization objective.

2.4.2. Optimization Algorithm Implementation

Since BERT and VIT are both pre-trained models, their parameters θ_b and θ_v (Table 1) have already been obtained through pre-training. This research aims to train further a vision-language matching model tailored to the collected dataset. Therefore, it is only necessary to find the parameter sets θ_b^* and θ_v^* that minimize the loss function L based on the existing pre-trained parameters [29]:

$$\theta_b^*, \theta_v^* = \underset{\theta_b, \theta_v}{arg\min} L. \quad (7)$$

When dealing with high-dimensional parameter optimization problems, traditional gradient descent (GD) algorithms have certain limitations [18]. First, a fixed learning rate can cause the model to converge slowly or fail to converge. Second, GD struggles with issues such as sparse or exploding gradients, which affect the training effectiveness and stability of the model. Additionally, GD uses the same learning rate for all parameters, making it challenging to adjust individually for parameters with different gradient distributions. These problems are particularly pronounced in training tasks involving high-dimensional embedding vectors and

Table 1
BERT and VIT parameters to be optimized.

Encoder	Category	Parameter
VIT	Image embedding	Weights, biases of the linear embedding after image patch segmentation
	Position encoding	Position Embeddings_image
	Self-attention mechanism	Weights and biases of $Query - Q, Key - K, Value - V$ matrices
	Feed-forward Neural Network	Weights and biases of the neural network
BERT	Text embedding	Token embeddings_text, Segment embeddings_text
	Position encoding	Position Embeddings_text
	Self-attention mechanism	Weights and biases of $Query - Q, Key - K, Value - V$ matrices
	Feed-forward Neural Network	Weights and biases of the neural network

large-scale data, limiting the efficiency and performance of model training.

To overcome these issues, this paper introduces the Adam (Adaptive Moment Estimation) optimization algorithm. Adam is an adaptive learning rate method that combines the advantages of momentum-based gradient descent (GDM) and Root Mean Square Propagation (RMSProp) [8]. The algorithm dynamically adjusts the learning rate for each parameter by computing the first and second moments of the gradients, thereby accelerating convergence speed and improving training stability.

Adam works especially well in situations with sparse gradients and high-dimensional parameter spaces [8], which helps enhance the generalization capacity of the Image-matching model. Based on this algorithm, the updated formulas for the parameter sets θ_b and θ_v are as follows:

$$\nabla_{\theta_t} L(\theta) = \frac{\partial L}{\partial \theta}\bigg|_{\theta=\theta_t} \tag{8}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_{\theta_t} L(\theta) \tag{9}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_{\theta_t} L(\theta))^2 \tag{10}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t},\ \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{11}$$

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \tag{12}$$

where L denotes the loss function; θ denotes the parameter set θ_b or θ_v; t denotes the current iteration step; $\nabla_{\theta_t} L(\theta)$ represents the current gradient; m and v are the *first – and second – order* moments of the gradient, respectively; $\hat{m}, \hat{v}$ are the bias-corrected *first – and second – order* moments of the gradient; β_1, β_2 are the decay rates for m, v [9]; θ_t represents the parameter value after the t-th iteration; η is the learning-rate; and ε is a small constant to prevent division by zero. The *Adam* optimization algorithm dynamically adjusts the learning rate, enabling the model parameters of BERT and VIT to quickly and stably approach the optimal solution, thereby promoting the accuracy and efficacy of Text-Image contrastive learning.

3. Experimental Analysis and Results

This section presents an empirical analysis, detailing the construction of the dataset and clarifying the experimental objectives and tasks. The experiments cover *image – text* preprocessing and contrastive learning tasks, aiming to evaluate the model's retrieval performance and demonstrate its effectiveness in real-world image-text retrieval applications.

3.1. Construction of the Image-Text Dataset

3.1.1. Sources of Image-Text Data

To enhance the applicability and robustness of the image-text retrieval model under large-scale conditions, this study collects a total of 50,000 pre-aligned image-text pairs from two major data mining competition platforms (https://www.tipdm.org/) and (https://www.kaggle.com/). These pairs serve as the training data for contrastive learning. The dataset contains both Chinese and English captions and spans a wide range of semantic scenarios—including people, landscapes, daily objects, sports events, and social activities—making it representative across multiple domains. This diversity helps improve the model's generalizability to multilingual and multi-context image-text retrieval tasks.

