Journal of Organizational and End User Computing
Volume 36 • Issue 1

An Image-Text Matching Method for Multi-Modal Robots

Ke Zheng, Hunan Biological and Electromechanical Polytechnic, China
Zhou Li, Hunan Biological and Electromechanical Polytechnic, China*

ABSTRACT

With the rapid development of artificial intelligence and deep learning, image-text matching has gradually become an important research topic in cross-modal fields. Achieving correct image-text matching requires a strong understanding of the correspondence between visual and textual information. In recent years, deep learning-based image-text matching methods have achieved significant success. However, image-text matching requires a deep understanding of intra-modal information and the exploration of fine-grained alignment between image regions and textual words. How to integrate these two aspects into a single model remains a challenge. Additionally, reducing the internal complexity of the model and effectively constructing and utilizing prior knowledge are also areas worth exploring, therefore addressing the issues of excessive computational complexity in existing fine-grained matching methods and the lack of multi-perspective matching.

KEYWORDS

Image-Text Matching, Multi-View Matching, Transformer

INTRODUCTION

With the continuous advancement of technology, robotics has made significant progress in various fields. Especially with the fusion of multimodal perception and artificial intelligence, robots have evolved from simple tools for task automation into partners with multisensory capabilities and intelligent interactions (B·Hme et al., 2012; Zhang et al., 2022; Paolanti et al., 2017). For example, tour guide robots, as prominent representatives of robotics technology, have garnered widespread interest in the tourism and cultural heritage sectors. In this challenging domain, multimodal robots with multi-view image-text matching capabilities are emerging, providing richer and more precise ways of information exchange for tour guide robots. Robots typically interact with their environment and humans through visual and textual data. Understanding images enables robots to interpret the physical world, while comprehending text helps them communicate with humans and access information on the internet. A deep understanding of both modalities allows robots to have a comprehensive perception of their surroundings, combining visual and textual information to make sense of complex situations. However, images are a form of visual data, while text is linguistic data, and they represent information with inherent differences. To bridge the gap between images and text, image-text matching technology

DOI: 10.4018/JOEUC.334701 *Corresponding Author

for robots requires a deep understanding of both modalities and their seamless integration, which adds complexity to the task of feature extraction (Russell et al., 2002; Yang et al., March 2019). Furthermore, reducing the model's complexity while enhancing its representation capabilities and interpretability is a significant challenge in this context (Paolanti et al., 2019). For the task of image-text matching, traditional methods mainly relied on manually annotating images and then comparing the text words with the manually assigned image labels (Changet al., 1981; Li et al., 2016). These methods involve fixed extraction of features from images and text words followed by matching, making them highly dependent on the quality of manually labeled images. These traditional methods also suffer from several disadvantages: weak feature extraction capabilities, poor noise resistance due to noise in manual annotations, mostly linear structures leading to weak generalization abilities. These drawbacks limited their applicability in real-world scenarios. Subsequently, researchers started to explore more sophisticated learning-based approaches for achieving image-text matching. For instance, Rasiwasia et al. used scale-invariant feature transform algorithms and document topic generation models to represent images and text, and then applied Canonical Correlation Analysis to learn the cross-modal correlations (Rasiwasia et al., 2010). Zhuang et al. leveraged commonality in multimodal data to construct a unified cross-modal association graph, which helped explore the connections between visual and textual data (Zhuang et al., 2008). Yang et al. established a cross-modal index space by mining heterogeneous multimodal data, subsequently generating a semi-semantic graph for cross-modal retrieval (Yang et al., 2010). While these methods have provided valuable insights and made significant progress in image-text matching research, they are often limited to specific small datasets. They may have excellent performance on those datasets but struggle to generalize to broader applications and different domains.

With the rapid advancement of deep learning, cross-modal research has become a popular field (Ma et al., 2022). In the feature learning of multimodal data, deep learning has the capability to nonlinearly map low-level features of multimodal data into high-level abstract representations (Salman et al., 2022). Image-text matching tasks, as a fundamental task in cross-modal research, have garnered extensive attention from scholars. In terms of images, early models utilized Convolutional Neural Networks (CNNs) to extract image features, often pre-trained on image classification tasks. For example, models such as 2WayNet (Eisenschtat et al., 2017), sm-LSTM (Huang et al., 2017), and SAN (Ji et al., 2019) used pre-trained VGG (Simonyan et al., 2014) networks to extract image features, while other models like VSE++ (Fartash et al., 2018), DPC (Zheng et al., 2020), and SCO (Huang et al., 2018) employed deep residual networks (ResNet) (He et al., 2016) pre-trained on the ImageNet (Deng et al., 2009) dataset for image feature extraction. On the text side, early models like m-RNN (Mao et al., 2014) and LRCN (Donahue et al., 2017) used recurrent neural networks (RNNs) or Long Short-Term Memory (LSTM) (Hochreiter, et al. 1997) networks to represent textual information and mapped each text sequence into a feature vector. However, these methods typically represented images and texts as global vectors, resulting in the loss of fine-grained information among image regions and text words, leading to relatively lower accuracy.

Due to the limitations of the aforementioned pre-trained networks, many researchers have attempted to use finer-grained feature representations: dividing images into multiple regions and sentences into words and phrases for representation. Regarding images, SCAN (Lee et al., 2018) was the first to use the object detection pre-trained model Faster-RCNN (Ren et al., 2015) to extract image features. SCAN utilized Faster-RCNN to extract 36 salient regions from an image, encoding each region into a feature vector. This approach allowed for the inclusion of detailed image features within these 36 salient regions. For text, VSE++ introduced the use of Bidirectional Gated Recurrent Units (Bi-GRU (Schuster et al., 1997)) to extract text feature vectors. Bi-GRU consists of both a forward and a backward gated unit, aggregating information from both directions of the sentence words to represent the word's features. Following this, almost all image-text matching models have used Faster-RCNN and Bi-GRU for extracting image and text features. Models such as CAMP (Wang et al., 2019), IMRAM (Chen et al., 2020), VSRN (Li et al., 2019), and others employed these pre-trained models

and achieved state-of-the-art results at the time, to some extent benefiting from Faster-RCNN and Bi-GRU. The above-mentioned methods have made significant advancements in feature extraction from both images and language. However, due to the complexity and sequential nature of language, there is still room for further exploration and improvement in language feature extraction models.

Recently, neural network has been attracted considerable attention (Bin et al., 2022). The Transformer model has achieved state-of-the-art results in various natural language processing tasks (Vaswani et al., 2017), such as sentence classification, sentence prediction, text translation, and more. BERT, based on the Transformer architecture, is a prominent natural language pre-training model composed of multiple stacked self-attention layers. BERT is trained on two unsupervised prediction tasks: the "Masked Language Model" and the "Next Sentence Prediction," using Wikipedia data and text corpora for pre-training (Devlin et al., 2018). BERT's results demonstrate its ability to generate longer-range and more accurate contextual representations. Consequently, the latest image-text matching models have attempted to utilize advanced BERT for text feature extraction, such as TIMAM (Sarafianos et al., 2019), SAEM (Wu et al., 2019), MMCA (Wei et al., 2020), and others. The above analysis shows that Faster-RCNN and BERT both excel in the task of image-text matching. However, considering the nature of tasks for multi-modal robots in a multi-visual context, there is an urgent need to develop a simple and efficient inference model.

Transformer model is popular in image-text matching due to the introduction of an attention mechanism with strong reasoning ability (Jie et al., 2021; Wang et al., 2022; Yang et al., 2023; Messina et al., 2021). For example, in (Yang et al., 2023), the authors employ a transformer encoder to extract intra-modality relationships present within both the image and text. They then achieve alignment using an efficient aggregation technique, enhancing alignment efficiency while maximizing the utilization of intra-modality information. In (Messina et al., 2021), in pursuit of separate feature extraction for large-scale retrieval systems, the authors introduce a method, Transformer Encoder Reasoning Network (TERN) built upon the Transformer Encoder. TERN effectively reasons on distinct modalities while achieving a unified abstract concept space through shared weights in deeper transformer layers. These references highlight the excellent modeling capabilities of the Transformer model, but there is currently limited literature on the use of Faster-RCNN, BERT, and Transformer for image-text matching in a multi-visual context suitable for robots.

