Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.037463

Article

TECMH: Transformer-Based Cross-Modal Hashing For Fine-Grained Image-Text Retrieval

Qiqi Li[1], Longfei Ma[1], Zheng Jiang[1], Mingyong Li[1,*] and Bo Jin[2]

[1]College of Computer and Information Science, Chongqing Normal University, Chongqing, 401331, China
[2]Institute of Systems and Robotics (ISR), Department of Electrical and Computer Engineering (DEEC), University of Coimbra, Coimbra, Portugal
*Corresponding Author: Mingyong Li. Email: cqnu_lmy@163.com

Received: 04 November 2022; Accepted: 30 January 2023

Abstract: In recent years, cross-modal hash retrieval has become a popular research field because of its advantages of high efficiency and low storage. Cross-modal retrieval technology can be applied to search engines, cross-modal medical processing, etc. The existing main method is to use a multi-label matching paradigm to finish the retrieval tasks. However, such methods do not use fine-grained information in the multi-modal data, which may lead to sub-optimal results. To avoid cross-modal matching turning into label matching, this paper proposes an end-to-end fine-grained cross-modal hash retrieval method, which can focus more on the fine-grained semantic information of multi-modal data. First, the method refines the image features and no longer uses multiple labels to represent text features but uses BERT for processing. Second, this method uses the inference capabilities of the transformer encoder to generate global fine-grained features. Finally, in order to better judge the effect of the fine-grained model, this paper uses the datasets in the image text matching field instead of the traditional label-matching datasets. This article experiment on Microsoft COCO (MS-COCO) and Flickr30K datasets and compare it with the previous classical methods. The experimental results show that this method can obtain more advanced results in the cross-modal hash retrieval field.

Keywords: Deep learning; cross-modal retrieval; hash learning; transformer

1 Introduction

With the development of the Internet, a large number of pictures and text information have emerged on social networks and various platforms. Deep learning has made new progress and application in various fields, and it is no exception in the field of cross-modal retrieval. To explore the potential value of emerging visual and textual data and improve the efficiency of search engines, cross-modal retrieval has recently become a hot research field. Its goal is to use one modality to search for instances of semantically related information in another modality (for example, text query images).

Therefore, how to efficiently and quickly merge and match instances from different modalities has become a key issue in this field.

Benefiting from the development of deep learning, the current cross-modal retrieval method is characterized by using Convolutional Neural Networks (CNNs) in the image module and Recurrent Neural Networks (RNNs) in the text module [1–3]. However, the description extracted by the CNN network for extracting image features can only capture the global generalized features, and important local details in the image cannot be captured. Therefore, in some recent works, the attention mechanism has been widely used [4–7]. However, there is another unavoidable problem with these methods, that is, when faced with a large amount of data, their computations are often very large, and the response in a large-scale retrieval system will be very slow. Among the methods proposed for this challenging task, the Hamming-based hashing method has achieved remarkable success [8–10]. In this method, instances are transformed into binary hash codes, and the similarity between instances is measured by Hamming distance. The smaller the Hamming distance, the higher the similarity, and the larger the Hamming distance, the lower the similarity. The hash-based retrieval system can perform bit operations on hash codes, efficiently calculate the similarity between large-scale candidate sets, and match the similarity with less storage and faster time.

Most existing hashing methods use the co-occurrence information of input image text pairs or the semantic tags marked manually. Fig. 1 shows the way the text queries images of the existing hash method. These tasks represent text by labels and convert cross-modal hash retrieval to multi-label matching. These hashing methods simply rely on matching a pre-specified set of objects, resulting in missing semantic relationships between objects and attributes or numeral information. Different from these methods, our work focuses on the research of efficient cross-modal retrieval of images and texts with fine-grained semantics. Fine-grained semantics can be higher-order relationships between objects (e.g., *'sit'* in *'A man with a red shirt sit at an outdoor table'*) and attributes or numeral information (e.g., *'red shirt'* and *'A'* in *'A man with a red shirt sit at an outdoor table'*). To complete this task, we need to have a deep semantic understanding of images and texts and analyze the high-level semantic relationship between images and texts. Our method combines the transformer encoder with the bottom-up visual attention mechanism. In order to obtain refined visual-semantic information, we utilize the salient regions and objects produced by bottom-up attention, which are not used by traditional hashing methods.

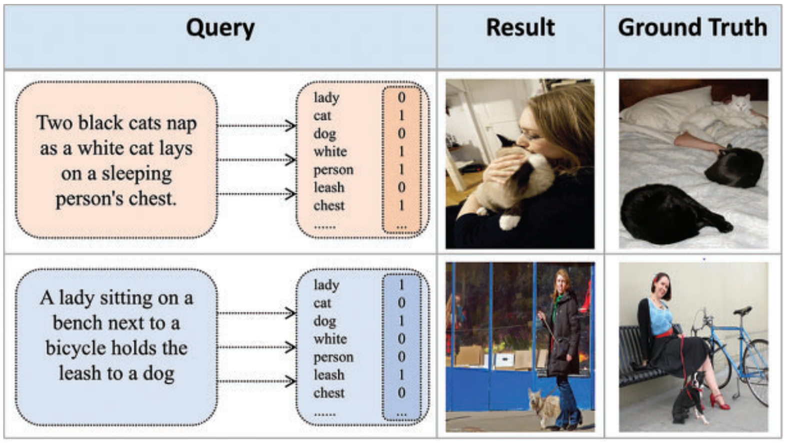

Query
Result
Ground Truth
Two black cats nap as a white cat lays on a sleeping person's chest.
lady
cat
dog
white
person
leash
chest
......
0
1
0
1
1
0
1
...
A lady sitting on a bench next to a bicycle holds the leash to a dog
lady
cat
dog
white
person
leash
chest
......
1
0
1
0
0
1
0
...

Figure 1: Most existing hashing methods lack semantic relational objects and attributes or numerical information, leading to suboptimal results

For these reasons, we propose a transformer-based model that can generate compact and rich information. This model can process image and text data independently, and then match them into the same public space for efficient retrieval in the next step. Using the self-attention mechanism contained in the transformer is used to process image regions just like natural language. The two branches of the model are used to process the image and title respectively, avoiding the communication between the two pipelines, then obtaining the compact image and title description, and finally generating the hash code from the image and title description. This represents that our work is an early attempt at fine-grained cross-modal retrieval in the hash method.