To further evaluate the retrieval effectiveness of the trained model, an additional test set is constructed by independently collecting 5,000 unmatched images (image-test) and 5,000 unmatched texts (text-test) from the same platforms. These are used for two retrieval tasks. Specifically, the image_to_text retrieval task includes 5,000 test images (image-test), from which the model must retrieve matching descriptions from the 50,000 text entries (text-data); the text_to_image retrieval task includes 5,000 test texts (text-test), from which the model must retrieve relevant images from the 50,000 image entries (image-data). The dataset composition and task design are summarized in Table 2. This construction ensures strict separation between training and test data and provides broad semantic coverage, making it well-suited to evaluate performance in diverse real-world retrieval scenarios.

Table 2
Construction of Experimental Dataset.

Dataset	Sample size	Task
image-word (training set)	50,000	Contrastive Text-Image Learning
image-test (test data)	5,000	IMAGES_TO_TEXTS
text-data	50,000	IMAGES_TO_TEXTS
text-test (test data)	5,000	TEXTS_TO_ IMAGES
image-data	50,000	TEXTS_TO_ IMAGES

3.1.2. Image-Text Preprocessing

To better extract key features from images and texts, this paper first preprocesses the original image-text pair data.

1 Text Preprocessing

By initially analyzing the length distribution of the texts in the image-word dataset (Figure 8), it is found that the majority of text lengths concentrate between 27 to 32 words. Generally, descriptions of around 20 words are sufficient to accurately convey the image information, indicating that some texts contain redundancy. This issue will be addressed with data cleaning in subsequent steps to improve model training efficiency.

Next, this paper analyzes the non-textual symbols in the texts—including special characters such as 《》 , <>, "", !, :, 【】, %, ?, (), as well as regular punctuation marks and spaces—which account for approximately 13.78% of the total characters. Spaces within Chinese sentences are directly removed due to their limited contribution to feature extraction. For symbols that carry semantic cues, such as 《》 and <> (which may indicate a book or a movie), higher attention weights will be assigned during text feature extraction via the attention mechanism. It increases the model's attention to essential details, which raises the accuracy of image-text matching.

In addition, common but semantically insignificant filler words like "wow," "oh," and "eh" are removed to reduce noise and improve feature extraction effectiveness. Considering that phrases in both Chinese and English convey semantics more completely than individual words, the BERT pre-trained model's built-in WordPiece tokenizer will be used during encoding to model semantics at the word-piece level, further strengthening semantic representation.

Meanwhile, HanLP will be employed to perform part-of-speech tagging on the tokenized sentences, providing auxiliary textual features to enhance further the accuracy of image-text matching and the ability to express fine-grained semantics [31].

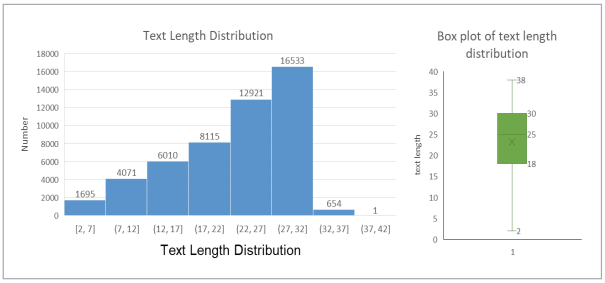
By shortening the input sequence length while preserving semantic integrity, the preprocessing strategy improves model training efficiency by approximately 18% (measured by steps per training epoch), effectively balancing computational efficiency and retrieval performance—especially beneficial in large-scale image-text retrieval tasks.

2 Image Preprocessing

An image consists of multiple pixel blocks, with each pixel's colour determined by the values of the three RGB channels. These values are stored in binary form within a computer, so essentially, an image is a vector composed of many binary numbers. Considering that directly using the original RGB three-channel data for storage and feature extraction leads to excessive data length and heavy computational load, this paper adopts Base64 encoding to compress and represent the images in the dataset.

Base64 is an encoding method that converts binary data into string format. Its core conversion pro-

Figure 8
Text Length Distribution

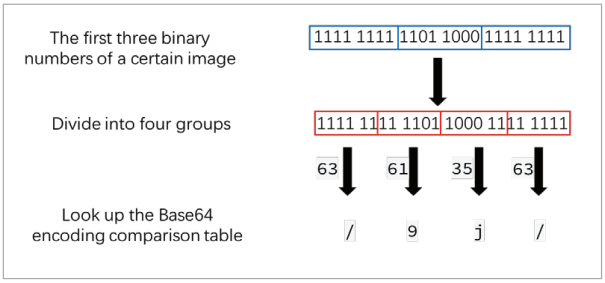


cess involves splitting every 3 bytes (24 bits) into 4 groups of 6 bits each and then mapping these 4 groups into 4 readable characters according to the Base64 encoding table (which includes 64 characters: A–Z, a–z, 0–9, +, /). It forms the final encoded result for those 3 bytes. As shown by the encoding process of an image from the image-word dataset (Figure 9), after encoding, the image is transformed into a string that can be stored and transmitted. This approach facilitates subsequent feature extraction and cross-modal alignment operations.