Based on the discussions mentioned above, this paper proposes a multi-view image-text matching method based on the Transformer architecture. The main contributions are as follows:

(1) The proposed scheme utilizes Faster-RCNN to enhance the efficiency of image feature extraction, while employing the BERT model to improve the efficiency of text feature extraction, followed by a selection process to filter and retain valuable features.
(2) The use of a Transformer inference network to map images and text into a common space, obtaining compact vectors for both modalities. This inference network not only captures rich image and text information but also avoids redundant calculations of local similarities between image regions and text words, effectively reducing the computational burden of the model.
(3) To explore text descriptions from different viewpoints, this method introduces a multi-view matching module. This module leverages dilated convolutions to model image information from different perspectives, leading to more accurate similarity calculations between images and text.

PROBLEM STATEMENT AND RELATED THEORY

Problem Statement

Exploring methods for MVITM aims to achieve a deeper understanding and exploration of the relationships between images and text to address complex problems in visual understanding and natural language processing. Research in this field strives to enable computer systems to consider

information from multiple visual and linguistic perspectives, similar to how humans do, in order to accomplish more advanced intelligent tasks.

Taking a tourism navigation robot as an example, a tourism intelligent robot equipped with MVITM methods can provide travelers with a more personalized and intelligent tourism experience. For example, as shown in Figure 1, the image is described from three different angles in the text. The first one focuses on describing the architectural style in the image, using terms like "style of building" and "unique". The second angle emphasizes the appearance and location attributes of the building in the image, such as "red" and "middle". The third angle provides a more abstract summary of the entire image.

Faster-RCNN Model

CNN is a widely used deep learning model for tasks such as image classification, feature extraction, and pattern recognition. However, it focuses on the entire image and cannot pay attention to the important content within the image. As a result, significant regions may not be fully extracted, and many irrelevant region features may be generated. This type of global feature extraction, driven by non-salient regions, is referred to as "top-down feature extraction."

Different from CNN, Faster-RCNN is a feature extraction approach that focuses on salient, prominent regions, and it is known as "bottom-up feature extraction."(Anderson et al., 2018). From Figure 2, it generates candidate regions in the first stage using the Region Proposal Network (RPN) and then precisely locates and classifies objects in the second stage through Region of Interest (ROI) pooling, classification, and regression heads. Here are the main components and working principles of Faster-RCNN:

Figure 1. Textual descriptions from different perspectives

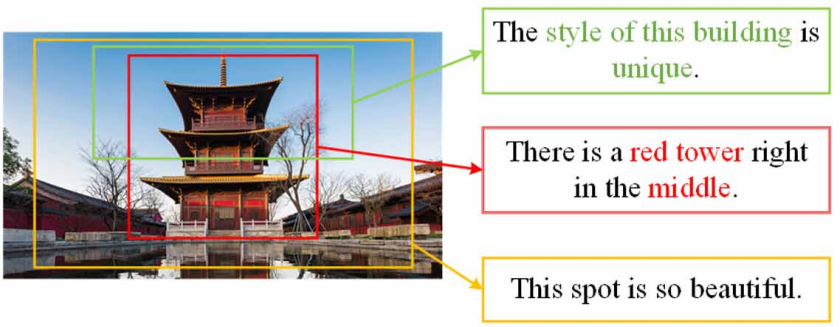
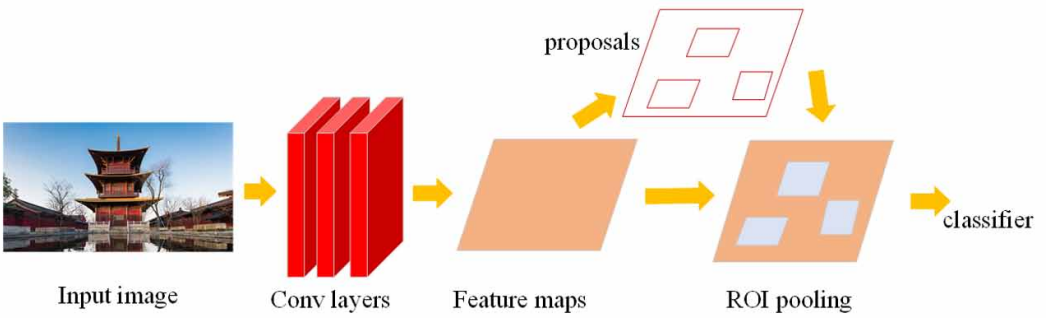


Figure 2. The flowchart of Faster-RCNN



(1) CNN: Faster-RCNN is built upon a CNN, typically pretrained models like ResNet or VGG, to extract features from images. This CNN network is usually pretrained on a large dataset to learn general features.
(2) Region Proposal Network (RPN): Faster-RCNN introduces RPN, which is a subnetwork specialized in generating candidate bounding boxes. RPN's job is to propose regions in the image that may contain objects, considering them as potential object locations. RPN accomplishes this task using anchor boxes and related classification and regression tasks.
(3) Region of Interest Pooling (ROI Pooling): Once candidate regions are obtained, Faster-RCNN employs ROI pooling to crop and resize these regions into feature maps with a fixed size. This ensures that candidate regions of different sizes have the same input dimensions for subsequent processing.
(4) Classification and Regression Heads: Candidate regions are processed through classification heads to determine the object's category and regression heads to precisely locate the object's position. These heads typically consist of fully connected neural network layers.
(5) Non-Maximum Suppression (NMS): To eliminate overlapping candidate boxes, Faster-RCNN uses the NMS algorithm to select the final detection results for each object. This ensures that each object is detected only once.

BERT Model

BERT is a bidirectional language pre-training model based on the Transformer architecture. Unlike fixed word embeddings in models like Word2Vec (Pennington et al., 2014) and GloVe (Wu et al., 2016), BERT consists of multiple stacked Transformer layers and is designed to learn deep bidirectional representations of unlabeled text by leveraging context information from all layers. It falls under the category of self-supervised language tasks.

The input to BERT is a continuous sequence of text, which can be a single sentence or a combination of multiple sentences. BERT's input representation is constructed through a combination of three types of embeddings: character embeddings, segment embeddings, and position embeddings, as shown in Figure 3. Specifically:

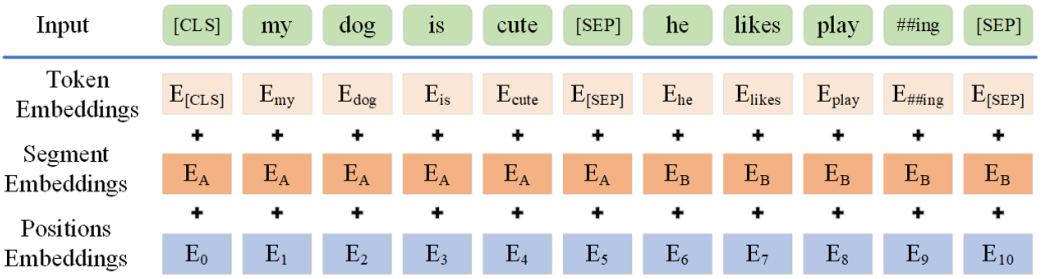
(1) Character Embeddings: Character embeddings are created using WordPiece (Hu et al., 2019) tokenization, which embeds 30,000 words into character vectors.
(2) Segment Embeddings: Segment embeddings are used to differentiate between different sentences in the input. This is done in two ways: by using special tokens to separate each sentence and by adding a learnable embedding layer to each character to indicate which sentence it belongs to.
(3) Position Embeddings: Position embeddings are added to each position in the input segment. These position vectors have a dimension equal to the maximum input segment length, which is typically set to 512.

In summary, BERT combines character embeddings, segment embeddings, and position embeddings to create rich input representations for understanding text with context, making it a powerful model for various natural language processing tasks.

Transformer Model

The attention mechanism is inspired by the way human visual perception works. Just like humans don't randomly scan an entire scene when observing it, but rather focus on specific parts of the image based on their points of interest, attention mechanisms aim to allocate attention to specific parts of input data. With the rapid development of deep learning, attention mechanisms have found wide applications across various fields. The attention mechanism can be seen as a mechanism that assigns

Figure 3. Input combinations for BERT



appropriate weights to input information, enhancing the learning of important information while suppressing responses to irrelevant information. It allows models to selectively focus on relevant parts of the input, making it a powerful tool for tasks involving sequential data, such as natural language processing and computer vision.

In recent years, Transformer models have made significant advances in various tasks (Pan et al., 2023 & Cao et al.,2023). When given a set of feature vectors as input, the Transformer can learn the correlations between any two features within this set. As shown in Figure 4., the Transformer's structure consists of two sub-layers: the multi-head self-attention layer and the feedforward neural network layer.