In general, we perform fine-grained matching through the following three key points. The first is the feature extraction stage. Using the bottom-up attention feature of the image, we can extract the region of interest and related position information of the image, and further refine the extraction of image features. In terms of text features, we no longer use multiple labels to represent the text but use BERT to extract features. Second, image and text features use the powerful reasoning ability of the transformer encoder to learn more detailed global features through their respective channels. Third, the training process in the traditional cross-modal hash method tends to make the model biased towards label matching, so we use the datasets (MS-COCO and Flickr30k) of the image text matching task in vision language pre-training (VLP) models to pay more attention to the semantic information of images and texts rather than labels.

The contribution of this paper can be summarized as follows:

1. We point out the problem that most of the current cross-modal hash retrieval methods focus on rough concept relationships and such coarse-grained methods often lead to suboptimal results. We explore the fine-grained semantic matching problem in the current domain, which is an early attempt in the field of cross-modal hashing.

2. We propose a one-step fine-grained hashing retrieval cross-modality that can generate compact and rich information to convert image and text captions into binary hash codes for efficient retrieval. We use evaluation metrics in the field of image-text matching rather than those in the field of hash retrieval. Experimental results show high performance on benchmark datasets (MS-COCO, Flickr30K).

2 Related Work

In this part, we briefly reviewed some previous work about hash learning, and briefly introduced the related content of image text processing.

2.1 Hash Learning

Because of its low storage and high-efficiency retrieval, hash learning can play a very good role in processing large-scale data. Cross-modal hash retrieval for texts and images is a binary code that maps data points in the original feature space of texts and images into a common Hamming space, the similarity is judged by calculating the Hamming distance between the hash code of the query data and the hash code of the data to be queried. The smaller the Hamming distance, the higher the similarity, and the larger the Hamming distance, the lower the similarity. Sort the similarity of the two modal data, and then get the retrieval results, greatly improving the retrieval efficiency. Moreover, because the binary hash code is used to replace the original data storage, the requirement for storage space is greatly reduced. Preserving similarity is the key principle of cross-modal retrieval based on the hash method. Nowadays, we can summarize hash learning methods into two types: shallow hash method and deep learning-based hash method.

Shallow hash methods, such as [11–13], extracted hand-made features through some shallow structures. The Collective Matrix Factorization Hashing (CMFH) proposed by Ding et al. [12] assumes that the same sample in different modes generates the same hash code, and learns different hash codes in the shared potential semantic space. The Semantic Preserving Hashing (SePH) method proposed by Lin et al. [13] converts the semantic affinity of training data into a probability distribution, and it is similar to the hash code to be learned in Hamming space by minimizing Kullback-Leibler Divergence (KL divergence). The features extracted by these methods may not be compatible with the process of hash learning, which may lead to poor results. Compared with the shallow hash method of manually extracting features, the hash method based on deep learning greatly improves the distinctiveness and effectiveness of the extracted features.The Deep Cross-modal Hashing (DCMH) [8] first combined deep learning with hash learning, and proposed an end-to-end learning framework that integrates feature learning and hash learning. After this, many hash methods based on deep learning have been studied successively, such as [9,14–18]. Most of the existing hash methods use the co-occurrence information of input image text pairs or artificially annotated semantic tags. In fact, these tasks are to represent a text by tags and convert cross-modal hash retrieval into multi-tags matching. However, this tag-matching method can't capture the fine-grained semantic information in multi-modal data.

Recently, some researchers tried to use Vision Transformer (ViT) [19] for fine-grained image hash retrieval and achieved good results. These methods [20–22] are single-modal hashing methods for fine-grained retrieval. However, there are few fine-grained cross-modal hashing methods. We believe that fine-grained cross-modal hashing with transformer as the backbone is worth a try.

2.2 Image and Text Processing

Image-text matching in cross-modal retrieval is the intersection of computer vision and natural language processing. Generally, image-text matching is to map images and texts to the same semantic space, and then judge their similarity by distance, such as cosine similarity.

Image and text are processed by two pipes in the model, respectively, and then processed in a specific part of the model. For text processing, many works such as [1,23], process natural language on RNN or Long Short-Term Memory (LSTM). The most classic method for the image is to use the convolutional neural networks (CNNs) for preprocessing. After AlexNet and VGGNet [24,25] were proposed, the performance of computer vision, especially classification, has been further improved. Although these type of CNNs have greatly helped to improve the performance, their problem is that they usually only extract very general descriptions, which are global rather than fine-grained. Therefore for these networks, the fine-grained information needed in the matching task will be lost, leading to low accuracy. Therefore, the follow-up authors further solve these problems at the fine-grained level [26]. For example, [23] proposed to use Region-CNN (R-CNN) to detect and encode image regions and judge the degree of similarity by the similarity score of image region-word pairs. Reference [5] proposed Stacked Cross Attention (SCAN) to align image regions and words.

Transformer architecture [27] has been well applied in natural language processing and can achieve good results in some tasks, such as translation or sentence classification. Among them, BERT [28] shows the powerful role of the attention mechanism in accurately perceiving text context. Recently, some work has been inspired by the powerful transformer-encoder system. In image and text processing, BERT-like processing methods are used, such as Vilbert [7] and Imagebert [4]. Vilbert extends the BERT architecture to a multi-modal two-stream architecture. Two independent branches deal with visual and text input respectively, and the co-attention layer is used to interact with images and text. Imagebert uses the BERT idea and adds 100 times more training samples for pre-training, which significantly improves the model effect. The transformer encoder is used to process images because its attention mechanism can connect different image regions to capture important relationships between objects. In our system, we use the reasoning ability of the transformer encoder to make it work on the image and text pipeline.

3 Proposed Method

3.1 Problem Formulation

Our work is based on the transformer encoder structure, in which both visual and text pipelines depend on the transformer encoder architecture. We take the salient regions in the image as the input of the visual pipeline and the words in the text as the input of the language pipeline. The transformer encoder can reason about these entities without considering their essence. For example, it can be assumed that n regions of the image are expressed as set $I = \{r_1, r_2, \ldots, r_n\}$ and m words of the text are expressed as sequence $C = \{w_1, w_2, \ldots, w_m\}$.

