Image Retrieval based on Editable Scene Graph with Contrastive Representation Learning

PHAM Dinh Duy[1,a)] Itthisak PHUEAKSRI[2] Marc A. KASTNER[3]
Yasutomo KAWANISHI[2,1] Takahiro KOMAMIZU[1] Ichiro IDE[1]

Abstract

Traditional Content-Based Image Retrieval (CBIR) systems often have difficulty in capturing similar semantic information inherent in images because they rely on low-level visual features of images. The use of a scene graph, which is a representation of the contents of an image, is a promising way of filling this gap. Traditional CBIR systems suffer from the flexibility of the user's input because the query image is usually fixed and uneditable. If a query as a scene graph can be edited, it will allow more detailed queries or modified elements to fit the users' intents. Based on this idea, we propose an image retrieval framework with editable scene graph for CBIR. This framework enables capturing the user's query intents directly on the scene graph by allowing the edition of its contents by adding, deleting, and replacing objects and/or relationships. Meanwhile, leveraging the scene graph to represent objects and their relationships in an image makes it difficult to compare two images because scene graphs are discrete representations of images. Therefore, simple comparison fails to find relevant images properly. To tackle this problem, we also propose to encode scene graphs into a continuous embedding space using contrastive learning. The proposed framework is evaluated on public datasets, demonstrating promising results in retrieving semantically similar images.

1. Introduction

The number of digital images is increasing tremendously, thanks to the advancements of camera-embedded devices. As a result, image retrieval has become a crucial task to manage such a large number of images. A user's query intent can be in various forms, such as text describing it or an image containing relevant information. Understanding such intent is crucial to realizing practical image retrieval systems. The focus of this paper is related to Content-Based Image Retrieval (CBIR) [1] in which a user provides a query image, and a system returns images containing contents the same or similar to those in the query image. As a variation of CBIR, we assume that users intend to find images that include similar classes of objects and relationships in the query image.

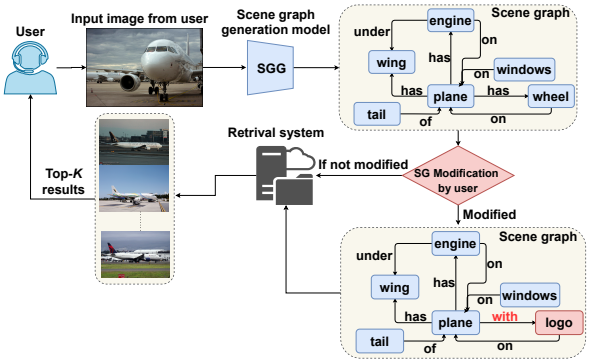


Fig. 1: Overview of the proposed framework. First, an input image is converted into a scene graph, which the user can edit to reflect the query intent. Then, the edited scene graph is passed to the system to retrieve the top-k most similar images based on it.

Traditional CBIR systems focus on low-level descriptors, e.g., color, texture and shape [2], [3], but they fail to capture object relationships. Convolutional Neural Network (CNN)-based methods learn richer features [4], [5], yet still ignore explicit modeling of object interactions, limiting semantic understanding in complex scenes. Scene graph encodes images as object nodes and relationship edges [6], [7], offering richer semantics used in captioning, generation, and Visual Question Answering (VQA), but their retrieval potential remains underexplored.

Current scene-graph-based retrieval systems neither show graphs to users nor allow edits, making it hard to capture the true query intent. To solve this, we propose an editable scene graph for CBIR. As shown in Fig. 1, when the user inputs an image, a scene graph is generated that represents its content, in the example, "plane" and related objects. The user can freely edit the scene graph to query the images that match his/her intent. Here, for example, the user would want images that contain a "plane" along with its "logo", and the edited scene graph could yield the retrieval results shown below, which are more relevant to the user's intent.

