



DETR-CLIP-BLIP Integrated Framework for Context-Guided Multimodal Knowledge Graph Retrieval

Raihan Atsal Hafizh¹, Kemas Rahmat Saleh Wiharja²

^{1,2}School of Computing, Telkom University, Bandung, Indonesia

*rhnatsal@student.telkomuniversity.ac.id

Abstract

Image search systems that rely on metadata tags or basic captions often miss the semantic relationships and visual details needed for complex queries. This study presents a multimodal knowledge graph framework that combines DETR for object detection, CLIP for attribute classification across 188 dimensions, and BLIP for caption generation. On the COCO 2017 validation set (5,000 images), the pipeline processed 4,906 images successfully, of which 4,243 contained the multiple objects required for relationship extraction. The resulting knowledge graph reached 99.98% relational coverage among multi-object images and 2.94 relationships per image, with 117 of the 188 attributes appearing in the data. Retrieval was evaluated under a self-consistency protocol relevance defined by the system's own top-ranked matches yielding an MRR of 0.96 for keyword-based queries and 0.29 for relationship-based queries, with Recall@50 of 88% for the latter. The framework runs on pretrained models without task-specific training and produces both structured triples and human-readable descriptions, supporting applications such as visual question answering, scene understanding, and context-aware image retrieval.

Keywords: attribute classification; image retrieval; multimodal knowledge graph; object detection; semantic relationships; vision-language models

How to Cite: R. A. Hafizh and K. R. S. . Wiharja, "DETR-CLIP-BLIP Integrated Framework for Context-Guided Multimodal Knowledge Graph Retrieval", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 5, pp. 1301–1312, Oct. 2026, doi: [10.29207/resti.v10i5.7647](https://doi.org/10.29207/resti.v10i5.7647).

Permalink/DOI: <https://doi.org/10.29207/resti.v10i5.7647>

Received: March 13, 2026

Accepted: September 5, 2026

Available Online: October 2, 2026

This is an open-access article under the [CC BY 4.0 License](https://creativecommons.org/licenses/by/4.0/)

Published by [Ikatan Ahli Informatika Indonesia](https://www.iaii.or.id/)

1. Introduction

Social media and digital platforms are now filled with photos and images that have completely changed the way information is created, shared, and consumed. Images have become the primary means of communication today, but the problem remains that it is still difficult to organize, understand, and search for these images based on what they actually contain. The approach commonly used today still relies on manual tags and text descriptions to classify images, but this method has fundamental weaknesses in capturing the meaning of more complex semantics and visual context [1], [2]. For example, when searching for a photo of a person feeding a cat in a park in the afternoon, a conventional keyword-based or visual similarity-based search system will encounter difficulties because it requires a deeper understanding of the interactions between objects, spatial configuration, and situational context contained in the photo.

Knowledge graphs have proven effective as structured representations of various types of knowledge, improving complex reasoning and search processes in various application domains [3], [4], but most implementations still focus on text data. Using multimodal knowledge graphs on visual data is considerably harder because visual data lacks natural structure, requiring us to find meaningful correlations in millions of pixel patterns without clear language clues, which demands advanced multimodal understanding skills [5], [6]. Fortunately, transformer architectures have shown they can process both visual and linguistic information simultaneously [7], [8], making it possible to create multimodal knowledge graphs. However, current systems treat multimodal knowledge graph construction and image retrieval as separate processes, rely on error-prone captions, use syntactic parsing rather than direct visual analysis [9], [10], and cannot understand complex semantic relations in natural language queries [11], [12]. This research addresses these gaps by proposing an integrated framework combining DETR-based object detection

[12], CLIP attribute classification across 188 semantic attributes, and BLIP caption generation [8] to extract

semantic relations directly from visual content and enable context-guided image retrieval.

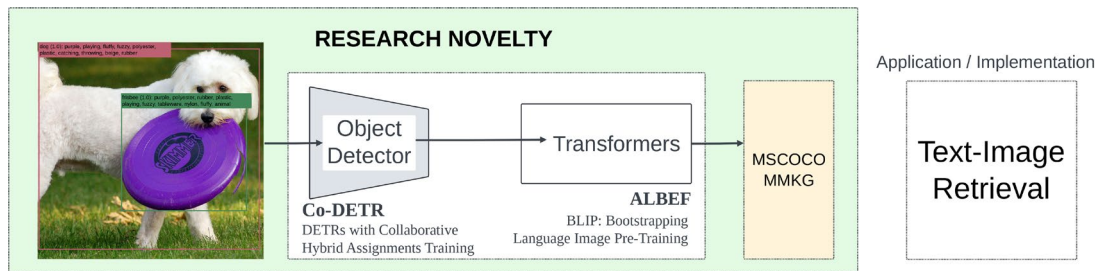


Figure 1. Proposed multimodal knowledge graph construction pipeline

Figure 1 shows the integrated approach proposed in this study, which combines object detection, transformer architecture, and multimodal knowledge graph construction for context-aware image retrieval. The pipeline begins with transformer-based object detection [12], [13] to localize and identify visual entities without manual feature engineering. The detected objects are then enriched through image captioning [8], [14], producing contextually rich descriptions that go beyond object identification to include attributes, activities, and spatial relations. The system maintains direct links between extracted semantic triples and their visual regions, so searches can be performed against structured semantic criteria rather than relying on keyword matching or visual similarity alone. This design supports text-to-image retrieval over the MSCOCO MMKG built by the pipeline.

Recent advances in vision-language pre-training have further accelerated progress in multimodal understanding, with models like ALBEF [15] demonstrating that aligning visual and textual representations before fusion significantly improves cross-modal reasoning, while LXMERT [16] and VL-BERT [17] established that joint pre-training on image-text pairs enables richer semantic grounding across modalities. Beyond global image understanding, research has also explored structured visual analysis such as simultaneous object attribute recognition [18] and unbiased scene graph generation from visual content [19], showing that fine-grained structured representations are achievable. However, these advances have largely been pursued in isolation, which motivates the integrated approach detailed in the next subsection.

The novelty of this work lies in unifying these previously isolated components. While prior systems have explored detection-grounded entities (e.g., DETR-based pipelines), multi-dimensional attribute taxonomies, and caption-driven relation extraction separately, no existing framework integrates all three into a single pipeline that also supports structured semantic retrieval. Each component has been studied in isolation, leaving the end-to-end multimodal knowledge graph construction and retrieval problem largely unsolved. The proposed framework addresses this gap through a two-layer architecture (DETR +

CLIP, plus BLIP) combined with explicit RDF-like triple construction and symbolic retrieval, producing a system that handles both knowledge graph construction and context-aware image search within one pipeline.

2. Methods

This section describes the methodology used to build a multimodal knowledge graph from the COCO dataset and develop an image retrieval system shaped by a query. This approach uses pipeline that converts raw images into structured, rich representations by combining visual detection, attribute classification, and natural language description. The design of the built system prioritizes modularity and scalability, enabling the system to efficiently handle large image datasets without compromising the quality of its semantic annotations.