For each input vector, the vectors q, k and v can be obtained from the results of input vector X is multiplied by the weight matrix W^{Q}, W^{K} and W^{V}, respectively, which is expressed as

$$\left(\mathrm{q},\mathrm{k},\mathrm{v}\right) = MatMul\left(\mathrm{X},\left(\mathrm{W}^{\mathrm{Q}},\mathrm{W}^{\mathrm{K}},\mathrm{W}^{\mathrm{V}}\right)\right) \tag{1}$$

Then, the attention score is expressed as

$$\mathrm{Score}\left(\mathrm{q},\mathrm{k}\right) = q{\cdot}k \tag{2}$$

where $\mathrm{q}{\cdot}\mathrm{k}$ denotes the dot product of q and k.

To ensure stable gradients during training, all the obtained relevance scores are normalized by passing them through the softmax function, resulting in normalized relevance weights. These weights are then multiplied with the v values of all words, thereby emphasizing the words that need attention while de-emphasizing the irrelevant words. Finally, the weighted v values are summed up to obtain the corresponding feature vector, which is described as:

$$\mathrm{Attention}\left(\mathrm{Q},\mathrm{K},\mathrm{V}\right) = \mathrm{softmax}\left(\frac{\mathrm{Score}\left(\mathrm{q},\mathrm{k}\right)}{\sqrt{\mathrm{d}_{\mathrm{k}}}}\right)\mathrm{v} \tag{3}$$

Therefore, the multiple attention mechanism can be expressed as

$$\mathrm{MultiHead}\left(\mathrm{SV}\right) = Concat\left(\mathrm{head}_{1},\cdots\mathrm{head}_{\mathrm{h}}\right)\mathrm{W}^{\mathrm{O}} \tag{4}$$

$$head_i = Attention\left(FW_i^Q, FW_i^K, FW_i^V\right) \tag{5}$$

After the multi-head self-attention layer, to further refine the feature representation of the input sequence, the feature information at each position is passed through the "feedforward neural network layer." The "feedforward neural network" consists of two fully connected layers and a ReLU activation function, and it can be described using the following formula:

$$FFN\left(x\right) = W_2 ReLu\left(W_1 x + b_1\right) + b_2 \tag{6}$$

where W_1 and W_2 are parameter matrix of the fully connected layer. b_1 and b_2 are bias vectors.

METHODS

This chapter proposes a multi-view image-text matching method based on the Transformer architecture, where the Transformer inference network maps images and text to a common space and obtains compact vectors for both image and text. This inference network not only captures rich information from images and text but also avoids redundant calculations of local similarities between image regions and text words, effectively reducing the computational load of the model. Subsequently, to explore text descriptions from different perspectives, this approach designs a multi-view matching module. It uses dilated convolutions to model image information from different viewpoints, thereby calculating image-text similarity more accurately.

Image Feature Extraction

As described in Section 2.1, this paper adopts Faster-RCNN for image feature extraction. During the training phase, we employed several strategies to continuously enhance the performance of the Faster-RCNN model. Firstly, we conducted pretraining of the model using the benchmark dataset ImageNet. This step helps the model learn rich visual features. Subsequently, to further enrich the model's knowledge, we performed additional training using the Visual Genome dataset, which

Figure 4. Transformer self-attention mechanism architecture diagram

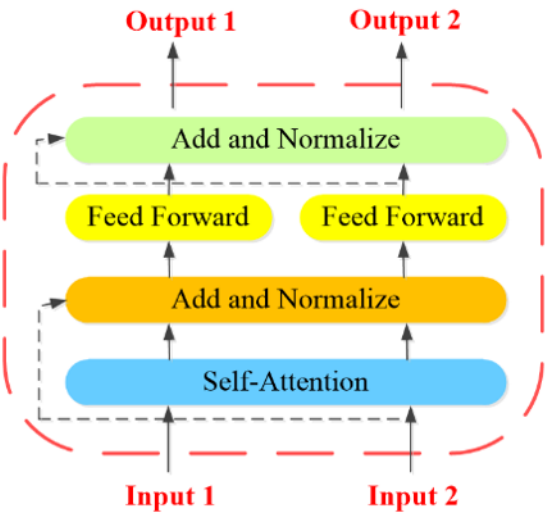
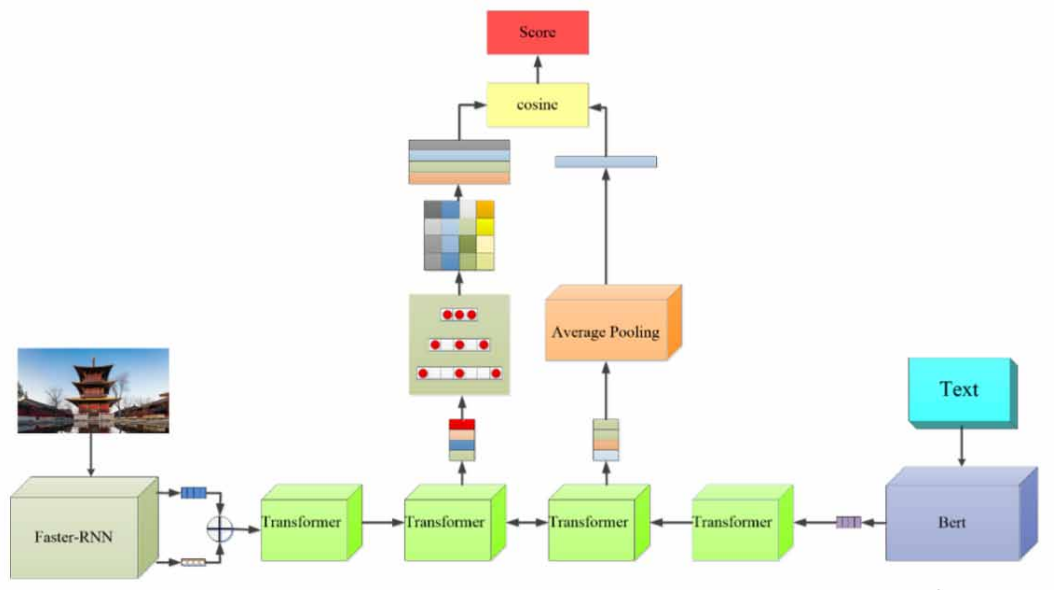


Figure 5. The model diagram of the proposed multi-view image text matching method



contains images along with corresponding textual descriptions. This dataset assists the model in better adapting to the image-text matching task, enabling it to understand semantic information in images.

Furthermore, to enhance the model's representational capacity, we employed a "bottom-up" approach. In the model's output, we added an additional output for predicting attribute classes. For instance, when identifying an object like "car" in "red car," the model not only predicts the object class but also predicts attribute classes like "red." This strategy helps the model more accurately associate salient regions with textual words, improving the model's correlation between images and text, and consequently, its performance in image-text matching tasks.

Finally, after extracting image region features using Faster-RCNN, we apply a fully connected layer to transform the feature dimension of image regions into 2048 dimensions. Hence, the feature of the selected picture can be expressed as $\mathrm{I} = \left\{\mathrm{I}_1, \mathrm{I}_2, \ldots, \mathrm{I}_\mathrm{n}\right\}$, where n denotes the number of image areas.

Text Feature Extraction

In this chapter's approach, we adopted the BERT model introduced in Section 2.3 for extracting context-enhanced word embeddings. BERT is a model that has been pre-trained on various natural language processing tasks, including sentence prediction and sentence classification. Unlike traditional Word2Vec and Glove models, BERT captures contextual information around each word in a more effective manner.

This effective capture of contextual information allows the model to generate different feature representations for sentences based on their semantic differences. For example, consider the word "bank." When it appears in a context related to "river," its embedding representation will contain semantic information about bodies of water and natural landscapes. However, when it appears in a context related to "finance," its embedding representation may contain semantics related to the financial and economic domain. This flexibility in capturing the meaning of words in different contexts makes BERT a powerful tool for processing textual data, enabling a better understanding and

expression of complex semantic information in text. The final obtained text feature vector can be represented as $T = \left\{t_1, t_2, \ldots, t_m\right\}$, where m indicates the number of words in the text.

Position Coding

Since the Faster-RCNN model only extracts image features and does not include any spatial information about image regions, the visual reasoning module would be unable to reason about spatial relationships. This is disadvantageous for tasks involving multiple text descriptions and multi-view matching, as many text descriptions contain spatial information, such as "top" or "bottom." To introduce spatial information into the visual reasoning module, this approach uses the coordinates of the image region bounding boxes to construct position encodings.