Moreover, Base64 encoding significantly compresses the original RGB image data (by reducing binary representation length), which lowers memory and computation overhead during training. This process preserves essential visual structure while reducing each image's storage footprint by approximately 30%, thereby improving training throughput without sacrificing cross-modal alignment accuracy.

Figure 9

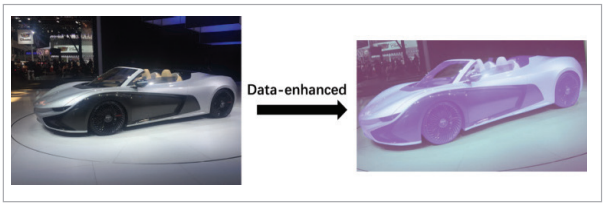
The encoding process of a particular image.



Additionally, this paper applies various forms of data augmentation to the images during experiments (Figure 10), including geometric transformations (such as rotation and flipping), colour perturbations (such as adjustments to brightness and hue), and noise interference (such as Gaussian blur). These augmentation operations highlight the key features of the images without altering their semantics, helping to improve the model's robustness and generalization ability in diverse environments, thereby enhancing the adaptability of the image-text retrieval task in practical applications.

Figure 10

Data-enhanced image.



3.2. Experiments on Image-Text Contrastive Learning Based on the CLIP Framework

3.2.1. Programming Environment

Given the characteristics of multi-modal models and the high computational and memory requirements of this task, selecting a reliable and stable experimental environment is crucial. For this experiment, cloud-based rented servers were chosen to ensure efficient program execution. The specific experimental environment is shown in Table 3.

Table 3

Program Operating Environment.

Environment	Settings
system	ubuntu22.04LST
CPU	AMD EPYC 7T83
GPU	NAIDIA GeForce RTX 4090
internal storage	90 GB
video memory	24 GB GDDR6X
Python	Miniconda python3.10
Pytorch	pytorch2.3.0
CUDA	11.8

3.2.2. Model Parameter Settings

The image-word dataset (N=50,000) is divided into *training* and *validation* set in this experiment at a 7:3 ratio. To ensure fairness and coverage, the splitting maintained consistent distributions of languages and semantic categories between the two subsets. Relevant modules are first loaded in Python to extract features from images and texts. Then, in the *image – text* contrastive learning module, the temperature parameter τ of the loss-function L is set to 0.1 to enhance the distinction between negative and positive samples, thereby improving the accuracy of image-text retrieval. Considering that a smaller temperature parameter may cause overfitting, this study adjusts the learning rate of the *Adam* optimizer for improvement. To select the optimal learning rate, three candidates—0.01, 0.005, and 0.001—are

tested by observing the model's convergence performance. Table 4 displays the image-text contrastive model's precise parameter settings:

Table 4
Parameter Settings of the Image-Text Contrastive Learning Model

Module	Parameter	Settings
Feature Extraction	image feature	ViTFeature Extractor, ViTModel
	text feature	BertTokenizer, BertModel
Loss function L	temperature parameter τ	0.1
Adam	batch size	20
	learning rate η	0.001,0.005,0.01
	max_epochs	100
	Gradient first-moment decay factor β_1	0.9
	Gradient second-moment decay factor β_2	0.999

The gradient decay factors β_1 and β_2 in the Adam optimizer adopt the default values [8] (close to 1), which means the model relies more on the previous round of gradient information during parameter updates. It helps the parameter sets θ_b and θ_v of the BERT and VIT models converge to the optimal solution more quickly and stably. Considering the high feature dimensionality of the training samples (768 dimensions), this study performs iterative updates in each training epoch using every 20 image-text pairs as one batch in order to balance training efficiency and model stability.

3.2.3. Model Evaluation

Under the above parameter settings, the image-text contrastive learning model trained on the image-word dataset converges effectively under different learning rates (Figure 11), indicating that the model is effective and robust and capable of handling image-text retrieval tasks. Although larger initial learning rates (0.01, 0.005) can cause the loss function to drop rapidly in the early stages of training, they tend to induce oscillations in the later stages, affecting model stability. Therefore, this study ultimately selects the Adam optimization technique with an initial learning rate of 0.001, enabling the model to converge more stably toward the optimal solution while maintaining a relatively fast convergence rate and achieving good accuracy on both the training and validation sets.