Figure 2 illustrates the complete architecture of the proposed integrated multimodal knowledge graph framework. The system begins by processing 5,000 COCO dataset images through two parallel layers that work independently but complement each other. Layer 1 uses DETR for object detection and CLIP for attribute classification, producing layer1.json that contains detailed information about detected objects, their bounding boxes, and visual attributes. Simultaneously, Layer 2 employs BLIP with template-based construction to generate structured captions, outputting layer2.json with natural language descriptions of the image content.

These two outputs are then merged in the MMKG Construction stage, where the system extracts semantic relationships between objects and builds RDF-like triples to create a comprehensive knowledge graph stored in combined.json. This multimodal knowledge graph database contains entities, attributes, and relationships that enable sophisticated semantic understanding. Finally, the Retrieval System processes user queries through tokenization, matches them against the knowledge graph using conjunction-based keyword matching, and ranks results based on relevance scores to return the most appropriate images. This end-to-end architecture demonstrates how raw visual data transforms into structured semantic representations that support complex natural language queries

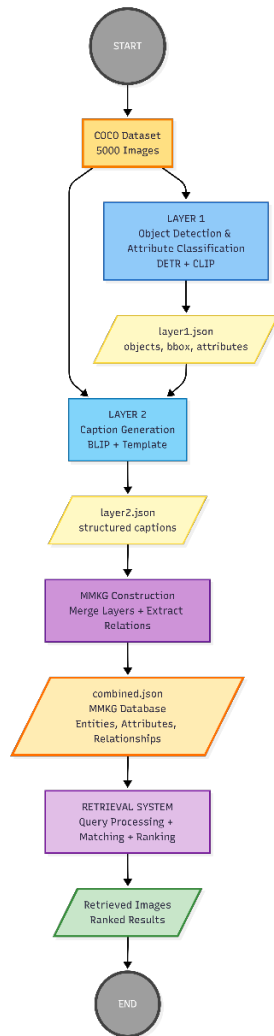


Figure 2. Architecture of the integrated multimodal knowledge graph construction framework supporting context-guided image retrieval

2.1 Dataset Description

This research uses the COCO 2017 Validation Dataset (MS COCO Val2017) [20] as its main data source. COCO is a popular benchmark dataset for object detection, segmentation, and captioning, containing 5,000 validation images that show different kinds of real-world scenes. These images typically have several objects in them, positioned in various ways, with lots of relationships between the objects. The dataset provides annotations that include what objects are in the image, where exactly they're located through bounding boxes, and segmentation masks. This makes COCO a good fit for building and testing multimodal knowledge graphs. We picked the validation split because it follows the standard evaluation approach used in computer vision research, and it has enough image variety to develop a working retrieval system. When applying this framework to other datasets, several components require adjustment. The confidence threshold (currently 0.6) should be tuned based on dataset characteristics: lower values (0.4-0.5) for cluttered images like medical scans, higher values (0.7-0.8) for clear objects. The attribute taxonomy (188 attributes) needs domain-

specific customization, such as health-related terms for medical images or weather conditions for traffic datasets. The DETR model may need fine-tuning if the target domain differs significantly from COCO's 80 categories. Relationship templates should reflect domain-relevant interactions, and batch sizes may require adjustment based on image resolution and computational resources. These modifications ensure effective adaptation while maintaining the core two-layer architecture.

2.2 Layer 1: Object Detection and Attribute Classification

The first layer of the pipeline does two main things, which are detecting objects in the image and assigning semantic attributes to each one. This creates a structured representation that captures both what the objects are and their descriptive details. Figure 3 shows how Layer 1 processing works from start to finish. For detection, we use the DETECTION TRAnsformer (DETR) model [12], specifically the pretrained ResNet-50 version from Facebook Research. DETR treats object detection as a direct set prediction problem using transformer encoders and decoders [13], [21], which means it doesn't need manually designed components like anchor generation and non-maximum suppression. For each input image I, we run DETR through the Hugging Face Transformers implementation, involving preprocessing with DetrImageProcessor to normalize images and then running inference using DetrForObjectDetection.

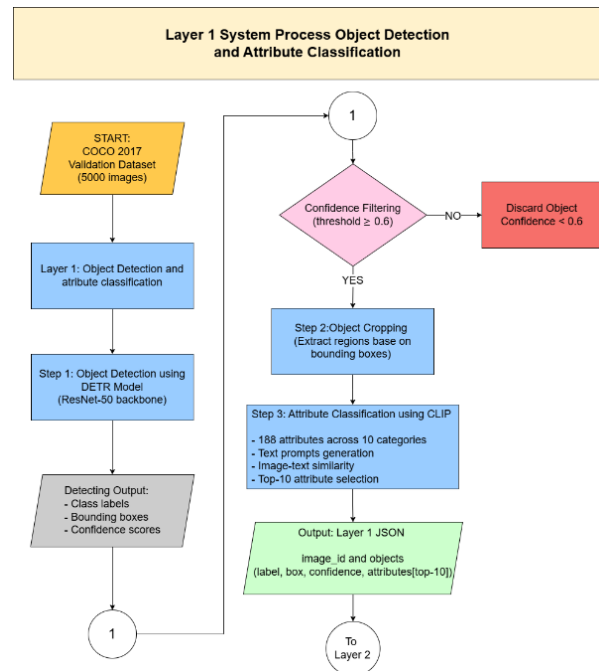


Figure 3. Layer 1 system process workflow showing object detection and attribute classification pipeline

The model gives us predictions including what object it is, where it's located through bounding box coordinates, and how confident the model is about each detection. We filter out anything with a confidence score below 0.6 to avoid false positives, then convert the normalized

coordinates to match actual image size using bilinear interpolation.

After detecting objects, each one goes through attribute classification using the CLIP model [5], which learned to connect visual and textual information by training on 400 million image-text pairs. Similar approaches using vision-language pre-training have demonstrated strong zero-shot classification capabilities [22-24]. CLIP has a dual-encoder setup that can classify things it's never seen before by measuring how similar image embeddings are to text embeddings in a shared space. We built a detailed attribute taxonomy with 188 unique attributes spread across 10 categories covering color (black, white, red, blue), size (tiny to huge), shape (round, square, curved), state and condition (new, old, clean, broken), material (metal, wood, plastic, glass), texture (smooth, rough, shiny, matte), category (animal, person, vehicle), spatial relation (on, under, beside, behind), scene location (indoor, outdoor, urban), and activity (running, eating, sitting, holding). For each detected object, that area gets cropped out from the original image using PIL's crop function with bounding box coordinates. Each attribute is turned into a text prompt following the format "image object or scene attribute", and CLIP figures out how closely the cropped object image matches each of the 188 attribute prompts by comparing their embeddings through cosine similarity. These similarity scores go through normalization using softmax with temperature scaling, as Equation 1 demonstrates.

$$p(a_j | I_{crop}) = \frac{\exp(\text{sim}(f_v(I_{crop}), f_t(t_j))/\tau)}{\sum_{k=1}^M \exp(\text{sim}(f_v(I_{crop}), f_t(t_k))/\tau)} \quad (1)$$

f_v and f_t represents CLIP's visual and text encoders, sim shows cosine similarity, I_{crop} is the cropped object image, t_j refers to the text prompt for a specific attribute, M represents total attributes, while T is the temperature parameter set at 0.07 by default. After calculating these probabilities, the system picks the top 10 attributes with the highest scores using PyTorch's top-k operation, giving enough semantic detail without making computation too heavy. Layer 1 produces output in JSON format containing the image identifier with a list of all detected objects, where each object includes its class based on DETR's vocabulary, bounding box coordinates, detection confidence, and the top 10 attributes ranked by probability. This structured data gets saved as output layer1.json for the next processing steps, as Figure 3 illustrates.