In the Faster-RCNN model, the coordinates of each region bounding box are typically represented as $p = \left(x_1, y_1, x_2, y_2\right)$, with x_1, y_1 being the coordinates of the upper left corner of the image region, and x_2, y_2 are the coordinates of the lower right corner of the image region. Subsequently, we calculate the absolute positions of the region coordinates and normalize both the region coordinates and the region area using the following formula:

$$P = \left(\frac{x_1}{W}, \frac{y_1}{W}, \frac{x_2}{H}, \frac{y_2}{H}, \frac{\left(x_2 - x_1\right)\left(y_2 - y_1\right)}{WH}\right) \tag{7}$$

where W denotes the width of the whole image and H denotes the height of the whole image.

Finally, the image feature vector I and the position encoding vector P are concatenated, and then passed through two fully connected layers with ReLU activation functions to obtain the final image feature with spatial information $\overline{I}$

$$\overline{I} = W_2 ReLu\left(W_1\left(I \oplus P\right) + b_1\right) + b_2 \tag{8}$$

where W_1 and W_2 are the fully connected layer parameter matrix. The symbol $\oplus$ denotes a splice operation.

Transformer Inference Networks

The image inference and text inference in this chapter are both based on the Transformer architecture. Specifically, the input to the inference model consists of a set of image region feature vectors $\overline{I} = \left\{r_1, r_2, \ldots, r_n\right\}$ and a set of text word feature vectors $T = \left\{t_1, t_2, \ldots, t_m\right\}$ after position encoding. For the image branch, the inference network uses 4 stacked Transformer architectures for image inference. For the text branch, since the BERT pretrained model is composed of the Transformer architecture, we choose an appropriately sized BERT pretrained model and fine-tune it. No additional Transformer layers are added during the inference stage for the text branch. Finally, a final layer of the Transformer architecture is added to both the image branch and the text branch, and the weights are shared between these branches in the final layer.

Based on the self-attention mechanism introduced in Section 2.4, the multi-head self-attention mechanism does not consider the order of input sequences. Therefore, it can be applied to unordered image region features. Specifically, given a set of image region features $\overline{I} = \left\{r_1, r_2, \ldots, r_n\right\}$, the output after the multi-head self-attention layer is represented as $O = \left\{o_1, o_2, \ldots, o_n\right\}$, which is expressed as follows

$$O = LayerNorm\left(\left(\mathrm{MultiHead}\left(\overline{\mathrm{I}}\right)\right) + \overline{\mathrm{I}}\right) \tag{9}$$

Subsequently, $\mathrm{O} = \mathrm{o}_1, \mathrm{o}_2, \ldots, \mathrm{o}_n$ is passed through the feedforward neural network layer in the Transformer model, resulting in an output represented as $\mathrm{I}^* = \left\{\mathrm{r}_1^*, \mathrm{r}_2^*, \ldots, \mathrm{r}_n^*\right\}$, where the calculation of r_i^* is as follows:

$$\mathrm{r}_i^* = LayerNorm\left(\mathrm{FFN}\left(\mathrm{o}_i\right) + \mathrm{o}_i\right) \tag{10}$$

The text inference process for a single-layer Transformer architecture is similar to the process described above. Specifically, given a set of text word features $\mathrm{T} = \{\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_m\}$, after passing through the multi-head self-attention layer and feedforward neural network layer within the Transformer model, the resulting text features are represented as $\mathrm{T}^* = \left\{\mathrm{t}_1^*, \mathrm{t}_2^*, \ldots, \mathrm{t}_m^*\right\}$.

To obtain a compact vector representation of the text features, the average pooling operation is applied to the text word features $\mathrm{T}^* = \left\{\mathrm{t}_1^*, \mathrm{t}_2^*, \ldots, \mathrm{t}_m^*\right\}$ to obtain the final text global vector

$$\mathrm{T}_{\mathrm{Global}} = \frac{1}{m}\sum_{i=1}^{m} \mathrm{t}_i^* \tag{11}$$

where $\mathrm{T}_{\mathrm{Global}} \in \mathrm{R}^{1\times d}$. Using this compact global vector for similarity measurement can effectively reduce the computational complexity of the model. Additionally, due to the modeling by multiple layers of Transformers, this compact vector contains rich semantic information.

Multi-View Matching

To focus on the entire image from different perspectives, this method utilizes dilated convolution to model the importance of image information under different perspectives. Dilated convolution is part of the Dense Prediction Models. It can not only predict what certain entities are in the image but also precisely locate the positions and contours of these entities and determine which pixels they belong to. The specific roles of dilated convolution can be summarized as follows:

(1) Expanding the receptive field: Traditional down sampling methods expand the receptive field while reducing the image's resolution. Dilated convolution expands the receptive field without losing image resolution. The expanded receptive field allows the model to detect larger objects, while maintaining resolution enables precise localization.
(2) Aggregating multiscale contextual information: The key parameter of dilated convolution is the dilation rate of the convolution kernel. Based on different dilation rates, dilated convolution can aggregate information from different-sized regions of the image, which is crucial in multi-perspective tasks.

For the task in this chapter, after obtaining the image features $\mathrm{I}^* = \left\{\mathrm{r}_1^*, \mathrm{r}_2^*, \ldots, \mathrm{r}_n^*\right\}$ from the Transformer inference network, this method uses dilated convolutions of various sizes and dilation rates to explore multi-perspective information in the image. The specific process is as follows: As the dilation rate increases, the convolutional kernel's receptive field expands without losing resolution. For example, for a standard 3×3 convolutional kernel, increasing the dilation rate from

1 to 2 increases its receptive field from 3 to 5. The formula to calculate the receptive field of a dilated convolution is as follows: $\mathrm{RF}=\mathrm{k}+\left(\mathrm{k}-1\right)*\left(\mathrm{d}-1\right)$, where RF (Receptive Field) represents the receptive field size, k is the original convolution kernel size, and d is the dilation rate coefficient. This method uses 9 dilated convolutions with different parameters to model image features $\mathrm{I}^{*}=\left\{\mathrm{r}_{1}^{*},\mathrm{r}_{2}^{*},\ldots,\mathrm{r}_{\mathrm{n}}^{*}\right\}$, and concatenates the final outputs. For an image region $\mathrm{r}_{\mathrm{i}}^{*}$, the output after dilated convolution is given by

$$\mathrm{F}_{\mathrm{i}}=Dilated\left(\mathrm{r}_{\mathrm{i}}^{*}\right)=concat\left(\mathrm{f}_{1},\mathrm{f}_{2},\ldots,\mathrm{f}_{9}\right),i\in\left(1,\mathrm{n}\right) \tag{12}$$

$$F=\left\{\mathrm{F}_{1},\mathrm{F}_{2},\ldots,\mathrm{F}_{\mathrm{n}}\right\} \tag{13}$$

where $\mathrm{Dilated}\left(\cdot\right)$ denotes the null convolution operation and f_{i} denotes the output of the ith convolutional kernel. F denotes the whole image with all significant regions of the whole image after hollow convolution. The parameters of the nine null convolution kernels are shown in Table 1.

Next, we added a fully connected layer and applied softmax normalization function to the feature vector $\mathrm{F}=\left\{\mathrm{F}_{1},\mathrm{F}_{2},\ldots,\mathrm{F}_{\mathrm{n}}\right\}$, resulting in a multi-perspective matrix $\mathrm{F}^{*}=\left\{\mathrm{F}_{1}^{*},\mathrm{F}_{2}^{*},\ldots,\mathrm{F}_{\mathrm{n}}^{*}\right\}\in\mathrm{R}^{\mathrm{n}\times\mathrm{M}}$, where n represents the number of image regions, and M represents the number of perspectives.

Finally, the image feature vector $\mathrm{I}^{*}=\left\{\mathrm{r}_{1}^{*},\mathrm{r}_{2}^{*},\ldots,\mathrm{r}_{\mathrm{n}}^{*}\right\}\in\mathrm{R}^{\mathrm{n}\times\mathrm{M}}$ is multiplied with the multi-perspective matrix $\mathrm{F}^{*}=\left\{\mathrm{F}_{1}^{*},\mathrm{F}_{2}^{*},\ldots,\mathrm{F}_{\mathrm{n}}^{*}\right\}\in\mathrm{R}^{\mathrm{n}\times\mathrm{M}}$ to obtain the multi-view image feature $\mathrm{I}_{\mathrm{view}}$:

$$\mathrm{I}_{\mathrm{view}}=\left(\mathrm{F}^{*}\right)^{\mathrm{T}}\mathrm{I}^{*} \tag{14}$$

where $\mathrm{I}_{\mathrm{view}}=\left\{\mathrm{c}_{1},\mathrm{c}_{2},\ldots,\mathrm{c}_{\mathrm{M}}\right\}\in\mathrm{R}^{\mathrm{M}\times\mathrm{d}}$, M denotes the number of viewpoints and d denotes the image feature dimension.