Since both the image and the text pipeline use the self-attention mechanism, we will first introduce the self-attention mechanism, then introduce the feature extraction of image regions and text words in detail, next introduce our whole model, and finally describe the loss function we use to learn.

3.2 Self-Attention

Literally, the attention mechanism is very similar to human attention. The attention of human beings is to scan the whole quickly, and then focus on the part that they are interested in, pay more attention to this part, and get more information about the part that they are interested in. In deep learning, the main purpose of the attention mechanism is to select more critical information about the current task from many goals. The attention mechanism plays a key role in many tasks in natural language processing (such as emotion classification [29], machine translation [30], and so on) and computer vision (such as image classification [31], object detection [31], and so on).

The attention function is described as a mapping from a query to a series of Key-Value pairs. Attention filters and focuses important information from large-scale information. The weight indicates the importance of the information, and the Value indicates the information corresponding to the weight. The calculation of the attention mechanism can be regarded as two processes: the first process is to use Query and Key to calculate the weight coefficient. Specifically, the first step of the first process is to calculate the similarity based on the Query and Key, and the second step is the same. The original score calculated in one step is normalized. The second process performs a weighted summation of Value according to the weight coefficient.

The self-attention mechanism is a variant of the attention mechanism, which reduces the dependence on external information and is better at capturing the internal correlation of data or features. The attention mechanism can be more formally expressed as:

$$Att\,(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right) V \tag{1}$$

Among $Q \in R^{t \times d_k}, K \in R^{s \times d_k}, V \in R^{s \times d_v}$, s and t is the length of the input sequence and adjustment sequence, respectively. $\sqrt{d_k}$ is used to ease the disappearance of the softmax gradient. When Q, K, and V are all calculated from the same input set, self-attention is derived from the general attention mechanism. Under the circumstances, $t = s$ and QK^{T} is a square matrix, which encodes the correlation between each element in the set and all other elements in the set. Then, a simple feed-forward layer with Relu on the vector $Att(Q, K, V)$ is used to further process the vector generated by the self-attention mechanism. Whether it is visual or textual, the transformer encoder self-attention mechanism can discover the hidden relationship between vector entities.

3.3 Feature Extraction

For image feature extraction, we use the pre-trained network and Faster-RCNN [32] with bottom-Up attention [33]. The bottom-up attention feature can usually be used as a substitute for CNN features in the attention-based image and visual question-answering (VQA) models. This method is used to realize the most advanced image captioning performance on MS-COCO. The previous work [33] realized the bottom-up attention model by using the object and attribute annotations of Visual Genome [34] and multi-GPU training of Faster R-CNN based on ResNet-101. Using the bottom-up attention feature can make the visual question answering (VQA) and image captioning achieve remarkable results, so we extract the bottom-Up attention feature from the image as the image description $I = \{r_1, r_2, \ldots, r_n\}$.

For text processing, we use BERT [28] to extract word embedding. BERT adopted the transformer encoder as the language model and used attention to calculate the input and output. It can be fine-tuned by adding an additional output layer to the pre-trained BERT to build advanced models.

3.4 Modal

Our model is based on the transformer encoder structure, in which both image and text pipes depend on the transformer encoder architecture. The transformer encoder takes the sequence or set of entities as input. We treat the salient regions in the image as image entities as input of the image pipeline, and the words in the text as text entities as input of the text pipeline. The transformer encoder can reason about these entities without considering their essence. The n regions of the image are represented as the set $I = \{r_1, r_2, \ldots, r_n\}$ as the input of the image pipeline, and the m words of the text are represented as the sequence $C = \{w_1, w_2, \ldots, w_m\}$ as the input of the text pipeline. Therefore, the reasoning module can operate on n objects of the image set and m objects of the text sequence respectively. In order to express the spatial relationship described by the image, we extracted the coordinates $(x_1, x_2, y_1, \mathrm{y}_2)$ and the area of each area, and the area is the coordinates $(x_1, x_2, y_1, \mathrm{y}_2)$ which are calculated as follows:

$$area = (x_2 - x_1)(y_2 - y_1) \tag{2}$$

Then normalize the coordinates and area to get a set S, put the bottom-Up attention feature x, and splice them together to obtain the bottom-Up attention spatial perception feature X. The splicing process is as follows:

$$X = map(concat(x, S)) \tag{3}$$

$map()$ is a simple $Linear - \mathrm{ReLU} - Linear$, which forwards the splicing information. Image and text features 'out' passing through the transformer encoder are processed by the hash layer, and the hash layer processes the features and uses the tanh () to generate the hash code B in the training process. The hash code is generated as follows:

$$B = \tanh(out - median(\mathrm{softmax}(out))) \tag{4}$$

We set a special token at the beginning of the image collection and text sequence of the model, which is called I-CLS and T-CLS respectively. These two tokens carry global information and are transmitted along two pipelines. Therefore, the number of image regions set becomes n+1, and the number of text word sequences becomes m+1. In every step of reasoning, this information is updated by the self-attention mechanism of the transformer encoder. Finally, I-CLS and T-CLS both carry important global information and are output by the last layer. The model shares the weight before calculating the loss, which can enhance the comparability of the image and the visual abstract representation. Our model is shown in Fig. 2.