2.3 Layer 2: Natural Language Caption Generation

Layer 2 takes the structured object information from Layer 1 and adds natural language descriptions to it. This creates captions that people can easily read and understand, combining what was detected with contextual information through vision-language models. Figure 4 shows how Layer 2 processes data and how everything gets integrated into the knowledge graph. For generating these captions, Layer 2 uses the BLIP model [8], which stands for Bootstrapping Language-Image Pretraining. BLIP brings something

new to the table with its pretraining approach that builds up vision language representations by working on both understanding and generation tasks at the same time. The model has a unified setup where the visual encoder is shared, but there are separate decoders for understanding and generation. This design lets BLIP do both discriminative tasks and generative tasks effectively. BLIP was trained on massive amounts of image-text pairs from the web, though these pairs were often noisy, so the model includes caption and tag filtering to clean up the data quality.

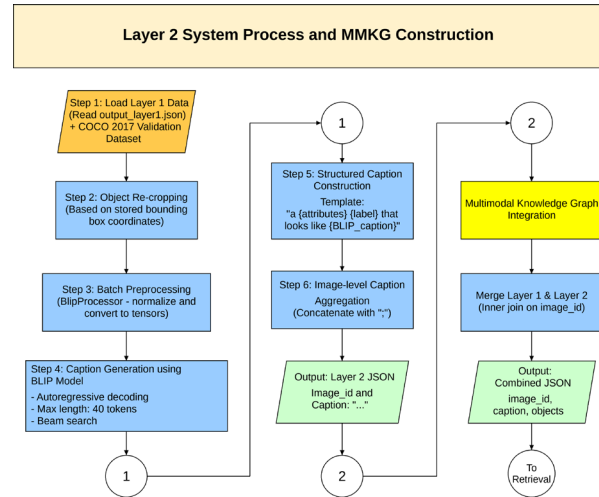


Figure 4. Layer 2 system process and multimodal knowledge graph (MMKG) integration workflow

For each image, the system takes the detection results from the Layer 1 JSON file. Each detected object then gets cropped out from the original image based on its bounding box coordinates. This produces smaller image sections that show just the object plus a little bit of what's around it. The reason for cropping is to make the caption model focus on describing that specific object instead of describing the whole scene. Once the cropping is done, BlipProcessor handles the preprocessing by normalizing the images and turning them into tensors.

To make things run faster, the system groups multiple object crops from the same image together and processes them as a batch instead of one by one. This batch of preprocessed images gets fed into BlipForConditionalGeneration, which uses an autoregressive transformer decoder to create captions word by word, or more technically, token by token. The way BLIP generates these captions follows its standard framework [8], using a cross-entropy objective that can be written as Equation 2.

$$\mathcal{L}_{gen} = -\sum_{t=1}^T \log p(y_t | y_{<t}, I_{crop}; \theta) \quad (2)$$

y_t is the token at position t , and $y_{<t}$ covers all tokens generated before position t . I_{crop} is the cropped object image, θ holds the model's parameters, and T is the total caption length, capped at 40 tokens to keep descriptions informative without inflating compute cost. This keeps descriptions detailed enough without making the computation too expensive. The system combines data

from both layers using templates to create descriptive labels for each object. Three pieces of information are used to create this caption that attributes found in Layer 1, object types from DETR, and descriptions created by BLIP. All information is arranged following the pattern of “a attributes label that looks like BLIP caption”.

2.5 Image Retrieval System

Multimodal knowledge graphs support context-aware image search using keywords that match object labels and semantic attributes. Figure 5 explains how the search process works. When a user enters a query, the system begins by cleaning the text, converting everything to lowercase, and splitting it into separate keywords. Multi word attributes are parsed into individual tokens, which makes matching more flexible. Matching is performed simply by searching for all keywords from the given query simultaneously, and each keyword must appear in one of the attributes or object labels before the image is considered a match. The retrieval process iterates through all images in the knowledge graph database, examining each detected object’s labels and attributes against the query tokens. When a complete match is found where all query keywords are satisfied within at least one object the image identifier is added to the result set. This approach enables the system to retrieve images based on precise semantic criteria rather than visual similarity alone, supporting complex queries that combine object types with specific visual properties like color, size, material, and spatial configuration.

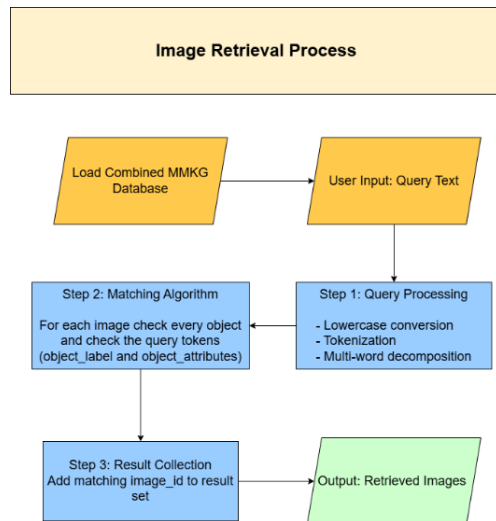


Figure 5. Image retrieval system workflow demonstrating the context-guided text-to-image retrieval process

Formally, this strict matching condition is grounded in set theory, where relevance is defined as Equation 3.

$$R(Q, I) = \exists o \in O_I: Q_{tokens} \subseteq (L_o \cup A_o) \quad (3)$$

In Equation 3, Q tokens is the set of keywords from what someone searched for, while O_I represents all objects detected in a particular image I. L_o stands for the object’s label after it’s been tokenized, and A_o brings together all the tokenized attributes belonging to object o. This strict matching approach achieves high

precision by minimizing incorrect matches, yet it still allows for complex searches that specify both the object type and particular characteristics things like “small red round apple” or “clean white refrigerator”. The search algorithm goes through every image in the knowledge graph, examining each detected object to see if it matches the query. Whenever a match turns up, that image’s identifier gets added to the results. The system then returns all the matching image identifiers, which can be sorted or filtered further if needed. This method takes advantage of the detailed semantic information gathered throughout the entire pipeline, making it possible to search in ways that go well beyond just matching object categories.