Loss Function

To obtain multi-view matching, this method uses the multi-view image feature vector $\mathrm{I}_{\mathrm{view}}\in\mathrm{R}^{\mathrm{M}\times\mathrm{d}}$ and the text global vector $\mathrm{T}_{\mathrm{Global}}\in\mathrm{R}^{\mathrm{d}}$ to construct a multi-view similarity vector A. The specific construction method is as follows: calculate the cosine similarity between the image features c_{i} from all perspectives and the text global vector $\mathrm{T}_{\mathrm{Global}}$, the calculation process is as follows:

$$\mathrm{A}_{\mathrm{i}}=\frac{\mathrm{c}_{\mathrm{i}}^{\mathrm{T}}\mathrm{T}_{\mathrm{Global}}}{\mathrm{c}_{\mathrm{i}}\mathrm{T}_{\mathrm{Global}}},i=1,2,\ldots,M \tag{15}$$

Table 1. The settings of null convolution parameter

Num	1	2	3	4	5	6	7	8	9
k	1×1	3×3	3×3	3×3	3×3	5×5	5×5	5×5	5×5
d	1	1	2	3	4	2	3	4	6
RF	1×1	3×3	5×5	7×7	9×9	9×9	13×13	17×17	25×25

where the multi-view similarity vector $\mathrm{A} \in \mathrm{R}^{\mathrm{M}}$ contains similarity scores for M different perspectives. Subsequently, the similarity scores between the entire image and the text are obtained by applying max-pooling operation to the similarity vector A, resulting in the final image-text similarity score. The final image-text similarity score is given by

$$S\left(\mathrm{I},\mathrm{T}\right) = MaxPooling\left(\mathrm{A}\right), i = 1,2,\ldots,M \tag{16}$$

After obtaining the similarity scores, this approach utilizes the "Hardest Triplet Loss" function to strengthen the alignment between images and text. "Hardest Triplet Loss" prioritizes the most challenging data triplets, enabling the model to learn more accurate and discriminative feature representations faster. This enhances performance in similarity tasks and increases resilience to data noise, making it particularly well-suited for tasks like face recognition, image retrieval, and similarity-based recommendations. Unlike the "Triplet Loss" function, the "Hardest Triplet Loss" function focuses exclusively on the hardest negative samples within a small batch training set, assigning all gradients to the hardest negative sample rather than distributing gradients evenly among all negative samples. When dealing with a large number of easy negative samples dominating the loss function, the "Hardest Triplet Loss" function is less likely to get stuck in local minima, making it more beneficial for the image-text matching task. It can be formulated as

$$Loss\left(\mathrm{I},\mathrm{T}\right) = \max\left[0, \pm - \mathrm{S}\left(\mathrm{I},\mathrm{T}\right) + \mathrm{S}\left(\mathrm{I},\hat{\mathrm{T}}\right)\right] + \max\left[0, \pm - \mathrm{S}\left(\mathrm{I},\mathrm{T}\right) + \mathrm{S}\left(\hat{\mathrm{I}},\mathrm{T}\right)\right] \tag{17}$$

Where $\pm$ is the margin distance parameter, $\mathrm{S}\left(\cdot,\cdot\right)$ is the formula for calculating similarity scores, $\mathrm{S}\left(\mathrm{I},\mathrm{T}\right)$ represents the similarity score for correct matching image-text pairs, and $\mathrm{S}\left(\mathrm{I},\hat{\mathrm{T}}\right)$ and $\mathrm{S}\left(\hat{\mathrm{I}},\mathrm{T}\right)$ denote the similarity scores for the hardest negative samples from image to text and from text to image, respectively. In experiments, only the hardest negative sample from each mini-batch is used, instead of aggregating all negative samples.

EXPERIMENT RESULTS AND ANALYSIS

Dataset

This paper conducted experiments using two benchmark datasets in the field of image-text matching, namely the MS-COCO dataset and the Flickr30K dataset.

MS-COCO dataset, released by Microsoft Research, is widely used in computer vision tasks, including object detection, semantic segmentation, image captioning, and more. It comprises 123,287 images, each with 5 corresponding textual descriptions. The dataset contains three types of images: standard object images, standard scene images, and non-standard images. In this study, the MS-COCO dataset was split in the manner proposed by Andrej Karpathy, with 113,287 images used for training, 5,000 for validation, and 5,000 for testing. During testing, two modes were employed: 1K and 5K testing. In the 1K testing, the test set was divided into 5 equal parts, and 5-fold cross-validation was performed, with the average results considered as the final accuracy. In the 5K testing, all 5,000 test images were used directly for testing.

The Flickr30K dataset consists of 31,000 images, similar to MS-COCO, with each image having 5 textual descriptions. Unlike MS-COCO, Flickr30K does not categorize images into types. The images in the dataset represent everyday scenes, and their corresponding textual descriptions vary with distinct emphases. This dataset is known for its large volume, rich content, and broad coverage,

Algorithm Transformer model

Input:	Image feature vector $\mathrm{I}=\left\{\mathrm{I}_1,\mathrm{I}_2,\ldots,\mathrm{I}_n\right\}$ and text feature vector $\mathrm{T}=\left\{\mathrm{t}_1,\mathrm{t}_2,\ldots,\mathrm{t}_m\right\}$
Output:	Accuracy
1:	Obtain image features with spatial information according to Equation (8)
2:	Repeat:
3:	New image features $\mathrm{I}^*=\left\{\mathrm{r}_1^*,\mathrm{r}_2^*,\ldots,\mathrm{r}_n^*\right\}$ and new text features $\mathrm{T}^*=\left\{\mathrm{t}_1^*,\mathrm{t}_2^*,\ldots,\mathrm{t}_m^*\right\}$ are obtained by Transformer according to Equation (9)-(11)
4:	Obtain a multi-view matrix $\mathrm{F}^*=\left\{\mathrm{F}_1^*,\mathrm{F}_2^*,\ldots,\mathrm{F}_n^*\right\}$ according Equation (12)-(13)
5:	Obtain multi-view image features $\mathrm{I}_{\mathrm{view}}=\left\{\mathrm{c}_1,\mathrm{c}_2,\ldots,\mathrm{c}_M\right\}$ according Equation (14)
6:	Using the hardest ternary loss function to calculate the loss value according to Equation (17)
7:	Update the weight matrix in transformer
8:	until the maximum iteration reached
9:	Return the accuracy

making it a primary dataset in the field of image-text matching. For this study, 29,000 images were used for training, 1,000 for validation, and the remaining 1,000 for testing.

Experimental Setting

This approach conducted experiments and testing on the MS-COCO 1K dataset and the Flickr30K dataset, using R@1, R@5, and R@10 as evaluation metrics to assess the model.

For the feature extraction stage, the image branch utilized Faster-RCNN to extract n=36 salient regions and applied a fully connected layer to make the dimension of each salient region feature d=2048. The text branch used a pre-trained BERT model, consisting of 12 self-attention layers, each with 12 heads and 768 hidden units. During the training phase, the parameters of the pre-trained BERT model were kept fixed.

In the inference network stage, the visual branch utilized 4 stacked Transformer layers. Since the text branch already embedded the Transformer architecture in the BERT model, only the Transformer layer parameters within the BERT model were fine-tuned, without adding extra Transformer layers. In the final two Transformer architectures, both image and text features were mapped to a common space of 1024 dimensions.

For the multi-view matching stage, the number of views (M) was set to 10, and the edge parameter $\pm$ in formula (17) was set to 0.2. The approach used the Adam optimizer for training for 30 epochs, with a minimum batch size of 128. The learning rate was set to 0.0001 for the first 20 epochs and then lowered to 0.00001 for the remaining 10 epochs.

Experimental Results

In this experiment, we compared our method with several common algorithms, denoted as VSE++ (Fartash et al., 2018), SCAN (Lee et al., 2018), CAMP (Lee et al., 2018), SAEM (Wu et al., 2019), RDAN (Hu et al., 2019), VSRN (Li et al., 2019), IMRAM (Chen et al., 2020), MMCA (Wei et al., 2020), GSMN (Liu et al., 2020), GAMERA (Qu et al., 2020), MIRPF (Jie et al., 2023), and the

proposed scheme (TMIM). Furthermore, we conducted matching from image to text and from text to image separately. The experimental results are presented in Figures 6 and 7, respectively.