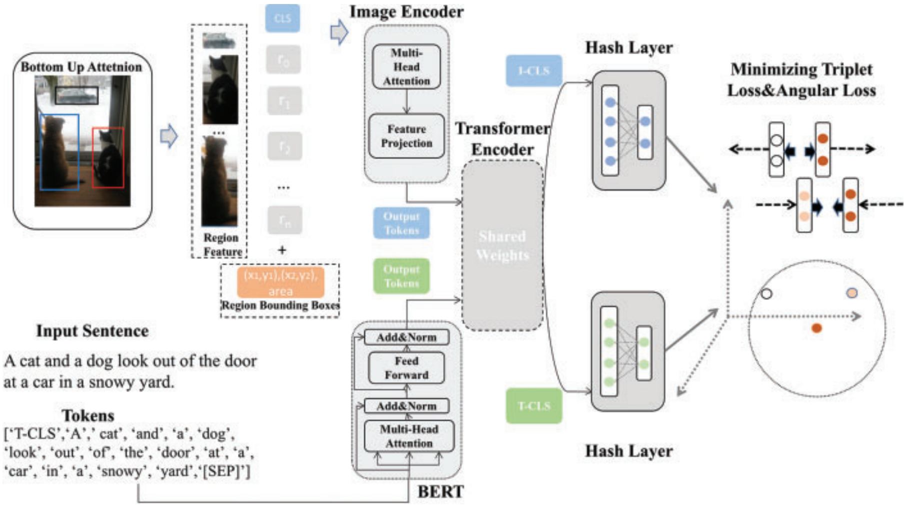


Figure 2: In our model framework, the salient regions of the image are extracted by the bottom-up attention feature based on Faster-RCNN, and the extracted salient region features are spliced with the area and coordinates of each region. This is done by a full connection stack before the image encoder. Words are extracted by BERT. Image Encoder and Text Encoder respectively generate global information of image regions and text words, which are then processed by transformer encoder with shared weights to ensure the consistency of advanced concepts. The output of the image and text channels should be comparable, and it needs to be better connected to the generation of hash codes. We use the transformer encoder with weight sharing for processing to strengthen this constraint. T-CLS is in BERT and I-CLS is initially a zero vector. During training and learning, the transformer encoder's self-attention mechanism updates this information. After passing through the last layer of the transformer encoder, T-CLS and I-CLS carry the global information of text and images and are output by the hash layer, and then calculate the loss

3.5 Loss Function

We use a hinge-based triplet ranking loss [1]. In order to match images and texts in the same space, we define a scoring function with inner product s, scoring function s (i, j) equivalent to the cosine similarity between image and text features. Therefore, our triplet loss function is expressed as:

$$L_{triplet}(i,c) = \max\left[0, m - s(i,c) + s(i,c')\right] + \max\left[0, m - s(i,c) + s(i',c)\right] \tag{5}$$

among m represents the boundary parameter in triplet loss, (i, c) represents a positive pair, c' represents a negative sentence of image I, and i' represents a negative image of text C. The relationship of (i, c, c') and (c, i, i') can be expressed as Fig. 3. c' and i' are calculated by the following:

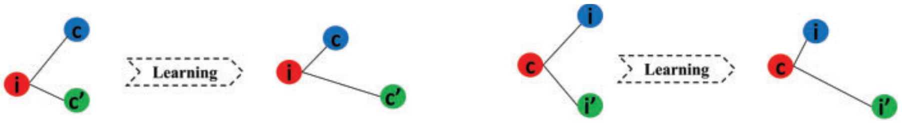


Figure 3: The most relevant caption to image i is c and for caption c, it is image i

$$\begin{aligned} c' &= \underset{d \neq c}{\operatorname{argmax}}\, s\left(i, d\right) \\ i' &= \underset{j \neq i}{\operatorname{argmax}}\, s\left(j, c\right) \end{aligned} \tag{6}$$

Here, considering the performance, we only select hard negatives from one mini-batch, which can obtain better retrieval performance and computational efficiency than all.

In addition, in order to improve the robustness, we also introduce the angular loss [35]. It can get more local structures of triplet triangles than the contrast loss or triplet loss. Compared with the contrast loss, it considers the angle relation, aiming at restraining the angle at the negative point of triplet triangles and improving the robustness of the target to feature changes. Specifically, the angular loss adds the triplet geometric constraint and captures the additional local structure of triplet triangles. Angular loss encodes the triplet relationship within triplet triangles by constraining the angle at the negative point of triplet triangles. Given the same triples, it provides additional constraints to ensure that dissimilar points can be separated. Angle is a measure of invariant rotation and scaling, which makes the target more robust to the large changes of feature mapping in real data. In the original paper of angular loss, the author integrated angular loss and N-pair loss [36]. N-pair loss allows only one positive sample pair for each class. N-pair loss uses all negative samples in the batch to guide gradient update, thereby accelerating convergence. We optimize bi-directional angular loss, using the following formula:

$$f\left(a, p, n\right) = 4\tan^2\alpha\left(a + p\right)n^{\mathrm{T}} - 2\left(1 + \tan^2\alpha\right)ap^{\mathrm{T}} \tag{7}$$

$$L_{angular}\left(i, c\right) = \log\left[1 + \exp\left(f\left(i, c, c'\right)\right)\right] + \log\left[1 + \exp\left(f\left(c, i, i'\right)\right)\right] \tag{8}$$

In which a, p and n respectively represent anchor, positive and negative, α represents the angle boundary parameter. The angle parameter limits the angle of triplet triangles in angular loss. c′ and i′ are hard negatives selected from a mini-batch, which are calculated by the following formulas:

$$\begin{aligned} c' &= \underset{c \neq d}{\operatorname{argmax}} f\left(i, c, d\right) \\ i' &= \underset{i \neq j}{\operatorname{argmax}} f\left(c, i, j\right) \end{aligned} \tag{9}$$

Finally, the angular loss is combined with the hard-negative-based triple loss, which is our formula:

$$L\left(i, c\right) = L_{triplet}\left(i, c\right) + \eta L_{angular}\left(i, c\right) \tag{10}$$

In the following part, we will introduce our experiments and compare the influence of different value settings of hyperparameter η on the model.

4 Experiments

After training the model, we made a comparison with some previous work on the MS-COCO [37] dataset and the Flickr30K dataset [38], and we also did some comparative experiments on our model.

The 2014 edition of Microsoft COCO contains 82,783 training images, 40,504 verification images and 40,775 test images (about 1/2 training, 1/4 verification and 1/4 test). Each picture has 5 captions. We followed the allocation method in [23], using 113,287 pictures for training, 5,000 pictures for validation and 5,000 pictures for testing. The test result is reported by the average of 1,000 test images converted 5 times.

Flickr30K collected 31,783 pictures on the Flickr website, among which each picture has 5 descriptions corresponding to it. Following the segmentation method disclosed in [1,23], we used 1,000 images for validating, 1,000 for testing and the rest for training.