2.6 Evaluation Metrics Design

Evaluating the multimodal knowledge graph framework requires two distinct metric categories that address different aspects of system performance. For knowledge graph construction, coverage-based metrics are adopted since the COCO validation dataset lacks ground truth relationship annotations. Relational coverage measures the percentage of multi-object images that successfully yield at least one relationship triple, while relational density quantifies the average number of relationships extracted per image. These metrics assess the breadth and depth of semantic information extraction across the dataset. Additionally, predicate distribution analysis examines which relationship types occur most frequently, and attribute coverage measures the utilization of the 188-dimensional CLIP attribute taxonomy, providing insight into the diversity of semantic representations.

For retrieval performance evaluation, standard information retrieval metrics are employed to assess query effectiveness across different complexity levels. Mean Reciprocal Rank (MRR) measures the ranking quality by calculating the reciprocal of the rank at which the first relevant result appears, averaged across all queries. Precision at K (P@K) evaluates the proportion of relevant results within the top-K retrieved items, while Recall at K (R@K) measures the proportion of all relevant items that appear within the top-K results. F1-Score at K provides a harmonic mean of precision and recall, balancing both metrics. For relationship-based queries, an additional Query Success Rate metric is introduced to measure the percentage of queries that retrieve at least one relevant result, reflecting practical utility for exploratory search scenarios. These metrics are computed at multiple cutoff points (K = 1, 5, 10, 20, 50) to provide comprehensive evaluation across different search depths, with separate assessments for keyword-based semantic queries and relationship-aware queries to capture performance across varying task complexities. The complete quantitative results for all metrics defined above P@K, R@K, and F1@K at K ∈ {1, 5, 10, 20, 50} are presented in Tables 5 and 6 (Section 3.2) for keyword-based and relationship-aware retrieval respectively.

Because COCO Val 2017 does not include query-image relevance annotations, the keyword-based evaluation uses a self-consistency protocol. For each of the 100 queries, candidate images are ranked by token overlap and the top 20% are taken as the relevance set. The MRR, P@K, R@K, and F1@K reported in Tables 5 and 6 therefore measure how consistent the system's ranking is with its own matching criterion, not retrieval quality against human judgments. This limitation is discussed in Section 3.4, and the comparisons in Section 3.3 should be read as paradigm positioning rather than benchmarking.

3. Results and Discussions

3.1 Experimental Results

The implementation of this multimodal knowledge graph construction system was carried out and evaluated using the COCO 2017 validation dataset, which comprises 5,000 images representing various real-world visual scenarios. The system successfully processed all images through both Layer 1 and Layer 2 pipelines, producing comprehensive results that combined object detection with precise bounding box localization, attribute classification based on 188 attributes across 10 semantic categories, and natural text descriptions through template-based construction. For object detection, the system employs DETR with a pre-trained ResNet-50 backbone to identify objects and mark their spatial locations, retaining only detections with confidence scores above 0.6 to minimize false positives and ensure high-quality entity extraction. The processing pipeline demonstrated robust performance across diverse image complexities, from simple single-object scenes to complex multi-object compositions with intricate spatial relationships. These results demonstrate that the proposed framework is capable of scaling to large-scale visual datasets while maintaining semantic richness and structural integrity across the constructed knowledge graph.

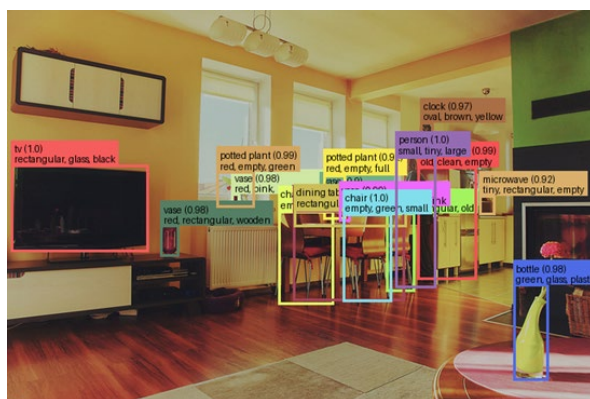


Figure 6. Output object detection from one image of COCO Val 2017

Figure 6 shows the results when analyzing a dining room photo - the system successfully spotted multiple objects like chairs, a dining table, potted plants, a clock, a microwave, a vase, and a bottle. Each object gets a box drawn around it with a confidence score and a list

of characteristics. The attribute classification works through CLIP, assigning descriptive labels that cover colors like red, brown, yellow, and green, states like empty or full, sizes from tiny to large, shapes like rectangular or oval, and materials like glass or wooden. All this information gets stored in output layer1.json, which includes the image ID, objects found, exact coordinates, confidence levels, and sorted attributes. Table 1 shows the system design decisions that balance detection accuracy with computational requirements.

Table 1. Layer 1 System Configuration

Parameter	Value	Description
Dataset	COCO Val 2017	5,000 validation images
Detection Model	DETR ResNet-50	Pretrained object detector
Attribute Model	CLIP ViT-B/32	Zero-shot classifier
Confidence Threshold	0.6	Detection filtering threshold
Attribute Taxonomy	188 attributes	Across 10 semantic categories
Top Attributes Selected	10	Per detected object
Computing Device	CUDA/CPU	GPU-accelerated processing

The BLIP-based caption generator produces context-rich descriptions for each detected object. Instead of processing entire images, the system processes cropped sections showing just the object, which helps generate specific captions that capture subtle visual details and contextual information. The caption generation uses autoregressive decoding with beam search, limited to 40 tokens to keep descriptions informative without being too long. The template-based approach works really well for combining data from both layers into structured captions that merge object attributes, categories, and BLIP descriptions.



Figure 7. Structured caption generation example for a kitchen scene showing two detected objects

Figure 7 shows a kitchen scene with two objects where the oven is described as "a flat oven inside a room located on a shelf, looking like a complete kitchen with a stove and sink" while the refrigerator gets described as "a mediumsized refrigerator inside a room on a shelf, looking like a kitchen with a refrigerator and stove". This shows the system can combine object location like "inside a room" or "on a shelf", physical characteristics like "flat" or "medium size", the object type itself, and scene context from BLIP all in one description. The

template follows a simple format: "an [attribute] [object name] that looks like [BLIP description]". When there are multiple objects in an image, each description gets combined using semicolons as separators. Table 2 summarizes how the caption generation was designed to combine various information sources into coherent descriptions.