Our contributions can be summarized as follows:

[1] Nagoya University
[2] RIKEN
[3] Hiroshima City University
[a)] phamd@cs.is.i.nagoya-u.ac.jp

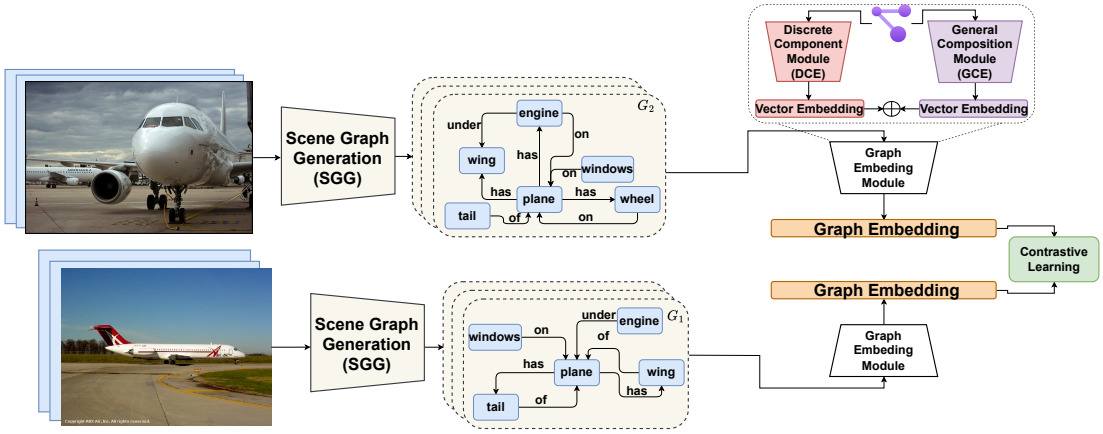


Fig. 2: Overview of contrastive learning: Similar image pairs are converted to scene graph by Scene Graph Generation (SGG) and embedded by the Graph Embedding Module, then contrastive loss aligns the embeddings so that similar pairs become closer.

- **Editable scene graph for CBIR.** We introduce a novel method that allows users to edit a generated scene graph to specify precise query intents.
- **Contrastive representation learning for scene graph.** To address the limitation of discrete graph representation, we introduce a contrastive representation learning to obtain a continuous representation, which enables more meaningful comparisons between graphs.
- **Effective similarity-based matching.** We evaluate our method on public datasets, demonstrating promising results against existing embedding-based CBIR methods.

2. Related Work

Image Retrieval. Traditional CBIR systems focus on low-level features (e.g., Scale-Invariant Feature Transform (SIFT) [8] and color histograms), while CNN-based embeddings [4] improve retrieval but ignore object relations. Scene-graph-based methods model objects and relations [9] and use contrastive learning for robust retrieval [6], [7].

Scene Graph Generation (SGG). Early methods used region based reasoning [10] and bias mitigation [11], Relation Transformer (RelTR) [12] introduced one-stage triplet prediction, and recent cross-modal/self-supervised methods enhance flexibility and scalability [13], [14].

Metric Learning for Graph Embeddings. Contrastive learning aligns scene-graph embeddings for structured similarity [15], [16], with hard negatives [17] and text-based models [18]. While [6], [7] learn scene-graph similarities using Transformers or relative supervision, they do not support user-editable queries. Our method differs by introducing a dual-path embedding trained with Simple Contrastive learning of Sentence Embeddings (SimCSE)-style contrastive loss [19], enabling fine-grained alignment for semantically edited graphs.

3. Proposed Method

The proposed CBIR method is based on editable scene graph and consists of two main components: First, an Image Retrieval Framework based on Editable Scene Graph enables the retrieval of semantically similar images by allowing users to modify the query scene graph. Second, Contrastive Representation Learning for Scene Graph learns embeddings by bringing semantically similar graphs closer and separating dissimilar ones.

3.1 Image Retrieval Framework based on Editable Scene Graph

Our CBIR framework, illustrated in Fig. 1, consists of three main phases:

(1) **SGG.** Given an image I, this module processes it to generate a structured graph representation G as:

$$G = (V, E, \delta, \eta), \quad \delta : V \to L_c, \ \eta : E \to L_r,$$

where V is the set of detected objects, E is the set of their relationships, and δ and η assign class and relationship labels, respectively. L_c and L_r are sets of class and relationship labels, respectively.

(2) **Graph Editing.** Users edit the scene graph G to express their intents by:

- *Relabeling*: Change $\delta(v) \leftarrow \ell_c$ or $\eta(e) \leftarrow \ell_r$.
- *Addition*: Insert a new node $v_n \notin V$ with its δ label or an edge $(v_i, v_j) \notin E$ with its η label.
- *Removal*: Delete node v or edge e.

(3) **Retrieval.** The edited scene graph is mapped to an embedding, then matched against database embeddings via Facebook AI Similarity Search (FAISS) [20].

