Instance-Level Object Retrieval Using CNNs

1. Problem Formulation

We are dealing with an instance-level image retrieval problem.

Given:
- Query Image Patch → A cropped region containing a fine-grained object (e.g., a specific bird species, car model, or brand logo).
- Search Image Set → A large collection of cluttered, high-resolution images containing multiple objects.
- Bounding Box Annotations (for training only) for some search images.

Objective: Retrieve all images from the search set that contain the same object instance as in the query, even if it appears at a different scale, under partial occlusion, or inside cluttered backgrounds.

Formally, this is a metric learning and representation learning task, where we learn a function:
$f\theta: R^{(H\times W\times 3)} \rightarrow R^d$

that maps any input image patch to a feature embedding in R^d, such that:
- Images containing the same instance are close in embedding space.
- Images containing different instances are far apart.

2. Model Architecture

We propose an end-to-end CNN-based retrieval pipeline.

a. Backbone Feature Extractor
- A ResNet-50 or EfficientNet-B3 backbone, pretrained on ImageNet for good generalization.
- Input: Resized query and search patches (224×224)
- Output: High-level feature maps

b. Region Handling for Scale & Occlusion
- Apply a Region Proposal Network (RPN) or Selective Search to focus on the likely location of the object.
- Alternatively, use RoI Align to extract candidate object features.

c. Embedding Layer
- Global Average Pooling over spatial dimensions
- Fully Connected Layer → 512-D embedding
- L2 normalization to ensure embeddings lie on a hypersphere

d. Retrieval Mechanism
- During training: Learn embeddings via a metric-learning loss
- During inference: Extract embeddings for all database images and perform nearest neighbor search (e.g., FAISS).

3. Loss Function Design

We need a loss function that makes the model scale-invariant, occlusion-robust, and resistant to background clutter.

a. Triplet Loss with Hard Negative Mining
L = max(0, ||f(q)-f(p)||^2 - ||f(q)-f(n)||^2 + m)

Where:
- q = query image
- p = positive (same object instance)
- n = negative (different object)
- m = margin hyperparameter

b. Data Augmentation for Invariance
- Random Resizing → teaches scale invariance
- Random Erasing / Cutout → simulates occlusion
- Background Blurring → reduces background dominance

c. Multi-Similarity Loss (Optional)
Can be used instead of Triplet Loss for richer similarity constraints.

4. Evaluation Strategy

a. Dataset Split
- Training: Search images with bounding box annotations
- Validation: Hold-out set from annotated images
- Test: All unannotated search images + queries

b. Metrics
- mAP (Mean Average Precision)
- Recall@K
- Precision-Recall Curve

c. Retrieval Procedure
1. Compute embeddings for all search images
2. Compute embeddings for query patches
3. Use cosine similarity or Euclidean distance for ranking
4. Evaluate retrieval performance.

5. Efficiency Improvements (Optional)

- Dimensionality Reduction: PCA (512 → 128 dimensions)
- Approximate Nearest Neighbor Search: FAISS, Annoy
- Pre-Computed Embeddings: Store embeddings offline
- Hashing Methods: Product quantization or deep hashing.

6. Conclusion

We framed the problem as an instance-level retrieval task and proposed a CNN-based embedding model trained with triplet loss and hard negative mining. The model handles variations in scale, occlusion, and clutter. With region-aware processing, strong feature extraction, and retrieval-friendly embeddings, the system is scalable for large image databases while maintaining retrieval accuracy.