In Figures 6 and 7, our approach showcased impressive performance on the MS-COCO 1K dataset, nearly reaching the state-of-the-art benchmarks. When compared to the current state-of-the-art CAMERA model, our method displayed substantial improvements, boasting a 3.8% increase in R@1 accuracy for image-to-text matching and a 0.8% enhancement in text-to-image matching. These results underscore the superiority of our stacked Transformer architecture and the efficacy of our multi-view matching strategy.

Moreover, we evaluated the computational efficiency of our approach on the MS-COCO dataset using a single GPU. For image-to-text matching, our method requires a mere 0.2 seconds to compute and rank the similarity scores for the top 5000 text descriptions when given an image. Similarly, for text-to-image matching, it takes only 0.06 seconds to perform the same task when provided with a text description. In contrast, the SCAN model, which utilizes a cross-modal attention mechanism, takes substantially longer at 3.8 seconds and 1.5 seconds for these respective tasks. This demonstrates the remarkable efficiency of our method, rendering it suitable for mobile applications and large-scale datasets.

Turning our attention to Figures 8 and 9, our method delivered strong results on the Flickr30K dataset. In comparison to the suboptimal GSMN model, our approach improved the R@1 accuracy by 0.5% for image-to-text matching and an impressive 1.2% for text-to-image matching. Furthermore, our method significantly closed the performance gap when compared to the state-of-the-art MIRPF model.

Figure 6. Experimental results of image-to-text on MS-COCO 1K dataset

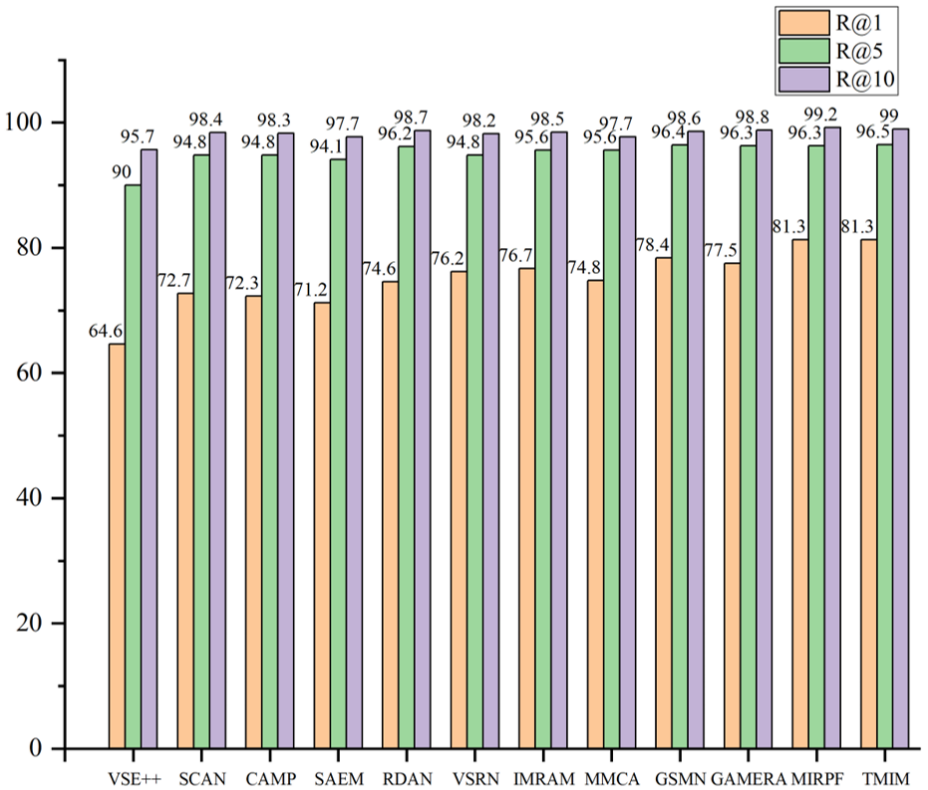
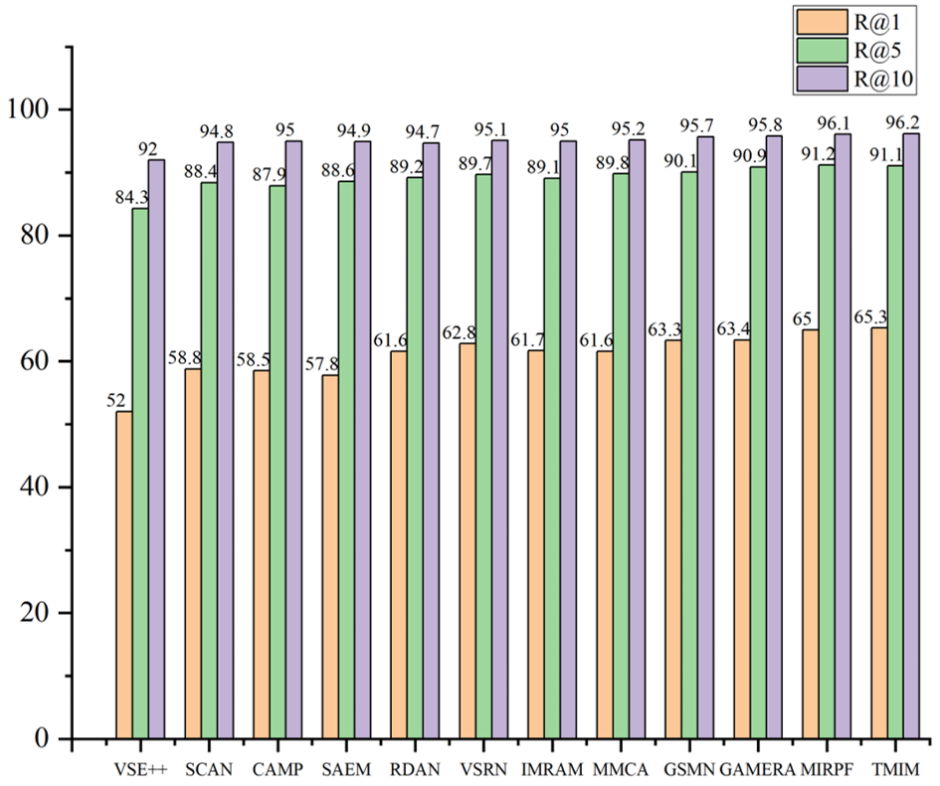


Figure 7. Experimental results of text-to-image on MS-COCO 1K dataset



To explore the impact of different modules on the image-text matching results, we conducted ablation experiments on both the Image-Text Matching Module and the Prior Knowledge Graph Module.

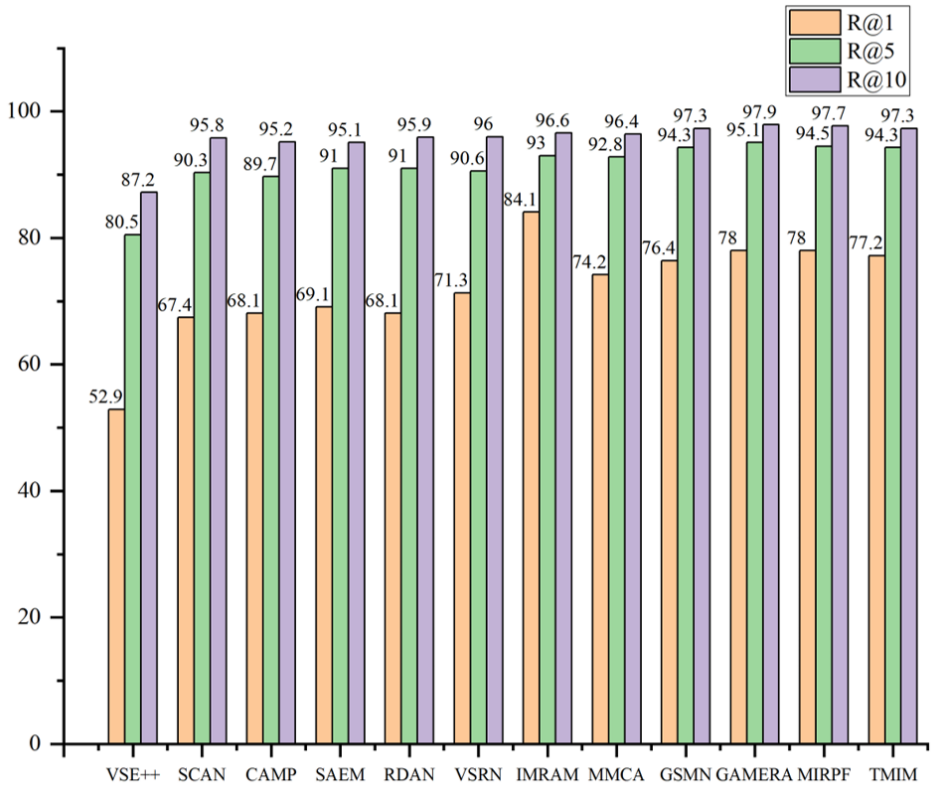
First, ablation experiments for the Image-Text Matching Module (A):