Table 2. Layer 2 System Configuration

Parameter	Value	Description
Caption Model	BLIP Base	Pretrained vision-language model
Max Caption Length	40 tokens	Per object description
Template Format	Fixed structure	Integrates attributes, label, caption
Aggregation Method	Semicolon separator	For multi-object scenes
Input Source	Layer 1 JSON	Detection results with attributes
Output File	output_layer2.json	Structured captions

Figure 8 presents an interactive knowledge graph visualization using a force-directed layout to display multi-hop semantic relationships for the query "banana." The central red node (H0) represents the

query object "BANANA," which connects to three categories through 1-hop relationships. Cyan nodes (H1) represent direct attributes such as "yellow" (34 occurrences), "food" (40), and "sticky" (20), connected via HAS ATTRIBUTE edges, indicating banana's visual and semantic characteristics. Yellow nodes (H1) show co-occurring objects, with "person" (41) being most frequent, followed by "bowl" (19) and "orange" (17), connected via CO OCCURS WITH relationships. The green nodes (H2) represent second-hop connections, showing extended relationships such as "unused" and "smooth," which are attributes of the first-hop co-occurring objects. The orange nodes (H3) display third-hop connections, revealing deeper semantic relationships like "white" connected through multi-level attribute chains. Light blue nodes represent individual images (e.g., "000000078959.jpg") containing the queried object. The interface supports dynamic hop selection (1-Hop, 2-Hop, 3-Hop) for controlling visualization depth. The statistics show 95 images retrieved across 53 nodes connected by 52 edges, demonstrating effective multi-hop exploration of multimodal knowledge graph relationships with up to 3 degrees of separation from the query node.

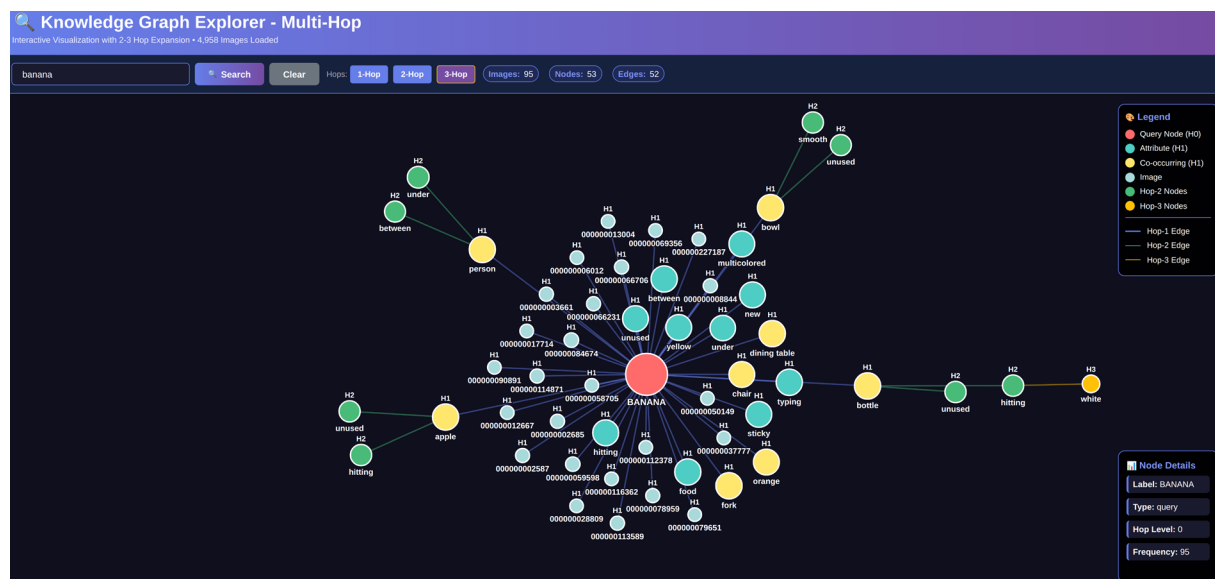


Figure 8. Graph visualization of the knowledge graph query

The final knowledge graph integration combines detection results from Layer 1 with captions from Layer 2 through an inner join operation based on image identity. Everything gets stored in output combined.json, which contains 5,000 entries with image identities, combined captions, and detailed object metadata like labels, bounding boxes, confidence scores, and attributes. Figure 9 shows a network visualization of the knowledge graph structure where objects like dog, motorcycle, and person appear as nodes connected through various relationship types. There are spatial relations like "near", possession relations like "has", and activity relations like "riding", "wearing", "using", and "holding". The visualization shows thumbnails connected to objects with lines

illustrating different relationships position relationships like "close to", ownership like "has", or activities like "ride", "wear", and "hold". What's interesting is the graph looks quite dense and well connected, which means the system stores important information without making random connections. Related objects are grouped together in a way that makes sense.

Table 3 shows how well the system extracts relationships between objects using two important metrics. First is relational coverage, which reached 99.98% of the entire dataset - this means almost all detected object pairs were successfully labeled with their relationships. Second is relational density, averaging 2.94 relationships per image, so the system

finds several semantic relationships in each image rather than just detecting objects separately. The predicate distribution reveals some interesting patterns. Spatial predicates dominate with "next to" appearing 3,754 times, "on" appearing 2,223 times, and "near" appearing 778 times. There are also action-based predicates like "holding" with 1,530 cases and "sitting on" with 652 cases, proving the system captures dynamic relationships and how people interact with objects, not just static positional arrangements.



Figure 9. Network visualization of the multimodal knowledge graph structure showing objects as nodes connected through various relationship types including spatial relations, possession relations, and activity relations

Table 3. Knowledge Graph Relationship Statistics

Metric	Value	Interpretation
Relational Coverage	99.98%	Successfully extracted relationships from object pairs
Relational Density	2.94	Average relationships per image in the knowledge graph
Top 5 Predicates	next to (3754)	Most frequent spatial relationship type
	on (2223)	Second most common positional relation
	holding (1530)	Third most common action-based relation
	near (778)	Spatial proximity relationship
	sitting on (652)	Action and position combined relation

The context-guided retrieval system uses the knowledge graph to let users search for images using text descriptions, especially for specific combinations of objects and their attributes. The keyword matching works with conjunction logic, meaning all words in your query need to match a single object's labels or attributes to keep results precise. We built this as an interactive web app using Streamlit with a clean interface where you just type what you're looking for in natural language and get matching images in a grid layout. Figure 10 shows what happens when searching for "traffic light images" the system found 10 relevant images showing traffic lights in different scenarios like highways, with traffic signs, in front of buildings, and in outdoor settings. What's cool is the system isn't rigid

it recognizes traffic lights despite variations in size, angles, lighting, or composition, with some showing close-ups while others show traffic lights as small details in bigger street scenes. The system handles multi-word attributes by splitting them up, so you can search using full phrases or just parts of it. Testing showed it works well for different search types - simple single-attribute queries bring back relevant images, while detailed multi-attribute queries filter down to more specific results, with real-time result counts letting you tweak searches on the fly.