3.2 Contrastive Representation Learning for Scene Graph

The core of the method is an embedding-based retrieval system shown in Fig. 2 that encodes both database and edited input scene graphs via a graph embedding and contrastive learning.

3.2.1 Graph Embedding

Graph Embedding module consists of two parts: Discrete

Component Embedding (DCE) and General Composition Embedding (GCE).

DCE is calculated by splitting a scene graph G into three text sets: subjects S, objects O, and relations R, where $S = \{\delta(v_i) \mid (v_i, v_j) \in E\}$, $O = \{\delta(v_j) \mid (v_i, v_j) \in E\}$ and $R = \{\eta(e) \mid e \in E\}$. Each set $X \in \{S, O, R\}$ is converted to a concatenated text string. A pretrained language model $\mathcal{M}$ embeds this concatenated text into token level vectors $\mathbf{H}_X = \mathcal{M}(\|_{x \in X} x)$ where $\|$ is a text concatenation operation. A Feed Forward Network (FFN) specific to X (denoted as FFN_X) compresses $\mathbf{H}_X$ into a single vector $\mathbf{h}_X = \mathrm{FFN}_X(\mathbf{H}_X)$. The final DCE embedding of G is the average of $\mathbf{h}_S$, $\mathbf{h}_O$, and $\mathbf{h}_R$ calculated as:

$$\mathbf{h}_{\mathrm{DCE}} = \frac{\mathbf{h}_S + \mathbf{h}_O + \mathbf{h}_R}{3}. \tag{1}$$

Meanwhile, GCE splits the scene graph to triplets $\langle subject, relation, object \rangle$ converted from each edge $e = (v_i, v_j)$ in G as $t = \langle \delta(v_i), \eta(e), \delta(v_j) \rangle$. For set T of triplets, each triplet $t = \langle \delta(v_i), \eta(e), \delta(v_j) \rangle$ in T is transformed into a vector $\mathbf{H}_t$ by using a pretrained language model $\mathcal{M}$ as:

$$\mathbf{H}_t = \frac{\mathcal{M}(\delta(v_i)) + \mathcal{M}(\eta(e)) + \mathcal{M}(\delta(v_j))}{3}. \tag{2}$$

FFN compresses this into $\mathbf{h}_t = \mathrm{FFN}(\mathbf{H}_t)$. Finally, the graph's overall embedding is calculated as the average of all triplet vectors as:

$$\mathbf{h}_{\mathrm{GCE}} = \frac{1}{|T|} \sum_{t \in T} \mathbf{h}_t \,. \tag{3}$$

Embedding of a Graph is obtained from two embeddings of DCE and GCE by concatenating them into a unified embedding, computed as:

$$\mathbf{h}_G = [\mathbf{h}_{\mathrm{DCE}}; \mathbf{h}_{\mathrm{GCE}}], \tag{4}$$

where $[;]$ is the vector-concatenation operation.

3.2.2 Optimization with Contrastive Learning

We optimize the graph embedding module using a contrastive learning inspired by SimCSE [19], which pulls together embeddings of similar scene graphs and pushes apart dissimilar ones. Given a batch $B = \{(G_1^i, G_2^i)\}_{i=1}^{|B|}$ of positive pairs, we compute their L_2-normalized embeddings $\mathbf{h}_{G_1^i}$ and $\mathbf{h}_{G_2^i}$. The similarity logits are calculated as:

$$z_{ij} \;=\; \mathbf{h}_{G_1^i}^{\mathsf{T}} \cdot \mathbf{h}_{G_2^j}, \tag{5}$$

which are scaled by $\overline{z} = \log(1/\tau)$ to obtain $\widehat{z}_{ij} = \overline{z}\, z_{ij}$ with τ= 0.07 as the default Temperature value in SimCSE [19]. For each positive pair (G_1^i, G_2^i), all other $\{G_2^j \mid j \neq i\}$ serve as negatives. The contrastive loss is calculated as:

$$\mathcal{L} \;=\; -\frac{1}{|B|} \sum_{i=1}^{|B|} \log \frac{\exp(\widehat{z}_{ii})}{\sum_{j=1}^{|B|} \exp(\widehat{z}_{ij})}. \tag{6}$$

4. Experiments

This section describes the settings and the evaluation procedure of scene-graph-based image retrieval experiments.

Table 1: Comparison of retrieval performance (Precision@k) between the proposed and baseline methods on two benchmark datasets. Bold font shows the best and underlined font shows the second best.