For the evaluation indicators, we believe that the MAP used in the hash retrieval field is based on a multi-label matching scheme. As a fine-grained matching method, we believe that the Recall@topK method in the image-text matching field may be more suitable. K is set to 1, 5, 10, indicating that the fraction of queries for which the ground-truth item is retrieved in the closest 1, 5, 10 points to the query.

4.1 Implementation Details

For the image description that the image pipeline needs to process, we use the 2048-D bottom-up features (36 features per image) that have been extracted from the MS-COCO dataset and the Flickr30K dataset. They can be obtained from here. In the image reasoning part, our model is processed by a stack of four transformer encoder layers with non-shared weights. For the text, we use the 6 hidden layers BERT model pre-trained in the covering language task of English sentences implemented by HuggingFace. We use the PyTorch implementation from here. After the image and text pipes, we use two transformer encoder layers with shared weights to connect.

The feedforward layer of transformer encoders is 2048 dimensions, and the dropout is set to 0.1. Because the image and text pipelines have to pass through the transformer encoder layer with shared weight, their output is set to 1024 dimensions, and the output dimension of the transformer encoder layer with shared weight is 1024 dimensions. m and α are set to 0.2 and 45° respectively, as in [1,35]. We set the batch size to 80 and trained 30 epochs using the Adam optimizer with a learning rate of 0.00001.

4.2 Performance Comparison

We first compare with some advanced open-source cross-modal hashing methods on the two datasets in Table 1. We follow the methods and some results used in [39–42]. The code length of the previous hashing method was set to be the longest code length in the original paper that did not exceed 256 bits because they are a class of coarse-grained methods, and shorter hash codes can achieve good performance, longer code lengths have little impact on the performance of these methods. For a fair comparison, we conduct the experiments with the hash code set to 128 bits compared to the previous hashing method. The more samples in the retrieval database, the lower the corresponding value of Recall@topK(R@K), which is determined by the performance of the model and the calculation method of Recall@topK. Due to the poor performance of previous methods, we only report the test results on 100 samples. It is obvious that our model can achieve much better results than previous hashing methods. DCMH is the first to combine deep learning with hashing learning, which has a great influence on the research of cross-modal hashing. However, DCMH, Cross-modal hamming hashing (CMHH) and Self-constraining and attention-based hashing (SCAHN) are based on multi-label matching, which cannot capture more fine-grained information. The Deep joint-semantics reconstructing hashing (DJSRH) is an unsupervised method, it proposes a joint semantic affinity matrix for input multi-modal instances that carefully integrates raw neighborhood relations from different modalities, thus being able to capture potential inter-instance intrinsic semantic affinity. So the performance of DJSRH and Transformer Encoder Cross-modal Hashing (TECMH) will be better.

Table 1: Comparison with other classical methods under two datasets (MS-COCO, Flickr30K) on 100 samples test set

		MS-COCO			Flickr30K		
Task	Method	R@1	R@5	R@10	R@1	R@5	R@10
I->T	**TECMH (128)**	**59.0**	**88.0**	**94.0**	**40.0**	**70.0**	**80.0**
	DCMH [8]	4.0	17.0	29.0	2.0	6.0	12.0
	CMHH [43]	5.0	8.0	24.0	2.0	6.0	12.0
	SCAHN [44]	19.0	47.0	67.0	3.0	13.0	24.0
	DJSRH [45]	51.0	86.0	93.0	25.0	57.0	68.0
T->I	**TECMH (128)**	**47.0**	78.0	86.0	**30.0**	**58.0**	**69.0**
	DCMH [8]	4.0	14.0	27.0	2.0	8.0	9.0
	CMHH [43]	4.0	15.0	20.0	3.0	9.0	17.0
	SCAHN [44]	17.0	48.0	68.0	3.0	11.0	21.0
	DJSRH [45]	41.0	**85.0**	**93.0**	28.0	55.0	66.0

Then we compare with state-of-the-art fine-grained cross-modal hashing methods, [39] is the only fine-grained method currently. It is a recently proposed fine-grained cross-modal hashing method, but their method is a two-step strategy, the first step is coarse-grained primary screening, and the second is fine-grained re-ranking. For a fair comparison, we only compare with their proposed Fine-grained Re-ranking method (FullRerank). Under the same settings, the comparison results are shown in Table 2.

Table 2: Comparison with the SOTA method

Flickr30K	Image-to-text			Text-to-image		
Method	R@1	R@5	R@10	R@1	R@5	R@10
TECMH (1024)	**63.4**	**87.3**	93.1	**47.8**	**76.4**	**84.7**
FullRerank (1024)	62.3	86.7	**93.4**	43.1	74.0	83.4

We also compare the performance of the models for more hash code lengths on the 1k test set in Table 3, including 64, 128, 256 and 512. The results show that the length of the hash code is crucial for fine-grained hash retrieval. The length of the hash code is longer and the performance of model retrieval can be greatly improved. 16-bit, 32-bit and 64-bit hash codes are too short to represent raw data with fine-grained information well. Compared with the traditional global feature extraction method, in our proposed fine-grained method, the hash code needs to contain richer fine-grained information, so the hash code length is also required it is longer than traditional feature extraction methods, and a longer hash code can achieve better performance.

Table 3: Comparison of different code lengths of our method under two datasets (MS-COCO, Flickr30K) on 1K samples test set

		MS-COCO			Flickr30K		
Task	Method	R@1	R@5	R@10	R@1	R@5	R@10
I->T	TECMH (64)	49.6	83.3	90.4	28.4	60.6	73.5
	TECMH (128)	59.3	88.6	94.7	40.9	70.0	80.3
	TECMH (256)	63.3	91.5	96.2	49.3	78.0	86.8
	TECMH (512)	66.1	91.4	96.2	54.5	81.9	88.8
	TECMH (1024)	**72.4**	**94.9**	**98.1**	**63.4**	**87.3**	**93.1**
T->I	TECMH (64)	38.9	69.7	77.4	22.5	49.2	61.2
	TECMH (128)	47.5	78.3	86.1	30.0	58.1	69.3
	TECMH (256)	51.1	83.2	90.4	36.3	66.1	75.8
	TECMH (512)	53.5	85.4	91.3	40.7	70.8	79.6
	TECMH (1024)	**59.6**	**88.2**	**93.7**	**47.8**	**76.4**	**84.7**