Furthermore, a comparison between the proposed Adam optimizer and the traditional SGD optimizer with a fixed learning rate of 0.001 (Figure 12) shows that Adam leads to significantly faster and smoother reductions in the loss function for both training and validation sets. It also achieves a lower final loss value than SGD. This indicates that Adam effectively accelerates convergence and improves optimization efficiency, thereby enhancing the overall performance of contrastive learning in image-text retrieval.

Figure 11
Loss Function Curves of Adam under Different Initial Learning Rates.

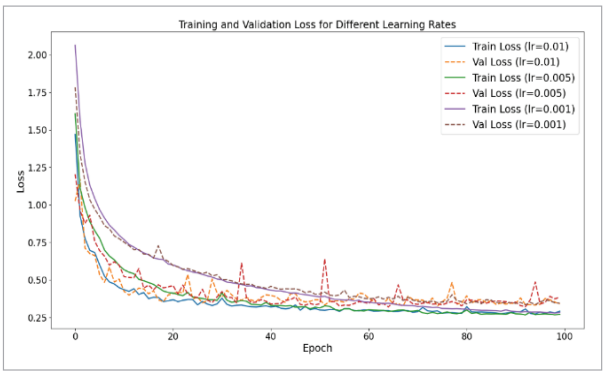
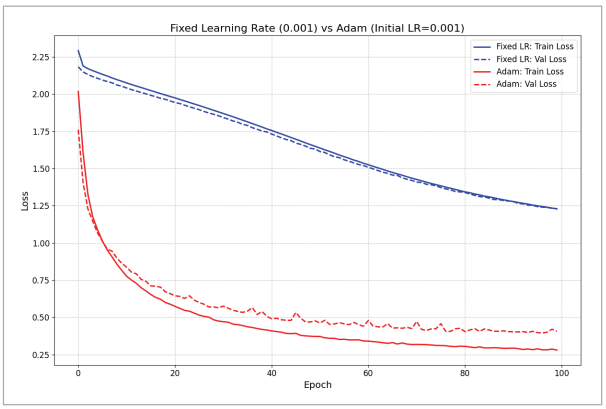


Figure 12
Comparison of Loss Curves Between Adam and Fixed-Learning-Rate SGD.



To gauge the performance of the *image–text* retrieval model, this paper adopts a widely used metric—Recall at K (*R@K*)—to assess the trained image-text contrastive learning model in both image retrieval and text retrieval tasks.

Specifically, in the image retrieval task (text_to_image), given a query text, the model retrieves a number of images from a candidate image pool that are semantically related to the text. R@K represents the proportion of query texts for which at least one correctly matched image appears in the top K retrieved results. The images are ranked based on their cosine similarity with the text features—the closer the similarity is to 1, the more relevant the image is and the higher it is ranked (see Figure 13). The expression for R@K in the *image* retrieval task is as follows:

$$R@K_{image} = \frac{\sum_{i=1}^{N_{texttest}} \mathbb{I}(GI_i \in Top\_K(sim(T_i, I)))}{N_{texttest}} \tag{13}$$

$$sim\ (T_i, I) = \{\cos(b_i, v_j) | j = 1,2,...,N_{imagedata}\}, \tag{14}$$

where $N_{texttest}$ denotes the total number of query texts, $N_{imagedata}$ denotes the total number of candidate images, T_i represents the *i–th* query text, and I denotes the set of candidate images. $sim(T_i, I)$ refers to the cosine similarity between the text T_i and all candidate images. $Top\_K(\cdot)$ selects the top K images with the highest score for similarity. GI_i is the true matching image for text T_i and $\mathbb{I}(\cdot)$ is an indicator function that returns 0 if the condition is false and 1 otherwise.