Methods	Visual Genome [21]			MS COCO [22]		
	P@10	P@20	P@50	P@10	P@20	P@50
$\mathrm{ST}_{\mathrm{MLM}}$ [23]	<u>6.8</u>	<u>15.2</u>	<u>38.0</u>	<u>8.6</u>	**17.2**	<u>40.0</u>
$\mathrm{ST}_{\mathrm{mpn}}$ [24]	5.5	13.5	36.6	7.6	14.2	35.2
BGE-M3 [25]	4.8	13.8	35.2	5.5	14.6	37.0
Proposed	**7.2**	**16.2**	**41.0**	**9.2**	**17.2**	**42.4**

4.1 Experimental Settings

Datasets. We evaluated on Visual Genome (VG) [21] and MicroSoft Common Objects in COntext (MSCOCO) [22] datasets. This setup validates our retrieval system both with manual annotations and automatically generated graphs, demonstrating its robustness and generalizability.

Baseline Methods. To evaluate our framework, we compared against three text-based baselines that embed scene graphs as concatenated triplet sentences generated and scored by Relation TRansformers (RelTR) [12]:

- **Sentence Transformers**: *all-MiniLM-L6* [23] and *all-mpnet-base-v1* [24], hereinafter denoted as $\mathrm{ST}_{\mathrm{MLM}}$ and $\mathrm{ST}_{\mathrm{mpn}}$, respectively. These models are commonly applied in natural language processing tasks to generate compact semantic embeddings for sentences and structured texts.
- **BGE M3-Embedding (BGE-M3)** [25]: A state-of-the-art lightweight text embedding model designed for efficiency and scalability in large-scale retrieval and recommendation tasks.

Implementation Details. We trained for 100 epochs with Adam (learningrate $= 1 \times 10^{-4}$, weightdecay $= 1 \times 10^{-4}$), decaying learning rate by 10 at epoch 50 (batch size 16 on NVIDIA RTX A6000 GPU), and index all embeddings with FAISS [20] for efficient retrieval.

4.2 Evaluation Procedure

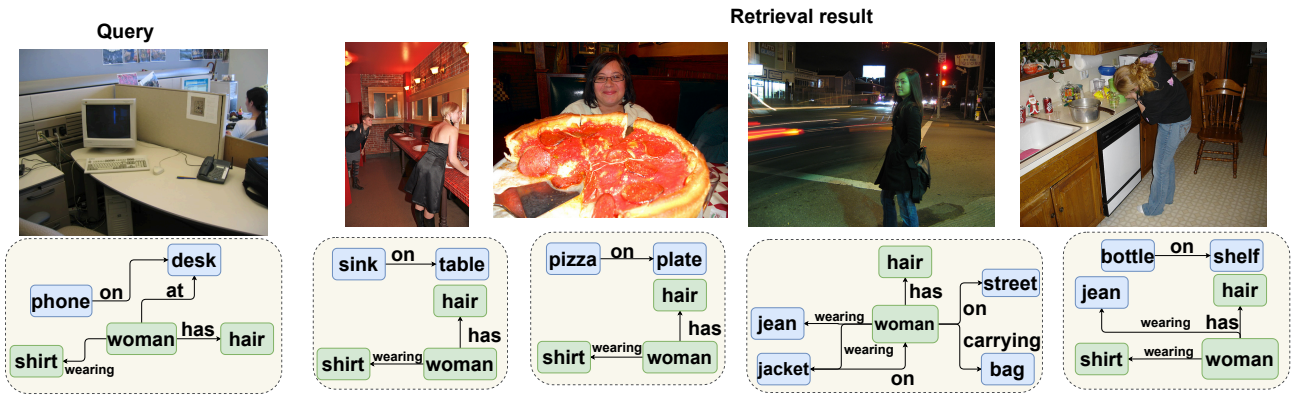
Human evaluation involved ten reviewers. For each scene-graph query, the system generates a ranked list of top-k images. Reviewers mark the images that they judged as semantically relevant to the input scene graph.