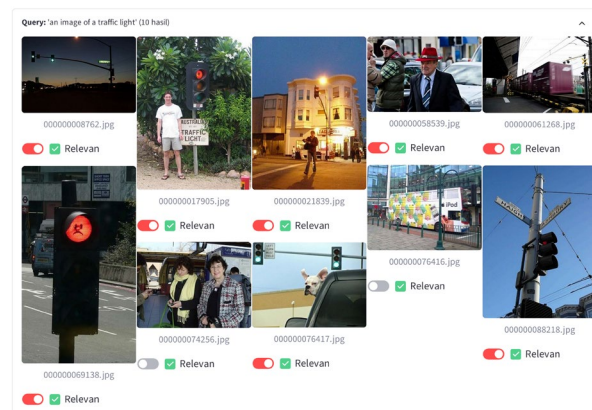


Figure 10. Image retrieval system results for query "an image of a traffic light" showing 10 retrieved images

Achieving relevant results requires more than simply finding matches the system must prioritize and rank them effectively while handling multiple interpretation paths. The scoring mechanism employs a weighted approach where different match types receive different point values based on their semantic importance. When the system identifies an image where objects exhibit the exact relationship described in the query, that match receives the highest score (weighted at 10.0 points per relational match). For queries with disjunctive operators like "or", the system evaluates both interpretation branches and as signs scores based on whichever path yields stronger matches. Matches where objects are found through direct label matching receive moderate scores (2.0 points per object), while contextual keyword matches from attributes or captions receive baseline scores (1.0 point each). This hierarchical weighting ensures that images truly satisfying the relational query rank significantly higher than those with mere keyword overlap. The cumulative scoring allows images to accumulate points from multiple match sources - for example, an image matching one complete relation (10 points) plus having three relevant keywords (3 points) would receive a total score of 13 points, classified as "High Confidence". Images scoring above 10 points are marked as high confidence results, indicating the presence of at least one complete relational structure. Medium confidence results (scores between 5 and 10) typically contain the queried objects without the specified relationships, while low confidence results (scores 5 or below) match only through contextual keywords.

Figure 11 demonstrates this ranking mechanism in action the query "Retrieve images showing a dog next to a person wearing glasses or person who are wearing glasses" successfully retrieved relevant images accommodating both interpretations. The first retrieved image shows three people standing together with a dog positioned at their feet, where at least one person appears to be wearing glasses, satisfying the relational component and accumulating high scores through both structural and attribute matching. The second image displays peoples with a dog in the foreground, capturing the spatial proximity aspect with visible relational structure. The third image captures an outdoor scene with dogs and people in close spatial proximity, demonstrating the system's ability to recognize the "next to" relationship across different visual contexts. The system processed these complex multi-relation queries with logical operators in an average of 0.15 to 0.35 seconds across a dataset of 5,000 images, demonstrating both computational efficiency and practical viability for real-time interactive applications.

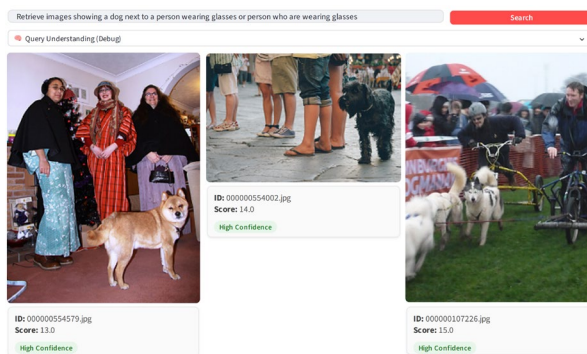


Figure 11. Image retrieval system results for hybrid search mechanism

3.2 Evaluation Metrics Result

Out of 5,000 COCO Val 2017 images, 4,906 (98.12%) produced at least one detection above the 0.6 confidence threshold, while 94 (1.88%) yielded none. Of the 4,906 processed images, 4,243 contain multiple objects the minimum needed for pairwise relationships and 663 contain a single object. The 757 images left out of the relationship analysis split into two groups: 94 detection failures and 663 single-object images that simply cannot form pairs. The detection failures match two known DETR weaknesses [12], [21]: abstract or texture-heavy scenes without clear object boundaries, and objects at extreme scales or under heavy occlusion.

Relational coverage measures the percentage of multi-object images with successfully extracted relationships (99.98% = 4,242 out of 4,243). Relational density gives the average number of relationships per image (2.94). Table 4 summarises the full pipeline statistics.

Table 4 focuses on knowledge graph construction. The retrieval metrics defined in Section 2.6 are reported separately in Tables 5 and 6. The two categories complement each other: coverage and density tell us how completely the system extracts content, while

MRR, P@K, R@K, and F1@K tell us how that content supports text-based search.

Two separate retrieval evaluations were conducted to assess performance across different query complexities. The first evaluation focused on keyword-based semantic retrieval using 100 queries combining object labels with visual attributes (e.g., "red car", "large table"). Table 5 presents the results.

Table 4. Evaluation Metrics Summary

Category	Metric	Value
Dataset Processing	Total Images	5,000 (COCO Val)
	Successfully Processed	4,906 (98.12%)
	Detection Failures	94 (1.88%)
Relationship Extraction	Multi-object Images	4,243 (84.86%)
	Coverage	99.98% (4,242/4,243)
	Density	2.94 per image
Object Detection	Total Triples	~12,480
	Unique Predicates	20+
	Total Objects	33,849
Attribute Classification	Avg per Image	6.90
	Mean Confidence	0.903
	Class Coverage	80/80 (100%)
Knowledge Graph	Taxonomy Size	188 attributes
	Avg Per Object	8.0
	Attributes Actively Used	117/188 (62.23%)
	Nodes	~33,849
	Edges	~12,480
	Density	2.94 edges/image

Table 5. Keyword-based Semantic Retrieval Evaluation

Metric	K=1	K=5	K=10	K=20	K=50
Precision@K	0.96	0.96	0.96	0.91	0.74
Recall@K	0.02	0.12	0.24	0.40	0.55
F1-Score@K	0.05	0.21	0.38	0.55	0.63

On keyword-based queries, the framework reached P@10 of 0.96 and MRR of 0.96 under the self-consistency protocol described in Section 2.6. In practice this means the top-ranked result almost always lands within the system's own top 20% relevance set, indicating stable internal ranking for structured object-attribute queries. Average query time was 0.044 seconds. These numbers should be read together with the methodological caveat in Section 3.4 rather than treated as a direct retrieval-quality benchmark.

The second evaluation assessed relationship-aware retrieval using 100 queries generated from actual knowledge graph relationships (e.g., "person next to motorcycle", "dog sitting on surfboard"). Table 6 shows the performance metrics.

Table 6. Relationship-aware Retrieval Performance

Metric	K=1	K=5	K=10	K=20	K=50
Precision@K	0.15	0.09	0.06	0.04	0.02
Recall@K	0.15	0.45	0.57	0.74	0.88
F1-Score@K	0.15	0.15	0.10	0.07	0.03

Relationship-based queries returned an MRR of 0.29 with 88% of queries finding at least one relevant image, and Recall@50 reached 88%. Precision is naturally lower than for keyword queries, but the recall figure suggests the knowledge graph covers most relationship

patterns present in the data. Query time averaged 0.0013 seconds faster than keyword retrieval because graph traversal exploits indexed structure. The gap between keyword (MRR: 0.96) and relationship (MRR: 0.29) performance is expected: matching a spatial or action relation requires reasoning between entities, while attribute matching reduces to direct token comparison.

3.3 Comparison with Previous Research

This section places the framework alongside two categories of prior work: vision-language embedding models for keyword retrieval (Table 7), and scene graph or knowledge graph systems for relationship retrieval (Table 8). Baseline numbers come from the original publications under their own COCO evaluation protocols. Because those protocols differ from ours in query formulation and ground-truth definition, the tables function as paradigm positioning rather than direct head-to-head comparison.