(1) No att: This involves removing the Transformer self-attention mechanism and replacing it with fully connected layers and average pooling to obtain a global image feature vector.
(2) No 1D-CNN: This experiment removes the 1D-CNN component and uses fully connected layers and average pooling to obtain a global text feature vector.
(3) Text LSTM: In this experiment, the BERT pretrained model is replaced with a LSTM model.
(4) Text Bi-GRU: Similarly, the BERT pretrained model is replaced with a Bidirectional Gated Recurrent Unit (Bi-GRU).

Ablation experiments for the Prior Knowledge Graph Module (B):
Second, ablation experiments for a priori knowledge graph modules include:

(1) No GCN: This experiment removes the Graph Convolutional Network (GCN) module, using only feature vectors derived from glove embeddings as prior knowledge and omitting the learning of relationships within the prior knowledge.
(2) No Wtag: In this case, the use of word labels to guide text feature representation is removed.

Figure 8. Experimental results of image-to-text on Flickr30K dataset



(3) No $»_2$: This experiment removes the entire Prior Knowledge Graph Module, relying solely on global image and text feature vectors for similarity measurement.

These ablation experiments were performed to analyze how each module contributes to the image-text matching performance.

From Table 2, we can observe the following:

(1) No att: Removing the self-attention mechanism significantly decreases the model's accuracy, indicating that self-attention is effective in aggregating complex relationships between image regions.
(2) No 1D-CNN: Eliminating the 1D-CNN results in a substantial drop in model performance. This is because text feature representation not only affects the text branch but also influences the training of the Transformer in the image branch. Therefore, good text feature representation is crucial.
(3) Text LSTM: The use of LSTM's may result in inadequate feature extraction of the text and a reduction in the final accuracy.
(4) Text Bi-GRU: Using Text Bi-GRU causes a slight decrease in model accuracy, suggesting that the new text pretraining method, BERT, has a beneficial effect on the model.
(5) No GCN: Removing the GCN module leads to a relatively small decrease in model performance, indicating that GCN indeed learns dependencies between prior knowledge. Additionally, using only the glove technique also improves model performance, showing that prior knowledge can provide valuable information.

Figure 9. Experimental results of text-to-image on Flickr30K dataset

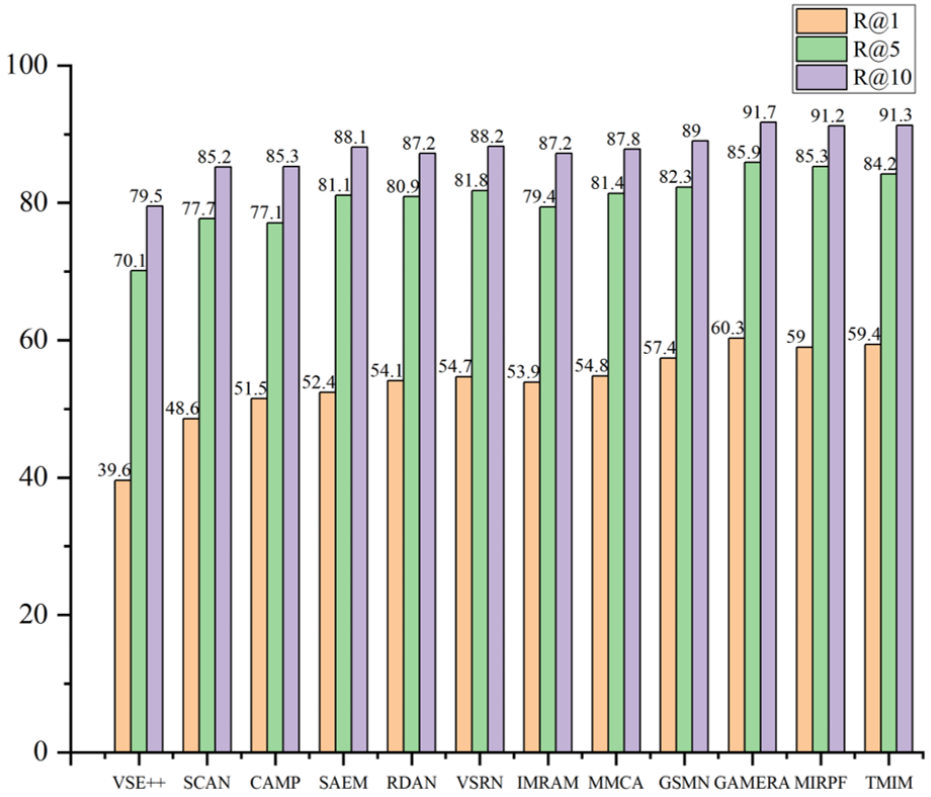


Table 2. Results of ablation experiments

Model	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
No att	71.4	90.7	94.9	52.1	80.3	87.6
No 1D-CNN	72.2	90.2	94.3	53.4	79.3	87.8
Text LSTM	73.1	90.5	94.7	54.3	81.2	88.3
Text Bi-GRU	74.8	91.0	95.8	56.9	83.2	89.0
No GCN	73.2	90.9	95.2	55.6	83.0	88.6
No Wtag	71.1	91.2	96.4	50.2	80.5	88.2
No $»_2$	70.4	89.5	94.7	47.7	79.2	89.8
TMIM	77.5	93.7	96.8	59.8	85.3	91.2

(6) No Wtag: The absence of word tags results in a significant decrease in model accuracy. This is because prior knowledge mined from the text corpus, along with word tags, enhances text feature representation, which contributes to higher model accuracy.

(7) No 2: Removing the prior knowledge graph module leads to a substantial decrease in model performance, underscoring the effectiveness and necessity of the prior knowledge graph module.

These ablation experiments highlight the importance of various components in the model and how they contribute to its overall performance.

Based on the analysis above, in the approach proposed in this paper, Faster-RCNN demonstrates stronger contour recognition capabilities, while the BERT language model excels at extracting textual contextual features. Finally, the utilization of the Transformer framework enables outstanding performance in inference tasks.

CONCLUSION

This paper introduces an image-text matching method based on prior knowledge graphs. It comprises three main modules: the prior knowledge graph module, the image-text matching module, and the integration module. The prior knowledge graph module and the image-text matching module are connected to the integration module. In the prior knowledge graph module, external prior knowledge graphs are used to guide image-text matching. Graph convolution is employed to establish relationships between prior knowledge, significantly enhancing the model's understanding of real-world scenes, reducing computational complexity and parameters, and improving inference speed. In the image-text matching module, the proposed method utilizes Transformer self-attention mechanisms to aggregate relationships between image regions. It also employs the pre-trained BERT model to extract text feature vectors and employs 1D convolution to capture phrase-level information, effectively improving model accuracy. Additionally, to reduce feature disparities between image and text modalities for better similarity measurement, the method introduces relative entropy in the loss function. Finally, extensive experiments are conducted to validate the substantial contributions of each module. In the future, research will concentrate on exploring novel fusion mechanisms to further improve text-image matching and enhance the accuracy of cross-modal matching.

REFERENCES

Anderson, P., He, X., & Buehler, C. (2018). Bottom-up and Top-down Attention for Image Captioning and Visual Question Answering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 6077-6086. doi:10.1109/CVPR.2018.00636

Bin, S., & Sun, G. (2022). Research on the Influence Maximization Problem in Social Networks Based on the Multi-Functional Complex Networks Model. *Journal of Organizational and End User Computing*, *34*(3), 1–17. doi:10.4018/JOEUC.302662

Cao, J., Qian, S., Zhang, H., Fang, Q., & Xu, C. (2021). Global Relation-Aware Attention Network for Image-Text Retrieval. *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21)*, 19–28. doi:10.1145/3460426.3463615

Chang S K, Kunil T L. (1981). Pictorial data-base systems. *IEEE Computer*, *14*(11).