The evaluation process includes the following steps:

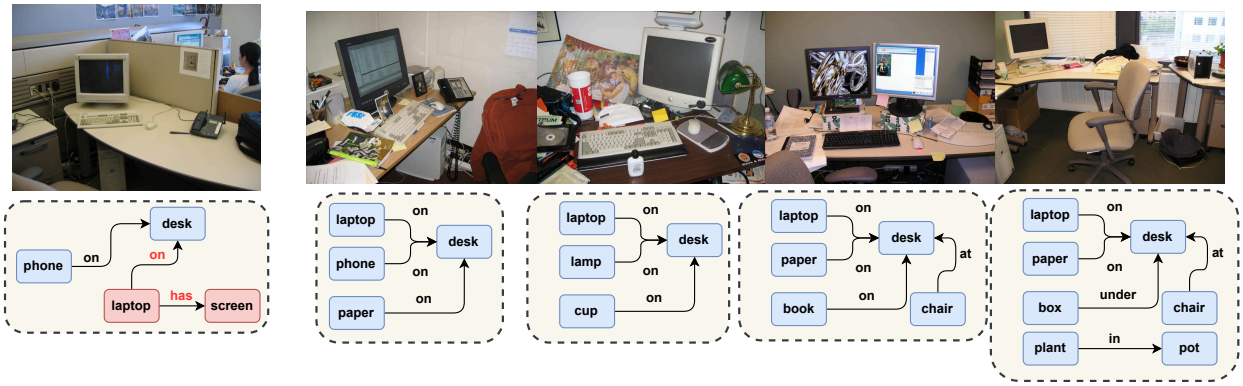
(1) **Query Preparation:** Scene graphs are generated from images in the VG [21] and MSCOCO [22] datasets, used as queries.

(2) **Review Task:** Reviewers evaluate the top-k images retrieved by the system for each query, and then modifies or leaves the scene graph unchanged based on their intents at that moment. They report semantically relevant images based on their relationship to the input scene graph.

Because reviewers could only evaluate a limited set of results, we measure the retrieval quality using Precision@k (P@k), which is the fraction of relevant images in the



(a) Retrieval results by the orginal scene graph.



(b) Retrieval results by the edited scene graph focusing on “desk” and “laptop”.

Fig. 3: Effect of scene graph modification: Since the original scene graph in (a) includes “woman” and relations with it (green nodes), the retrieval result returns images containing “woman” and its relations (green nodes) but not the user’s intention about “desk” or “laptop”. When the user intends to find images related to them, the scene graph can be modified (red nodes) as shown in (b), which returns images related to “desk” or “laptop”.

top-k retrieval results. For each query, we compute $P@k$ per reviewer, average these scores across all of them, and then report the mean over all queries. Here, we report $k \in \{10, 20, 50\}$.

4.3 Results

The results in Table 1 show that our method outperformed all baselines: On VG [21] we achieved 0.4% to 3.0% improvement over $\mathrm{ST}_{\mathrm{MLM}}$ [23], and on MSCOCO [22], up to 2.4% improvement over $\mathrm{ST}_{\mathrm{MLM}}$ [23]. Our method achieved promising results in retrieving semantically similar images. This highlights the system’s ability to align scene-graph embeddings in the latent space and retrieve images that share meaningful contextual relationships.

4.4 Qualitative Analysis of Retrieval based on Edited Scene Graph

Figure 3 shows how editing affects retrieval. The original scene graph (with “desk”, “phone”, and “woman”) returns images focused on the “woman” node shown in Fig. 3(a), even though the input image appears to be more related to “computer” and “desk” because “woman” appears only in the right of the image and is not the primary subject. After editing “desk”, “phone”, and “laptop” as shown in Fig. 3(b), the top-k results shift to those objects, moving the retrieval results closer to the user’s intention. This demonstrates that user-driven scene graph edition can directly refine retrieval, emphasizing input control over representation changes.

5. Conclusion

We proposed a CBIR method with an editable scene graph using graph embeddings to achieve interpretable, relationship-aware image retrieval. Evaluations on VG [21] and MSCOCO [22] datasets showed strong performance when relational context is crucial, though results depend on SGG quality and vocabulary limits. We also provided a mechanism that allows users to freely edit their intents based on scene graph in any instance they wish to retrieve. An evaluation of this impact is subject of future work.

Acknowledgments

This work was supported by JSPS Grants-in-Aid for Scientific Research JP22H03612/JP23K24868 and JP24H00733.

References

[1] T. Kato, "Database architecture for content-based image retrieval," in *Image Storage and Retrieval Systems, Proc. SPIE*, vol. 1662, (Bellingham, WA, USA), pp. 112–123, 1992.