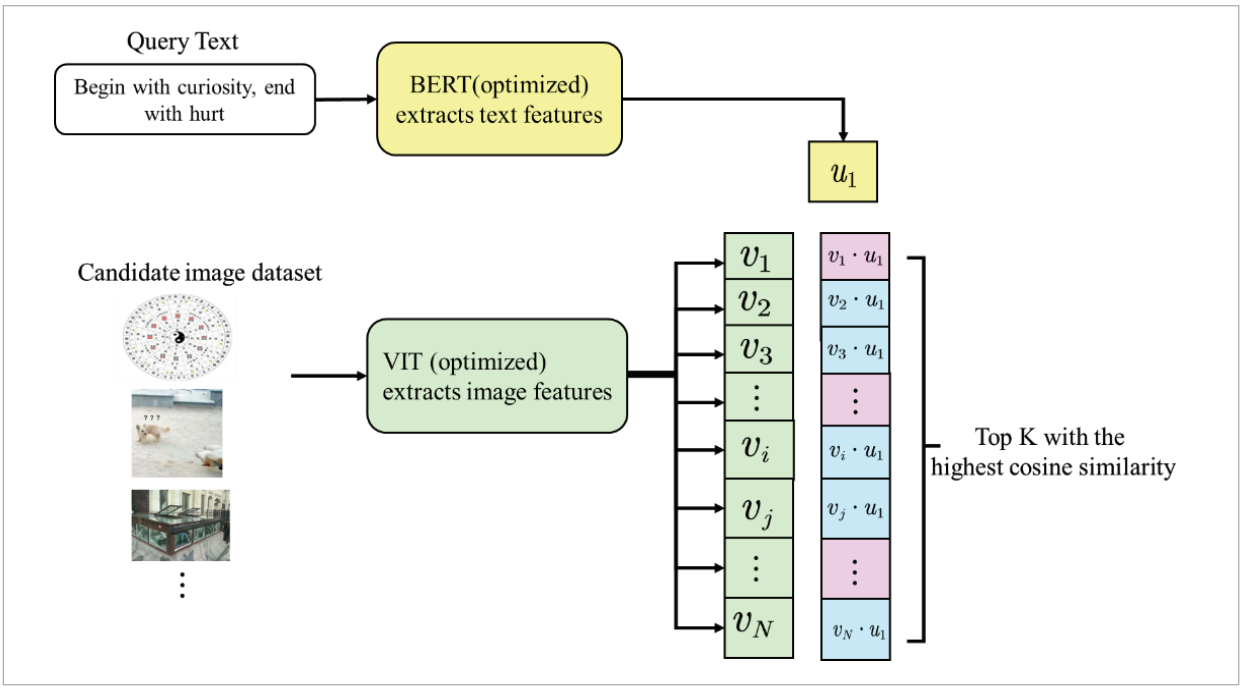
Similarly, based on the above approach, the recall rate R@K for the text retrieval task (Image_to_Text) can be defined as:

$$R@K_{text} = \frac{\sum_{i=1}^{N_{Imagetest}} \mathbb{I}(GT_i \in Top\_K(sim(I_i, T)))}{N_{Imagetest}} \tag{15}$$

$$sim\ (I_i, T) = \{\cos(v_i, b_j) | j = 1,2,...,N_{worddata}\}, \tag{16}$$

where $N_{Imagetest}$ denotes the total number of images to be matched; $N_{worddata}$ denotes the total number of candidate texts; I_i is the *i–th* image to be matched; T represents the set of candidate texts; $sim(I_i, T)$ is the cosine similarity between image I_i and all can-

Figure 13
Image Retrieval Process.



didate texts; $Top\_K(\cdot)$ selects the top K texts with the highest score for similarity; GT_i is the ground truth matching text for image I_i; $\mathbb{I}(\cdot)$ is an indicator-function that returns 0 if the condition is false, and 1 otherwise.

To validate the advantages of the proposed BERT-VIT contrastive learning model, this study introduces a baseline comparison using DeViSE [6], a classical model that maps visual features extracted via CNN into a semantic space learned from Word2Vec. While DeViSE does not adopt contrastive learning or large-scale pretraining, it remains a representative early framework in cross-modal retrieval research.

This paper uses R@K (K=1, 5, 10) to evaluate the performance of the image-text retrieval model (see Table 5). The results show that the proposed BERT-VIT contrastive learning model significantly outperforms the DeViSE baseline [6] across both retrieval directions, with improvements of 12.9% in R@1, 10.0% in R@5, and 9.0% in R@10, respectively. This indicates superior retrieval accuracy and semantic alignment capability.

To further validate the enhanced feature representation capability, the Feature Entropy and Mutual Information (MI) metrics introduced earlier reveal notable improvements: the entropy of text embeddings reaches 4.27 bits, and that of image embeddings 4.13 bits, both significantly higher than the DeViSE model (3.62 / 3.47 bits), demonstrating richer and finer semantic granularity. The MI between features and true labels reaches 1.83 bits for text and 1.75 bits for images, representing relative improvements of 28.3% and 31.5%, confirming enhanced semantic alignment and generalization.

Moreover, as previously discussed, the text and image preprocessing strategies adopted in this study contribute not only to performance but also to training efficiency: Shortening input text sequences while preserving semantic integrity reduces the number of training steps per epoch by approximately 18%; and Base64 encoding compresses raw RGB image data by around 30%, lowering memory and computational costs while retaining essential visual structures. These strategies improve training throughput and demonstrate practical value for large-scale image-text retrieval applications.