In addition, we also compare our best-performing model with several cross-modal matching methods on the 1k test set in Table 4. Transformer Encoder Reasoning Network (TERN) [6] is an architecture based on the transformer encoder, which can map visual modality and text modality into the same public space, to preserve the relationship between the two modalities. Visual-semantic Embeddings for cross-modal retrieval (VSE++) [1] introduces simple changes to the common loss function for multimodal embeddings, combining fine-tuning and enhanced data usage. Dual-path Convolutional image-text embeddings (DPC) [46] uses two-branch CNN network to extract text-image features. Stacked Multimodal Attention Network (SMAN) [41] uses the stacked multi-modal attention mechanism to take advantage of the fine-grained correlation between images and text. Multi-modality Cross Attention Network (MMCA) [47] jointly models the intra modal and inter modal relationships of image regions and sentence words in a unified depth model. Semantic Concepts and Order for image and sentence matching (SCO) [48] can improve image representation by learning semantic concepts and then organizing them in the correct semantic order. In Table 4, the results tell that our method can show competitive results compared with these excellent models based on continuously embedded image-text matching.

Table 4: Comparison with cross-modal matching methods under two datasets (MS-COCO, Flickr30K) on 1K samples test set

		MS-COCO			Flickr30K		
Task	Method	R@1	R@5	R@10	R@1	R@5	R@10
I->T	**TECMH**	**72.4**	**94.9**	**98.1**	**63.4**	**87.3**	**93.1**
	TERN [6]	63.7	90.5	96.2	53.2	79.4	86.0
	VSE++ [1]	64.6	90.0	95.7	52.9	80.5	87.2
	DPC [46]	65.6	89.8	95.5	55.6	81.9	89.5

(Continued)

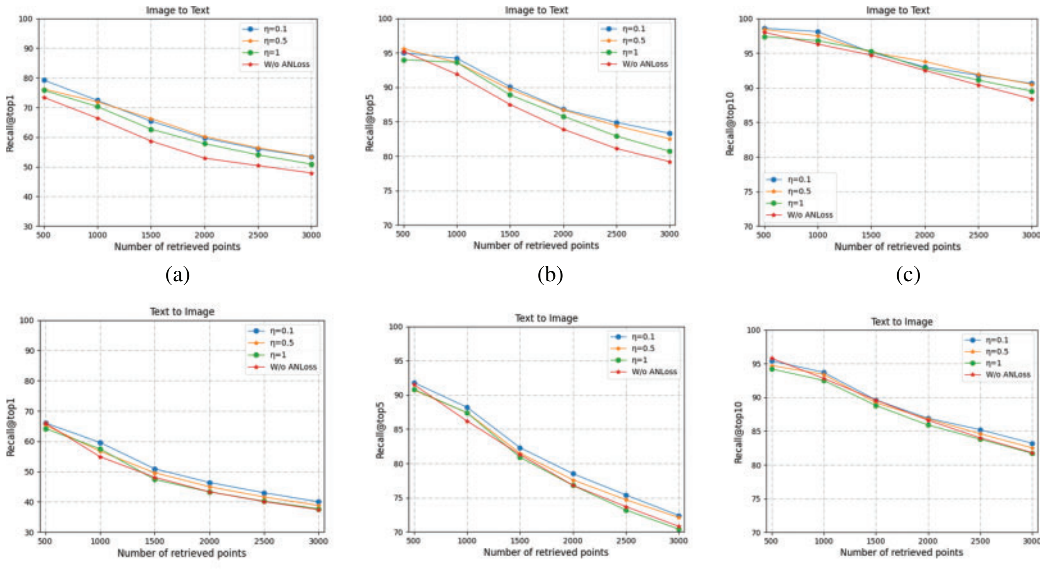
Table 4: Continued

Task	Method	MS-COCO R@1	MS-COCO R@5	MS-COCO R@10	Flickr30K R@1	Flickr30K R@5	Flickr30K R@10
	SMAN [41]	68.4	91.3	96.6	57.3	85.3	92.2
	MMCA [47]	70.4	91.7	96.8	58.1	82.8	90.1
	SCO [48]	69.9	92.9	97.5	55.5	82.0	89.3
T->I	**TECMH**	**59.6**	**88.2**	93.7	**47.8**	**76.4**	**84.7**
	TERN [6]	51.9	84.9	93.6	41.1	71.9	81.2
	VSE++ [1]	52.0	84.3	92.0	39.6	70.1	79.5
	DPC [46]	47.1	79.9	90.0	39.1	69.2	80.9
	SMAN [41]	58.8	87.4	92.0	43.4	73.7	83.4
	MMCA [47]	58.4	87.1	94.0	44.7	72.4	81.1
	SCO [48]	56.7	87.5	**94.8**	41.1	70.5	80.1

4.3 Ablation Experiment

Our loss function consists of triplet loss and angular loss. To study the impact of angular loss on the model, we compared the performance of the model on MS-COCO with different values of η. The experimental results are shown in Fig. 4.

We can see that by only using triplet loss, the performance of the model is not as good as using triplet loss and angular loss (AN loss). The results in Fig. 4 show that using angular loss can further improve the performance of the model, it can be seen from the figure that with the increase of retrieval samples, the performance of the model without the AN loss will decrease more.



(a) (b) (c)

(d) (e) (f)