Table 7. Comparison With VL Embedding-based

System	Method	R@1
CLIP [5]	Contrastive VL	R@1: 58.4%
ALIGN [6]	Large-scale noisy VL	R@1: 60.0%
BLIP [8]	Bootstrapped VL	R@1: 77.6%
BLIP-2 [25]	LLM-augmented VL	R@1: 82.4%
Our System	MMKG-based	R@1: 96.0%

Table 7 reports R@1 for the embedding-based baselines and the proposed framework. The baseline values come from the original publications under the standard COCO 5K caption-image retrieval protocol with human-annotated ground truth, while the 96.0% R@1 for the proposed framework is computed under the self-consistency protocol on 100 attribute-keyword queries described in Section 2.6. The two settings target different retrieval paradigms free-form natural language matching against full images for embedding models, versus structured token matching against object-attribute graphs for the proposed framework. The 96.0% therefore reflects internal ranking stability for structured queries rather than superiority over embedding methods. A hybrid setup combining both paradigms embedding similarity for free-form text and symbolic matching for structured specifications is a natural direction for further work.

For relationship-aware retrieval, the system was compared against scene graph and knowledge graph-based approaches. Table 8 presents the comprehensive comparison.

On the scene graph side, the framework achieved 99.98% relational coverage and 2.94 relationships per image. This addresses the caption-dependency limitation of COCKG [1], which relied on syntactic parsing of caption text and inherited the linguistic biases that come with it. Grounding extraction in DETR detections rather than captions means the resulting triples reflect actual visual content rather than how scenes happen to be described in words.

Recall@50 of 88% for relationship queries is higher than the 46% Recall@100 reported by Schroeder et al.

[26] on COCO-Stuff structured queries, though the query sets and dataset splits differ between the two evaluations and the comparison is therefore indicative rather than strict. The MRR of 0.29 places the first relevant result within the top 3–4 ranked positions on average, which is reasonable given that relationship queries require matching both entities and the relation between them simultaneously.

Table 8. Comparison with scene graph approaches on COCO dataset

System	Method	Performance	Dataset
COCOKG [1]	Caption-based MMKG	Not reported	MS-COCO (Full)
Schroeder et al. [26]	Structured query SG	Recall 46% @K=100	COCO-Stuff (3.1K)
VinVL [27]	Region-based SG	Not reported	MS-COCO
Our System	Direct visual MMKG	99.98% Relational Coverage MRR 0.29, Recall 88% @K=50 2.94 relations/image	COCO Val 2017 (5K)

Compared with VinVL [27], which learns region-based representations end-to-end for downstream tasks such as captioning and visual question answering, the proposed framework instead produces explicit subject-predicate-object triples that can be queried symbolically. This makes the framework more suitable for retrieval tasks where queries decompose cleanly into graph patterns, at the cost of the broader downstream flexibility that learned representations provide.

3.4 Limitations and Future Work

The keyword-based evaluation in Section 3.2 uses a self-consistency protocol because COCO Val 2017 does not include query-image relevance annotations. Under this protocol the relevance set is derived from the system's own top-ranked matches, so the resulting MRR and P@K measure internal ranking stability rather than absolute retrieval quality against human judgments. The Table 7 comparison against embedding-based baselines therefore positions the framework within the broader retrieval landscape rather than benchmarking it head-to-head. A rigorous evaluation would require manually annotated query-image relevance judgments, which we mark as a priority for future work.

Beyond the evaluation protocol, the framework has several technical limitations. There is a clear gap between keyword-based (MRR: 0.96) and relationship-based (MRR: 0.29) retrieval, which reflects the difficulty of spatial and action reasoning but also points to room for improvement in relationship matching. The reliance on pretrained DETR, CLIP, and BLIP means the system inherits biases from their training data, which can affect extraction accuracy for underrepresented concepts or unusual spatial configurations. Keyword retrieval requires exact token matching, so recall drops when users phrase queries with synonyms or variants outside the 188-attribute taxonomy. The absence of ground truth relationship

annotations also makes it hard to validate individual triples beyond coverage. Finally, each image passes through detection, attribute classification, captioning, and relationship extraction in sequence, which limits real-time scalability at large volume.

Future work can address these limitations along several directions. Adding semantic similarity matching with sentence embeddings could relax exact-token retrieval and improve recall on synonym-heavy queries. Relationship validation through geometric constraints or confidence-based filtering could reduce spurious triples. Extending the framework to video would enable temporal relationship modelling and capture how interactions unfold across frames. Linking the graph to external knowledge bases such as ConceptNet or WordNet could enrich relationship semantics and support inference of implicit relations. Incremental update mechanisms would let the graph grow without full reprocessing. End-to-end training across detection, attribute classification, and relationship extraction could tighten the coupling between stages. Most importantly, human-annotated query-image relevance judgments on a representative query set would enable rigorous benchmarking against established retrieval methods.

4. Conclusions

This study introduced an integrated framework that converts images into structured semantic representations by combining DETR object detection, CLIP attribute classification across 188 dimensions, and BLIP caption generation. The contribution lies in unifying these components into a single pipeline that produces both RDF-like triples and natural language descriptions, rather than treating them as separate tasks.

Applied to 5,000 COCO Val 2017 images, the pipeline processed 4,906 successfully, with 4,243 multi-object images contributing to relationship extraction. The constructed knowledge graph reached 99.98% relational coverage among those images, with 2.94 relationships per image on average and 117 of 188 attributes actively used. Retrieval evaluation under a self-consistency protocol gave an MRR of 0.96 for keyword queries and 0.29 with 88% Recall@50 for relationship queries numbers that reflect strong internal ranking and broad coverage rather than absolute retrieval quality against human ground truth. Direct comparison with embedding-based methods on standard protocols remains future work. Building on pretrained models without task-specific training keeps the framework portable across domains, and the dual structured-plus-natural output supports applications in visual question answering, scene understanding, and context-aware image retrieval.

Declarations

The authors provide the following declarations regarding the preparation of this manuscript.

AI Usage Statement

The authors used Claude (Anthropic) solely to clarify ambiguous sentences and check the consistency of reference formatting. The tool was not used to generate, analyze, or interpret any research data or findings. All scientific content, analysis, and conclusions are the authors' own.

Author Contributions

R.A.H.: methodology, software, investigation, formal analysis, writing original draft. R.A.H. designed the framework, implemented the code, conducted the experiments, and prepared the manuscript. K.R.S.W.: conceptualization, methodology, supervision, writing review and editing. K.R.S.W. guided the theoretical and conceptual development of the method and reviewed and corrected the manuscript.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. All work was carried out using the authors' own resources.

Conflict of Interest Statement

The authors declare no conflict of interest. No financial or personal relationships influenced the work reported in this manuscript.

Ethical Clearance

This study did not involve human participants or animal subjects, so ethical approval was not required. All images used are from the publicly available MS COCO dataset, and the authors complied with applicable ethical standards and the dataset's terms of use.