3.3. Image-Text Retrieval Task

To evaluate the retrieval performance of the model in real-world scenarios, this study tests the trained CLIP-based image-text contrastive learning model on two types of tasks. Specifically, for the text retrieval task, the model is required to retrieve relevant textual descriptions from the candidate text dataset (text-data, N=50,000) for the query image set (image-test, N=5,000). For the image retrieval task, the model must retrieve matching images from the candidate image dataset (image-data, N=50,000) for the query text set (text-test, N=5,000). Table 6 and Table 7 present examples of the retrieval results. The results indicate that the model can accurately identify key regions in images (e.g., retrieving football-related text based on a jersey) and core semantics in texts (e.g., retrieving parent-child scene images based on the keyword "Father's Day"). It demonstrates the model's strong capability in establishing semantic associations between images and texts, reflecting robust semantic alignment and cross-modal recognition performance.

Table 5

Performance Evaluation and Comparison of the Image-Text Matching Model.

Task	TEXTS-TO-IMAGES			IMAGES-TO-TEXTS		
R@K	R@1	R@5	R@10	R@1	R@5	R@10
BERT-VIT CLIP MODEL (Proposed)						
Training Set	86.70%	88.20%	93.60%	82.60%	91.30%	94.10%
Validation Set	79.60%	84.10%	88.80%	81.30%	83.40%	89.70%
DeViSE Baseline Model						
Training Set	72.80%	80.40%	84.30%	70.50%	79.10%	85.00%
Validation Set	66.70%	74.10%	78.90%	68.30%	72.60%	79.40%

Table 6
Text to Image Task.

text	NO.1	NO.2	NO.3	NO.4	NO.5
When the Dragon Boat Festival meets Father's Day					

Table 7
Image to Text Task.

NO.1	World No. 3 shares the stage — King Zha wins, but Neymar's skill still shines.
NO.2	We're playing the best football in all of Europe and want to face more top teams.
NO.3	Today's commentary: Roma vs Sassuolo, Liverpool vs Wolverhampton Wanderers.
NO.4	The first goal of this World Cup was born.
NO.5	Four years, six titles — extend for five more and win even more.

4. Conclusions

To address the challenges of balancing efficiency and performance in current cross-modal image-text retrieval systems, as well as limitations in semantic modelling, this paper constructs an image-text matching model based on the CLIP framework. The model integrates image-text feature extraction with contrastive learning modules, optimizing aspects such as semantic alignment, attention to key regions, and training efficiency.

In the feature extraction stage, the VIT and BERT models are employed to extract deep semantic features from images and text, respectively, enhancing the model's ability to perceive key image regions and comprehend contextual semantics in text. Since both models are pre-trained on large-scale image-text datasets, only fine-tuning is required for the current task, significantly reducing training costs. During contrastive learning, positive and negative sample pairs are constructed. A contrastive loss function based on cosine similarity is designed to guide the model in achieving cross-modal semantic alignment. The Adam optimizer is used to update the parameters of BERT and VIT, aiming to assign higher similarity to matching pairs and

lower similarity to non-matching ones, thereby enhancing the model's generalization ability while accelerating convergence and ensuring training stability.

In the experimental phase, the original image-text pairs were firstly preprocessed: images were encoded using Base64 and augmented through various techniques to enrich feature diversity, while redundant or meaningless characters were removed from text to improve semantic expression accuracy. The model was then trained on 50,000 paired samples with carefully tuned parameters to enhance the contrast between positive and negative samples and improve training efficiency and accuracy. Based on the convergence of the loss function, the optimal learning rate for the Adam optimizer was determined to be 0.001. Experimental results show that the model achieves high recall rates (R@K with K=1, 5, 10), consistently ranging between 80% and 90% on image-text retrieval tasks, demonstrating strong performance. Finally, the model is applied to large-scale image-text retrieval scenarios, with results indicating it can accurately identify semantic relationships between images and text, showing robust cross-modal matching capability.