[2] R. Fidel, "The image retrieval task: Implications for the design and evaluation of image databases," *New Rev. Hypermedia Multimed.*, vol. 3, no. 1, pp. 181–199, 1997.

[3] H.-l. Chen, "An analysis of image retrieval tasks in the field of art history," *Inf. Process. Manag.*, vol. 37, no. 5, pp. 701–720, 2001.

[4] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proc. 2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, (Boston, MA, USA), pp. 3668–3678, 2015.

[5] X. Li, J. Yang, and J. Ma, "Recent developments of Content-Based Image Retrieval (CBIR)," *Neurocomputing*, vol. 452, pp. 675–689, 2021.

[6] Y. Cong, W. Liao, B. Rosenhahn, and M. Y. Yang, "Learning similarity between scene graphs and images with Transformers." *Comput. Res. Reposit. arXiv Preprint*, arXiv:2304.00590, 2023.

[7] P. Maheshwari, R. Chaudhry, and V. Vinay, "Scene graph embeddings using relative similarity supervision," in *Proc. 35th AAAI Conf. Artif. Intell.*, (Virtual), pp. 2328–2336, 2021.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[9] B. Schroeder and S. Tripathi, "Structured query-based image retrieval using scene graphs," in *Proc. 2020 IEEE Conf. Comput. Vis. Pattern Recognit.*, (Seattle, WA, USA), pp. 178–179, 2020.

[10] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for scene graph generation," in *Proc. 15th Eur. Conf. Comput. Vis.*, vol. 10, (Munich, Bavaria, Germany), pp. 670–685, 2018.

[11] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proc. 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, (Seattle, WA, USA), pp. 3716–3725, 2020.

[12] Y. Cong, M. Y. Yang, and B. Rosenhahn, "RelTR: Relation TRansformer for scene graph generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11169–11183, 2023.

[13] L. Chen, X. Wang, J. Lu, S. Lin, C. Wang, and G. He, "CLIP-driven open-vocabulary 3D scene graph generation via cross-modality contrastive learning," in *Proc. 2024 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, (Seattle, WA, USA), pp. 27863–27873, 2024.

[14] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li, "Learning to generate scene graph from natural language supervision," in *Proc. 18th IEEE/CVF Int. Conf. Comput. Vis.*, (Montreal, QC, Canada), pp. 1823–1834, 2021.

[15] H. Singh, P. Zhang, Q. Wang, M. Wang, W. Xiong, J. Du, and Y. Chen, "Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality," in *Proc. 2023 Conf. Empir. Methods Nat. Lang. Process.*, (Singapore), pp. 869–893, 2023.

[16] A. Baldrati, M. Bertini, T. Uricchio, and A. Del Bimbo, "Composed image retrieval using contrastive learning and task-oriented CLIP based features," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 20, no. 3, pp. 62_1–24, 2023.

[17] L. Wang, P. Yang, X. Wang, Z. Xu, and Y. Dong, "Scene graph fusion and negative sample generation strategy for image-text matching," *J. Supercomput.*, vol. 81, no. 1, pp. 138–159, 2025.

[18] M. Zhang, O. Vallis, A. Bumin, T. Vakharia, and E. Bursztein, "RETSim: Resilient and Efficient Text Similarity." *Comput. Res. Reposit. arXiv Preprint*, arXiv:2311.17264, 2023.

[19] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive learning of Sentence Embeddings," in *Proc. 2021 Conf. Empir. Methods Nat. Lang. Process.*, (Punta Cana, Dominican Republic), pp. 6894–6910, 2021.

[20] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The FAISS library." *Comput. Res. Reposit. arXiv Preprint*, arXiv:2401.08281, 2024.

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual Genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, pp. 32–73, 2017.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in COntext," in *Proc. 13th Eur. Conf. Comput. Vis.*, vol. 5, (Zurich, Switzerland), pp. 740–755, 2014.

[23] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empir. Methods Nat. Lang. Process.*, (Hong Kong, China), pp. 3980–3990, 2019.

[24] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proc. 2020 Conf. Empir. Methods Nat. Lang. Process.*, (Virtual), pp. 4517–4529, 2020.

[25] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "BGE M3-Embedding: Multi-lingual, Multi-functionality, Multi-granularity text Embeddings through self-knowledge distillation." *Comput. Res. Reposit. arXiv Preprint*, arXiv:2402.03216, 2024.