Data and Code Availability

The MS COCO val2017 dataset used in this study is publicly available at <https://cocodataset.org>. The code used to build the framework and reproduce the experiments is available from the corresponding author upon reasonable request.

References

- [1] H. Hu, "COCOKG: A Multimodal Knowledge Graph Building on MSCOCO," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2021)*, Lecture Notes on Data Engineering and Communications Technologies, vol. 89, Q. Xie, L. Zhao, K. Li, A. Yadav, and L. Wang, Eds. Cham: Springer, 2022, pp. 822–829. doi: [10.1007/978-3-030-89698-0_84](https://doi.org/10.1007/978-3-030-89698-0_84)
- [2] 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, no. 1, pp. 32–73, May 2017. doi: [10.1007/s11263-016-0981-7](https://doi.org/10.1007/s11263-016-0981-7)
- [3] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A Survey of Knowledge Enhanced Pre-Trained Language Models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 4, pp. 1413–1430, Apr. 2024. doi: [10.1109/TKDE.2023.3310002](https://doi.org/10.1109/TKDE.2023.3310002)

- [4] Y. Liu, H. Li, A. Garcia-Duran, M. Niepert, D. Onoro-Rubio, and D. S. Rosenblum, "MMKG: Multi-Modal Knowledge Graphs," in Proc. ESWC, 2019, pp. 459-474. doi: [10.1007/978-3-030-21348-0_30](https://doi.org/10.1007/978-3-030-21348-0_30)
- [5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in Proc. 38th Int. Conf. Mach. Learn. (ICML), vol. 139, 2021, pp. 8748-8763. doi: [10.48550/arXiv.2103.00020](https://doi.org/10.48550/arXiv.2103.00020)
- [6] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision," in Proc. 38th Int. Conf. Mach. Learn. (ICML), vol. 139, 2021, pp. 4904-4916. doi: [10.48550/arXiv.2102.05918](https://doi.org/10.48550/arXiv.2102.05918)
- [7] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal Transformer with Multi-View Visual Representation for Image Captioning," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 12, pp. 4467-4480, Dec. 2020. doi: <https://doi.org/10.1109/TCSVT.2019.2947482>
- [8] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," in Proc. 39th Int. Conf. Mach. Learn. (ICML), vol. 162, 2022, pp. 12888-12900. doi: [10.48550/arXiv.2201.12086](https://doi.org/10.48550/arXiv.2201.12086)
- [9] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-Aware Graph Attention Network for Visual Question Answering," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, 2019, pp. 10312-10321. doi: [10.1109/ICCV.2019.01041](https://doi.org/10.1109/ICCV.2019.01041)
- [10] A. Khan, N. Chandran S, and D. R. Gangodkar, "Advanced Deep Learning Frameworks for Cross-Modal Information Retrieval: A Comprehensive Review of Techniques, Challenges, and Future Directions," Wiley Interdiscip. Rev.: Data Min. Knowl. Discov., vol. 16, no. 1, Art. no. e70055, 2026. doi: [10.1002/widm.70055](https://doi.org/10.1002/widm.70055)
- [11] R.-C. Tu, X.-L. Mao, B. Ma, Y. Hu, T. Yan, W. Wei, and H. Huang, "Deep Cross-Modal Hashing With Hashing Functions and Unified Hash Codes Jointly Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 560-572, Feb. 2022. doi: [10.1109/TKDE.2020.2987312](https://doi.org/10.1109/TKDE.2020.2987312)
- [12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in Proc. Eur. Conf. Comput. Vis. (ECCV), Lecture Notes in Computer Science, vol. 12346, Cham: Springer, 2020, pp. 213-229. doi: [10.1007/978-3-030-58452-8_13](https://doi.org/10.1007/978-3-030-58452-8_13)
- [13] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. W. Taylor, and M. Volkovs, "Context-Aware Scene Graph Generation with Seq2Seq Transformers," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15911-15921. doi: [10.1109/ICCV48922.2021.01563](https://doi.org/10.1109/ICCV48922.2021.01563)
- [14] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene Graph Generation with External Knowledge and Image Reconstruction," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1969-1978. doi: [10.1109/CVPR.2019.00207](https://doi.org/10.1109/CVPR.2019.00207)
- [15] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. Hoi, "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, 2021, pp. 9694-9705. doi: [10.48550/arXiv.2107.07651](https://doi.org/10.48550/arXiv.2107.07651)
- [16] H. Tan and M. Bansal, "LXMERT: Learning Cross-Modality Encoder Representations from Transformers," in Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP-IJCNLP), 2019, pp. 5100-5111. doi: [10.18653/v1/D19-1514](https://doi.org/10.18653/v1/D19-1514)
- [17] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "VL-BERT: Pre-training of Generic Visual-Linguistic Representations," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020. doi: [10.48550/arXiv.1908.08530](https://doi.org/10.48550/arXiv.1908.08530)
- [18] K. Pham, K. Kafle, Z. Lin, Z. Ding, S. Cohen, "Learning to Predict Visual Attributes in the Wild," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 13202-13211. doi: [10.48550/arXiv.2106.09707](https://doi.org/10.48550/arXiv.2106.09707)
- [19] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased Scene Graph Generation from Biased Training," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 3716-3725. doi: [10.1109/CVPR42600.2020.00377](https://doi.org/10.1109/CVPR42600.2020.00377)
- [20] 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. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740-755. doi: [10.1007/978-3-319-10602-1_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- [21] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in Proc. Int. Conf. Learn. Represent. (ICLR), 2021. [Online]. Available: <https://openreview.net/forum?id=gZ9hCDWe6ke>
- [22] K. Desai and J. Johnson, "VirTex: Learning Visual Representations from Textual Annotations," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11162-11173. doi: [10.1109/CVPR46437.2021.01101](https://doi.org/10.1109/CVPR46437.2021.01101)
- [23] L. Yao et al., "FILIP: Fine-grained Interactive Language-Image Pre-Training," in Proc. Int. Conf. Learn. Represent. (ICLR), 2022. [Online]. Available: <https://openreview.net/forum?id=cpDhcsEDC2>
- [24] Y. Li et al., "Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm," in Proc. Int. Conf. Learn. Represent. (ICLR), 2022. [Online]. Available: <https://openreview.net/forum?id=zq1iJkNk3uN>
- [25] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in Proc. 40th Int. Conf. Mach. Learn. (ICML), vol. 202, 2023, pp. 19730-19742. [Online]. Available: <https://proceedings.mlr.press/v202/li23q.html>
- [26] B. Schroeder and S. Tripathi, "Structured Query-Based Image Retrieval Using Scene Graphs," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 680-684. doi: [10.1109/CVPRW50498.2020.00098](https://doi.org/10.1109/CVPRW50498.2020.00098)
- [27] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "VinVL: Revisiting Visual Representations in Vision-Language Models," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 5575-5584. doi: [10.1109/CVPR46437.2021.00553](https://doi.org/10.1109/CVPR46437.2021.00553)