Despite the strong results, the model still faces issues such as insufficient feature extraction and heavy reliance on large-scale pre-training. The efficiency of real-time retrieval systems also needs improvement. Future research will focus on optimizing model architecture and training strategies to enhance fine-grained semantic understanding. Moreover, exploring more efficient multi-modal fusion methods and expanding to a unified retrieval framework that incorporates additional modalities such as audio and video are promising directions for further study.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Aleem, S., Wang, F., Maniparambil, M., Arazo, E., Dietlmeier, J., Curran, K. Test-Time Adaptation with SALIP: A Cascade of SAM and CLIP for Zero-Shot Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, 5184-5193. https://doi.org/10.1109/CVPRW63382.2024.00526
2. Chen, C., Zhang, B., Cao, L., Shen, J., Gunter, T., Madappally Jose, A., Toshev, A., Shlens, J., Pang, R., Yang, Y. STAIR: Learning Sparse Text and Image Representation in Grounded Tokens. arXiv Preprint arXiv:2301.13081, 2023. https://doi.org/10.18653/v1/2023.emnlp-main.932
3. Csizmadia, D., Codreanu, A., Sim, V., Prabhu, V., Lu, M., Zhu, K., O'Brien, S., Sharma, V. Distill CLIP (DCLIP): Enhancing Image-Text Retrieval via Cross-Modal Transformer Distillation. arXiv Preprint arXiv:2505.21549, 2025. https://doi.org/10.48550/arXiv.2505.21549
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv Preprint arXiv:2010.11929, 2020
5. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, 4171-4186. https://doi.org/10.18653/v1/N19-1423
6. Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., Mikolov, T. DeViSE: A Deep Visual-Semantic Embedding Model. Advances in Neural Information Processing Systems, 2013, 26.
7. Feng, D., He, X., Peng, Y. MKVSE: Multi-Modal Knowledge Enhanced Visual-Semantic Embedding for Image-Text Retrieval. ACM Transactions on Multimedia Computing, Communications and Applications, 2023, 19(5), 1-21. https://doi.org/10.1145/3580501

8. He, L., Liu, S., An, R., Zhuo, Y., Tao, J. An End-to-End Framework Based on Vision-Language Fusion for Remote Sensing Cross-Modal Text-Image Retrieval. Mathematics, 2023, 11(10), 2279. https://doi.org/10.3390/math11102279
9. Huang, T., Zhu, Z., Jin, G., Liu, L., Wang, Z., Liu, S. SPAM: Spike-Aware Adam with Momentum Reset for Stable LLM Training. arXiv Preprint arXiv:2501.06842, 2025. https://doi.org/10.48550/arXiv.2501.06842
10. Kanadath, A., Jothi, J. A. A., Urolagin, S. CViTS-Net: A CNN-ViT Network with Skip Connections for Histopathology Image Classification. IEEE Access, 2024. https://doi.org/10.1109/ACCESS.2024.3448302
11. Khan, A., Rauf, Z., Sohail, A., Khan, A. R., Asif, H., Asif, A., Farooq, U. A Survey of the Vision Transformers and Their CNN-Transformer Based Variants. Artificial Intelligence Review, 2023, 56(Suppl 3), 2917-2970. https://doi.org/10.1007/s10462-023-10595-0
12. Kaya, Y. B., Tantuğ, A. C. BERT2D: Two Dimensional Positional Embeddings for Efficient Turkish NLP. IEEE Access, 2024. https://doi.org/10.1109/ACCESS.2024.3407983
13. Koca, N. A., Do, A. T., Chang, C. H. Hardware-Efficient Softmax Approximation for Self-Attention Networks. In 2023 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2023, 1-5. https://doi.org/10.1109/ISCAS46773.2023.10181465
14. Leng, X. L., Miao, X. A., Liu, T. Using Recurrent Neural Network Structure with Enhanced Multi-Head Self-Attention for Sentiment Analysis. Multimedia Tools and Applications, 2021, 80, 12581-12600. https://doi.org/10.1007/s11042-020-10336-3
15. Li, Y., El Habib Daho, M., Conze, P.-H., Zeghlache, R., Le Boité, H., Tadayoni, R., Cochener, B., Lamard, M., Quellec, G. A Review of Deep Learning-Based Information Fusion Techniques for Multi-Modal Medical Image Classification. Computers in Biology and Medicine, 2024, 108635. https://doi.org/10.1016/j.compbiomed.2024.108635
16. Li, H., Wang, M., Ma, T., Liu, S., Zhang, Z., Chen, P.-Y. What Improves the Generalization of Graph Transformers? A Theoretical Dive into the Self-Attention and Positional Encoding. arXiv Preprint arXiv:2406.01977, 2024.
17. Mogan, J. N., Lee, C. P., Lim, K. M. Ensemble CNN-ViT Using Feature-Level Fusion for Gait Recognition. IEEE Access, 2024. https://doi.org/10.1109/ACCESS.2024.3439602
18. Marshall, N., Xiao, K. L., Agarwala, A., Paquette, E. To Clip or Not to Clip: The Dynamics of SGD with Gradient Clipping in High-Dimensions. arXiv Preprint arXiv:2406.11733, 2024. https://doi.org/10.48550/arXiv.2406.11733
19. Ma, J., Huang, P.-Y., Xie, S., Li, S.-W., Zettlemoyer, L., Chang, S.-F. MODE: CLIP Data Experts via Clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, 26354-26363. https://doi.org/10.1109/CVPR52733.2024.02489
20. Pham, L. M., The, H. C. LNLF-BERT: Transformer for Long Document Classification with Multiple Attention Levels. IEEE Access, 2024. https://doi.org/10.1109/ACCESS.2024.3492102
21. Qin, X., Li, L., Pang, G., Hao, F. Heterogeneous Graph Fusion Network for Cross-Modal Image-Text Retrieval. Expert Systems with Applications, 2024, 249, 123842. https://doi.org/10.1016/j.eswa.2024.123842
22. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. Learning Transferable Visual Models from Natural Language Supervision. In International Conference on Machine Learning, PMLR, 2021, 8748-8763.
23. She, L., Gong, H., Zhang, S. An Interactive Multi-Head Self-Attention Capsule Network Model for Aspect Sentiment Classification. The Journal of Supercomputing, 2024, 80(7), 9327-9352. https://doi.org/10.1007/s11227-023-05813-z
24. Singla, S., Priyanshu, P., Thakur, A., Swami, A., Sawarn, U., Singla, P. Advancements in Natural Language Processing: BERT and Transformer-Based Models for Text Understanding. In 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), IEEE, 2024, 372-379. https://doi.org/10.1109/ICACCTech65084.2024.00068
25. Wu, F., Ma, Y., Jin, H., Jing, X.-Y., Jiang, G.-P. MFECLIP: CLIP with Mapping-Fusion Embedding for Text-Guided Image Editing. IEEE Signal Processing Letters, 2023, 31, 116-120. https://doi.org/10.1109/LSP.2023.3342649
26. Wu, X., Ajorlou, A., Wang, Y., Jegelka, S., Jadbabaie, A. On the Role of Attention Masks and LayerNorm in Transformers. arXiv Preprint arXiv:2405.18781, 2024. https://doi.org/10.48550/arXiv.2405.18781
27. Wang, J., Huang, J. X., Tu, X., Wang, J., Huang, A. J., Laskar, M. T. R., Bhuiyan, A. Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges. ACM Computing Surveys, 2024, 56(7), 1-33. https://doi.org/10.1145/3648471