Chen, H., Ding, G., & Liu, X. (2020). IMRAM: Iterative Matching With Recurrent Attention Memory for Cross-Modal Image-Text Retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 12655-12663. doi:10.1109/CVPR42600.2020.01267

Deng, J., Dong, R., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 248-255.

Donahue, J., Hen, L. A., & Guada, S. (2017). Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence,* 677-691. doi:10.1109/TPAMI.2016.2599174

Eisenschtat, A., & Wolf, L. (2017). Linking image and text with 2 way nets. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 4601-4611. doi:10.1109/CVPR.2017.201

Fartash, F., Fleet, D. J., & Kiros, J. R. (2018). VSE++: improved visual-semantic embeddings. *British Machine Vision Conference,* 935-943.

Gao, Zhou, & Zhang. (2023). A Transformer-Based Deep Learning Model for Successful Predictions of the 2021 Second-Year La Nia Condition. *Geophysical Research Letters*.

He, K., Zhang, X., & Ren, S. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 770-778.

Hochreiter, S., & Schmidhuber, J. (1997). *LSTM can solve hard long time lag problem*. Advances in neural information processing systems. NIPS.

Hu, Z., Luo, Y., & Lin, J. (2019). Multi-Level Visual-Semantic Alignments with Relation-Wise Dual Attention Network for Image and Text Marching. *International Joint Conference on Artificial Intelligence,* 789-795.

Huang, Y., Wang, W., & Wang, L. (2017). Instance-aware image and sentence matching with selective multimodal LSTM. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2310-2318. doi:10.1109/CVPR.2017.767

Huang, Y., Wu, Q., & Song, C. (2018). Learning semantic concepts and order for image and sentence matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 6163-6171. doi:10.1109/CVPR.2018.00645

Ji, Z., Wang, H., & Han, J. (2019). Saliency-guided attention network for image-sentence matching. *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 5754-5763.

Lee, K. H., Chen, X., & Hua, G. (2018). Stacked cross attention for image-text matching. *Proceedings of the European Conference on Computer Vision (ECCV),* 201-216.

Li, K., & Zhang, Y. (2019). Visual semantic reasoning for image-text matching. *Proceedings of the European Conference on Computer Vision (ECCV)*, 4654-4662.

Li, X., Uricchio, T., Ballan, L., Bertini, M., Snoek, C. G. M., & Bimbo, A. D. (2016). Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys*, *49*(1), 1–39. doi:10.1145/2906152

Liu, C., Mao, Z., & Zhang, T. (2020). Graph structured network for image-text matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 10921-10930.

Ma, S., Wang, Y., & Zhang, S. (2022). Improved Artificial Bee Colony Algorithm for Multimodal Optimization Based on Crowding Method. *Journal of Organizational and End User Computing*, *34*(3), 1–18. doi:10.4018/JOEUC.302661

Messina, N., Falchi, F., Esuli, A., & Amato, G. (2021). Transformer Reasoning Network for Image- Text Matching and Retrieval, *2020 25th International Conference on Pattern Recognition (ICPR),* 5222-5229.

Mikolov, T., Sutskever, I., & Chen, K. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems (NIPS), 3111-3119.

Pan, R., & Vivancos Vicente, P. J. (2023). Evaluation of transformer-based models for punctuation and capitalization restoration in Catalan and Galician. *Procesamiento Del Lenguaje Natural*, *70*, 27–38.

Paolanti, M., Romeo, L., Martini, M., Mancini, A., Frontoni, E., & Zingaretti, P. (2019). Robotic retail surveying by deep learning visual and textual data. *Robotics and Autonomous Systems*, *118*, 118. doi:10.1016/j.robot.2019.01.021

Paolanti, M., Sturari, M., & Mancini, A. (2017). Mobile robot for retail surveying and inventory using visual and textual analysis of monocular pictures based on deep learning. 2*017 European Conference on Mobile Robots (ECMR)*.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference empirical methods in natural language processing (EMNLP).* doi:10.3115/v1/D14-1162

Qu, L. G., Liu, M., & Cao, D. (2020). Context-Aware Multi-View Summarization Network for Image-Text Matching. *Proceedings of the 28th ACM International Conference on Multimedia,* 1047-1055. doi:10.1145/3394171.3413961

Rasiwasia, N., Costa Pereira, J., & Coviello, E. (2010). A new approach to cross-modal multimedia retrieval. *The 18th ACM International Conference on Multimedia,* 251-260.

Ren, S., He, K., & Girshick, R. (2015). Faster-RCNN: Towards real-time object detection with region proposal network. *Advances in Neural Information Processing Systems*, 91–99.

Russell, S., & Norvig, P. (2002). *Artificial intelligence: A modern approach*. Academic Press.

Salman, R. A., Myeongbae, L., Jonghyun, L., Cho, Y., & Changsun, S. (2022). A comparative study of energy big data analysis for product management in a smart factory. *Journal of Organizational and End User Computing*, *34*(2), 1–17. doi:10.4018/JOEUC.291559

Sarafianos, N., Xu, X., & Kakadiaris, A. (2019). Adversarial Representation Learning for Text-to-Image Matching. *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 5814-5824.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Proceedings of the IEEE Transactions on Signal Processing,* 2673-2681.

Vaswani, A., Shazeer, N., & Parmar, N. (2017). *Attention Is All You Need* (Vol. 30). Advances in Neural Information Processing Systems. NIPS.

Wang, Y., Ma, J., Li, M., Tang, X., Han, X., & Jiao, L. (2022). Multi-Scale Interactive Transformer for Remote Sensing Cross-Modal Image-Text Retrieval. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium,* 839-842.

Wang, Z., Liu, X., & Li, H. (2019). Camp: Cross-modal adaptive message passing for text-image retrieval. *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 5764-5773. doi:10.1109/ICCV.2019.00586

Wei, X., Zhang, T. Z., & Li, Y. (2020). Multi-Modality Cross Attention Network for Image and Sentence Matching. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 10941-10950. doi:10.1109/CVPR42600.2020.01095

Wu, J., Wang, L., Chen, C., Lu, J., & Wu, C. (2023). Multi-view inter-modality representation with progressive fusion for image-text matching. *Neurocomputing*, *523*, 1–12. doi:10.1016/j.neucom.2023.02.043

Wu, Y. L., Wang, S. H., & Song, G. L. (2019). Learning Fragment Self-Attention Embeddings for Image-Text Matching. *Proceedings of the 27th ACM International Conference on Multimedia,* 2088-2096. doi:10.1145/3343031.3350940

Yang, L., Feng, Y., Zhou, M., Xiong, X., Wang, Y., & Qiang, B. (2023). Multi-level network based on transformer encoder for fine-grained image–text matching. *Multimedia Systems*, *29*(4), 1981–1994. doi:10.1007/s00530-023-01079-w

Yang, X., Shyu, M.-L., Yu, H.-Q., Sun, S.-M., Yin, N.-S., & Chen, W. (2019, March). Integrating Image and Textual Information in Human–Robot Interactions for Children With Autism Spectrum Disorder. *IEEE Transactions on Multimedia*, *21*(3), 746–759. doi:10.1109/TMM.2018.2865828

Yang, Y., Wu, F., Xu, D., Zhuang, Y., & Chia, L.-T. (2010). Cross-media retrieval using query dependent search methods. *Pattern Recognition*, *43*(8), 2927–2936. doi:10.1016/j.patcog.2010.02.015

Zhang, Y., Jiang, Y., Qi, H., Zhao, M., Wang, Y., Wang, K., & Wei, F. (2022). An Underwater Human–Robot Interaction Using a Visual–Textual Model for Autonomous Underwater Vehicles. *Sensors (Basel)*, *23*(1), 197. doi:10.3390/s23010197 PMID:36616794

Zheng, Z., Zheng, L., Garrett, M., Yang, Y., Xu, M., & Shen, Y.-D. (2020). Dual-path convolutional image-text embeddings with instance loss. *ACM Transactions on Multimedia Computing Communications and Applications*, *16*(2), 1–23. doi:10.1145/3383184

Zhuang, Y. T., Yang, Y., & Wu, F. (2008). Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval. *IEEE Transactions on Multimedia*, *10*(2), 221–229. doi:10.1109/TMM.2007.911822