Figure 4: Recall@topK curves on MS-COCO

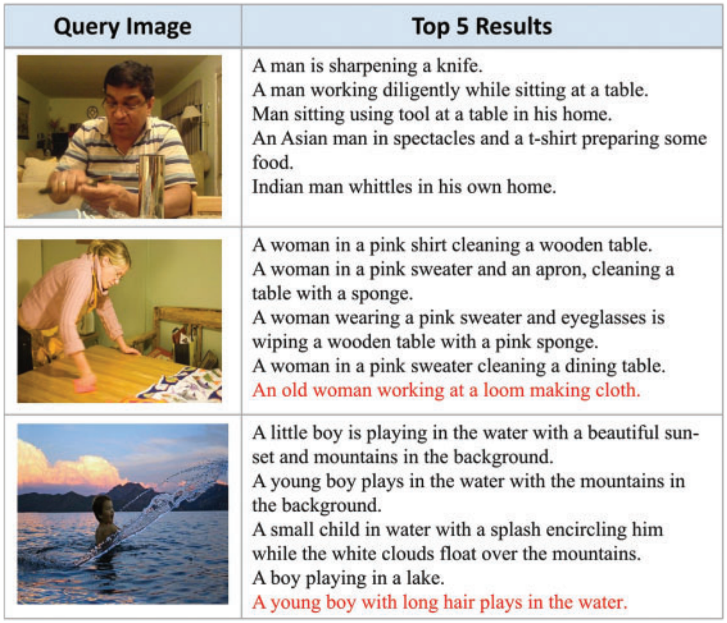
In addition, in order to verify the contribution of fine-grained feature extraction, we compare it with the Resnet-101 pre-trained model for extracting raw image features. The subsequent processing is the same as our original model. We conduct experiments on MS-COCO to verify the performance comparison of the shorter hash code (64 bits) and the longer hash code (1024 bits) when the input is n salient regions (r) and the whole image (w), respectively. We present the experimental results in Table 5. As can be seen from the reported results, adopting the Faster-RCNN to extract the n salient region as the input of the image pipeline is very effective.

Table 5: The impact of different feature extraction methods

MS-COCO	Image-to-text			Text-to-image		
Method	R@1	R@5	R@10	R@1	R@5	R@10
TECMH-r (64)	**49.6**	**82.3**	**90.4**	**38.9**	**69.7**	**77.4**
TECMH-w (64)	35.4	63.0	73.6	27.1	50.7	59.1
TECMH-r (1024)	**72.4**	**94.2**	**98.1**	**59.6**	**88.2**	**93.7**
TECMH-w (1024)	53.5	81.9	90.8	43.2	76.3	84.2

4.4 *Exemples of Retrieval*

In Fig. 5, we show examples of the results of image query text using our model. The displayed results are generated on the Flickr30K dataset. Each image has five sentence descriptions corresponding to it.



Query Image	Top 5 Results
	A man is sharpening a knife. A man working diligently while sitting at a table. Man sitting using tool at a table in his home. An Asian man in spectacles and a t-shirt preparing some food. Indian man whittles in his own home.
	A woman in a pink shirt cleaning a wooden table. A woman in a pink sweater and an apron, cleaning a table with a sponge. A woman wearing a pink sweater and eyeglasses is wiping a wooden table with a pink sponge. A woman in a pink sweater cleaning a dining table. An old woman working at a loom making cloth.
	A little boy is playing in the water with a beautiful sunset and mountains in the background. A young boy plays in the water with the mountains in the background. A small child in water with a splash encircling him while the white clouds float over the mountains. A boy playing in a lake. A young boy with long hair plays in the water.

Figure 5: Examples of the first five texts found in a picture query on the Flickr30K dataset. The black font represents the text description paired with the image

Fig. 5 shows three examples of image query text descriptions. We use red ink to identify the wrong matching result.

In Fig. 6, we show two examples of text query images. We experiment on the MS-COCO dataset. For each query, we show the top-5 retrieval results.

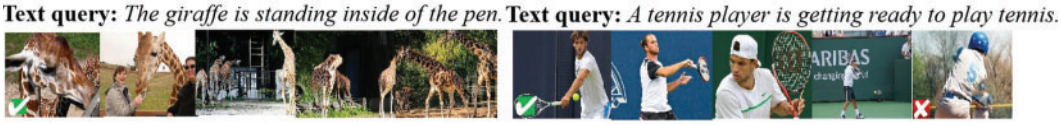

Text query: The giraffe is standing inside of the pen.
Text query: A tennis player is getting ready to play tennis.
ARIBAS

Figure 6: We show the first five results of image retrieval. Ground truth and unreasonable results are marked with green and red respectively

Through the query examples, we can see that our method can find the corresponding information effectively and accurately. In Figs. 5 and 6, we can retrieve almost all matching descriptions.

5 Conclusion

In this paper, we propose a new fine-grained cross-modal hashing method for efficient retrieval. We point out the deficiencies of multi-label matching in the face of complex semantics and fine-grained information. We mainly use the transformer encoders to process the fine-grained information of image and text and then embed the high latitude information of image and text features into the binary code of low latitude. Minimizing triplet loss and angular loss to improve retrieval efficiency and performance. Numerous comparison and ablation experiments demonstrate the effectiveness of our method. In future work, we will study more advanced techniques for more efficient and accurate retrieval.

Funding Statement: This work was partially supported by Chongqing Natural Science Foundation of China (Grant No. CSTB2022NSCQ-MSX1417), the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJZD-K202200513) and Chongqing Normal University Fund (Grant No. 22XLB003), Chongqing Education Science Planning Project (Grant No. 2021-GX-320) and Humanities and Social Sciences Project of Chongqing Education Commission of China (Grant No. 22SKGH100).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Faghri, D. J. Fleet, J. R. Kiros and S. Fidler, "VSE++: Improving visual-semantic embeddings with hard negatives," arXiv preprint, arXiv:170705612, 2017.

[2] R. Kiros, R. Salakhutdinov and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," arXiv preprint arXiv:1411.2539, 2014.

[3] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean *et al.,* "Devise: A deep visual-semantic embedding model," in *Proc. NIPS*, Nevada, USA, pp. 2121–2129, 2013.

[4] D. Qi, L. Su, J. Song, E. Cui, T. Bharti *et al.,* "ImageBERT: Cross-modal pre-training with large-scale weak-supervised image-text data," arXiv preprint arXiv:2001.07966, 2020.

[5] K. -H. Lee, X. Chen, G. Hua, H. Hu and X. He, "Stacked cross attention for image-text matching," in *Proc. ECCV*, Munich, Germany, pp. 201–216, 2018.

[6] N. Messina, F. Falchi, A. Esuli and G. Amato, "Transformer reasoning network for image-text matching and retrieval," in *Proc. ICPR*, Taichung, Taiwan, China, pp. 5222–5229, 2021.

[7] J. Lu, D. Batra, D. Parikh and S. Lee, "ViLBERT: Pre-training task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 13–23, 2019.

[8] Q. Y. Jiang and W. J. Li, "Deep cross-modal hashing," in *Proc. CVPR*, Honolulu, HI, USA, pp. 3232–3240, 2017.