28. Wang, A., Chen, H., Lin, Z., Han, J., Ding, G. RepViT: Revisiting Mobile CNN from ViT Perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, 15909-15920. https://doi.org/10.1109/CVPR52733.2024.01506

29. Wei, Y., Cao, Y., Zhang, Z., Peng, H., Yao, Z., Xie, Z. iCLIP: Bridging Image Classification and Contrastive Language-Image Pre-Training for Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 2776-2786. https://doi.org/10.1109/CVPR52729.2023.00272

30. Xia, L., Yang, Y., Chen, Z., Yang, Z., Zhu, S. Movie Recommendation with Poster Attention via Multi-Modal Transformer Feature Fusion. arXiv Preprint arXiv:2407.09157, 2024. https://doi.org/10.48550/arXiv.2407.09157

31. Xu, H., Fan, G., Kuang, G., Wang, C. Exploring the Potential of BERT-BiLSTM-CRF and the Attention Mechanism in Building a Tourism Knowledge Graph. Electronics, 2023, 12(4), 1010. https://doi.org/10.3390/electronics12041010

32. Yu, Z., Li, H., Feng, J. Enhancing Text Classification with Attention Matrices Based on BERT. Expert Systems, 2024, 41(3), e13512. https://doi.org/10.1111/exsy.13512

33. Zhang, X., Song, X., Feng, A., Gao, Z. Multi-Self-Attention for Aspect Category Detection and Biomedical Multilabel Text Classification with BERT. Mathematical Problems in Engineering, 2021, 2021(1), 6658520. https://doi.org/10.1155/2021/6658520

34. Zhang, M. Neural Attention: Enhancing QKV Calculation in Self-Attention Mechanism with Neural Networks. arXiv Preprint arXiv:2310.11398, 2023. https://doi.org/10.48550/arXiv.2310.11398

35. Zeng, R., Ma, W., Wu, X., Liu, W., Liu, J. Image-Text Cross-Modal Retrieval with Instance Contrastive Embedding. Electronics, 2024, 13(2), 300. https://doi.org/10.3390/electronics13020300

36. Zou, Z., Gan, H., Huang, Q., Cai, T., Cao, K. Disaster Image Classification by Fusing Multi-Modal Social Media Data. ISPRS International Journal of Geo-Information, 2021, 10(10), 636. https://doi.org/10.3390/ijgi10100636