[9] C. Li, C. Deng, N. Li, W. Liu, X. Gao *et al.,* "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 4242–4251, 2018.

[10] V. Erin Liong, J. Lu, Y. -P. Tan and J. Zhou, "Cross-modal deep variational hashing," in *Proc. ICCV*, Venice, Italy, pp. 4077–4085, 2017.

[11] D. Zhang and W. -J. Li, "Large-scale supervised multi-modal hashing with semantic correlation maximization," in *Proc. AAAI Conf. on Artificial Intelligence*, Québec, Canada, vol. 28, 2014.

[12] G. Ding, Y. Guo and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. CVPR*, Columbus, OH, USA, pp. 2075–2082, 2014.

[13] Z. Lin, G. Ding, M. Hu and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. CVPR*, Boston, USA, pp. 3864–3872, 2015.

[14] L. Zhen, P. Hu, X. Wang and D. Peng, "Deep supervised cross-modal retrieval," in *Proc. CVPR*, Long Beach, USA, pp. 10394–10403, 2019.

[15] Q. Lin, W. Cao, Z. He and Z. He, "Semantic deep cross-modal hashing," *Neurocomputing*, vol. 396, no. 2, pp. 113–122, 2020.

[16] G. Mikriukov, M. Ravanbakhsh and B. Demir, "Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing," arXiv preprint arXiv:2201.08125, 2022.

[17] C. Sun, H. Latapie, G. Liu and Y. Yan, "Deep normalized cross-modal hashing with bi-direction relation reasoning," in *Proc. CVPR*, New Orleans, USA, pp. 4941–4949, 2022.

[18] J. Tu, X. Liu, Z. Lin, R. Hong and M. Wang, "Differentiable cross-modal hashing via multimodal transformers," in *Proc. ACM MM*, Lisbon, Portugal, pp. 453–461, 2022.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai *et al.,* "An image is worth 16 × 16 words: Transformers for image recognition at scale," arXiv preprint, arXiv:2010.11929, 2020.

[20] Y. Chen, S. Zhang, F. Liu, Z. Chang, M. Ye *et al.,* "Transhash: Transformer-based hamming hashing for efficient image retrieval," in *Proc. ICMR, Association for Computing Machinery*, New York, NY, USA, pp. 127–136, 2022.

[21] S. R. Dubey, S. K. Singh and W. T. Chu, "Vision transformer hashing for image retrieval," arXiv preprint, arXiv:210912564, 2021.

[22] T. Li, Z. Zhang, L. Pei and Y. Gan, "Hashformer: Vision transformer based deep hashing for image retrieval," *IEEE Signal Processing Letters*, vol. 29, pp. 827–831, 2022.

[23] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. CVPR*, Boston, USA, pp. 3128–3137, 2015.

[24] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, arXiv:14091556, 2014.

[26] C. Sun, C. Gan and R. Nevatia, "Automatic concept discovery from parallel text and visual corpora," in *Proc. ICCV*, Santiago, Chile, pp. 2596–2604, 2015.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2, pp. 5999–6009, 2017.

[28] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint, arXiv:181004805, 2018.

[29] Y. Wang, M. Huang, X. Zhu and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. EMNLP*, Austin, TX, USA, pp. 606–615, 2016.

[30] M. T. Luong, H. Pham and C. D. Manning, “Effective approaches to attention-based neural machine translation,” arXiv preprint, arXiv:150804025, 2015.

[31] S. Woo, J. Park, J. -Y. Lee and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. ECCV*, Munich, Germany, pp. 3–19, 2018.

[32] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[33] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson *et al.,* “Bottom-up and top-down attention for image captioning and visual question answering,” in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 6077–6086, 2018.

[34] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata *et al.,* “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[35] J. Wang, F. Zhou, S. Wen, X. Liu and Y. Lin, “Deep metric learning with angular loss,” in *Proc. CVPR*, Honolulu, HI, USA, pp. 2593–2601, 2017.

[36] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *Proc. NIPS*, Barcelona, ESP, pp. 1857–1865, 2016.

[37] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* “Microsoft COCO: Common objects in context,” in *Proc. ECCV*, Zurich, Switzerland, pp. 740–755, 2014.

[38] P. Young, A. Lai, M. Hodosh and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[39] Y. Li, Y. Mu, N. Zhuang and X. Liu, “Efficient fine-grained visual-text search using adversarially-learned hash codes,” in *Proc. ICME*, Shenzhen, Guangdong, China, pp. 1–6, 2021.

[40] Y. Wu, S. Wang, G. Song and Q. Huang, “Learning fragment self-attention embeddings for image-text matching,” in *Proc. of the 27th ACM Int. Conf. on Multimedia*, Nice, France, pp. 2088–2096, 2019.

[41] Z. Ji, H. Wang, J. Han and Y. Pang, “SMAN: Stacked multimodal attention network for cross-modal image-text retrieval,” *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 1086–1097, 2022.

[42] N. Messina, G. Amato, A. Esuli, F. Falchi, C. Gennaro *et al.,* “Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 4, pp. 1–23, 2021.

[43] Y. Cao, B. Liu, M. Long and J. Wang, “Cross-modal hamming hashing,” in *Proc. ECCV*, Munich, Germany, pp. 202–218, 2018.

[44] X. Wang, X. Zou, E. M. Bakker and S. Wu, “Self-constraining and attention-based hashing network for bit-scalable cross-modal retrieval,” *Neurocomputing*, vol. 400, no. 10, pp. 255–271, 2020.

[45] S. Su, Z. Zhong and C. Zhang, “Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval,” in *Proc. ICCV*, Seoul, Korea (South), pp. 3027–3035, 2019.

[46] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu *et al.,* “Dual-path convolutional image-text embeddings with instance loss,” arXiv e-prints, pp. arXiv–1711, 2017.

[47] X. Wei, T. Zhang, Y. Li, Y. Zhang and F. Wu, “Multi-modality cross attention network for image and sentence matching,” in *Proc. CVPR*, Seattle, USA, pp. 10941–10950, 2020.

[48] Y. Huang, Q. Wu, C. Song and L. Wang, “Learning semantic concepts and order for image and sentence matching,” in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 6163–6171, 2018.