Assignment 1

24M0311 Rubel Rajbanshi | 24M0324 Ayush Sharma | 24M0315 Rita Mahato

1 Problem Formulation

We begin by defining the data and the task. Let the *query patch* be denoted by q, which is a cropped image region of size $h_q \times w_q \times 3$ containing the object of interest. The query patch might be much smaller than a typical search image and focuses on a single object instance.

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ denote the set of all search images in the database. Each search image I has dimensions $H \times W \times 3$ and may contain multiple objects and complex backgrounds. Some search images (for training) have bounding box annotations indicating the location of the same object as in the query. These bounding boxes are used only for supervision during training.

Our goal is two-fold:

1. Produce a ranked list $\mathcal{R}(q)$ of images from $\mathcal{I}$ such that images containing the same object as q are ranked higher.

2. For each retrieved image, produce a spatial *heatmap* H indicating the likely location(s) of the object instance.

The retrieval process is formalized as learning two functions:

$$f_q : q \to \mathbb{R}^d, \quad f_I : I \to \mathbb{R}^d$$

which map the query and search images into a shared embedding space of dimension d. A *similarity function* $s(\cdot, \cdot)$ measures how closely related two embeddings are. We use cosine similarity:

$$s(a, b) = \frac{\langle a, b \rangle}{\|a\|_2 \|b\|_2} \tag{1}$$

The objective is:

$$s(f_q(q), f_I(I^+)) > s(f_q(q), f_I(I^-)) + m \tag{2}$$

where I^+ is a *positive* image containing the object instance, I^- is a *negative* image without it, and $m > 0$ is a margin.

2 Model Architecture

The architecture extracts multi-scale features from both the query and the search image, compares them spatially, and fuses the results.

2.1 Feature Extraction

We use a convolutional backbone (e.g., ResNet) with an FPN. For an input image I, the backbone+FPN outputs feature maps at multiple pyramid levels:

$$X^l = F^l(I) \in \mathbb{R}^{C \times H_l \times W_l}$$

where l indexes the pyramid level (e.g., $l \in \{3, 4, 5\}$), C is the number of channels, and H_l, W_l are the spatial dimensions at that level. The finest level has the largest H_l, W_l and smallest stride; the coarsest level has the smallest H_l, W_l and largest stride.

For the query patch q, the same backbone+FPN produces:

$$Z^l = F^l(q) \in \mathbb{R}^{C \times h_l \times w_l}$$

where h_l, w_l are the query feature dimensions at level l.

2.2 Dense Cross-Correlation

At each level l, we want to measure how well the query's features match different locations in the search image's features. We first apply 1×1 convolutions ϕ and ψ to Z^l and X^l to align feature dimensions or adapt them for correlation.

The dense cosine cross-correlation heatmap at level l is:

$$H^l(u, v) = \frac{\langle \phi(Z^l), \psi(X^l_{u:u+h_l, v:v+w_l}) \rangle}{\|\phi(Z^l)\|_2 \cdot \|\psi(X^l_{u:u+h_l, v:v+w_l})\|_2} \tag{3}$$

Here:

- $H^l(u, v)$ is the similarity score when the query's feature map is aligned at position (u, v) of the search image's feature map.
- $X^l_{u:u+h_l, v:v+w_l}$ is the spatial crop of X^l matching the size of Z^l.

2.3 Multi-Scale Fusion

Since each H^l has different spatial resolution, we upsample each to match the finest resolution (largest spatial size), denoted $H^l_\uparrow$.

We fuse the upsampled heatmaps with a weighted Log-Sum-Exp:

$$H(u, v) = \frac{1}{\gamma} \log \left(\sum_l \alpha_l \exp(\gamma H^l_\uparrow(u, v)) \right) \tag{4}$$

where:

- α_l is the weight for level l, learned during training.
- γ is a temperature controlling the ”softness” of the fusion; large γ approximates a max, small γ averages.
- $H(u, v)$ is the fused multi-scale heatmap in the search image coordinate space.

2.4 Region Pooling and Similarity Scoring

From the fused heatmap H, we take the top-K peaks as candidate locations for the object. For each peak, we extract a region of interest (ROI) from the corresponding search image feature maps (using ROIAlign). This gives a set of *region descriptors* $\{r_k\}$, where each:

$$r_k \in \mathbb{R}^C$$

is L2-normalized.

We also compute a *global query descriptor* z by applying Generalized Mean (GeM) pooling to the highest resolution query feature map Z^{l^*}, followed by an MLP and L2 normalization:

$$z \in \mathbb{R}^C$$

The final image-level similarity is computed by a soft aggregation of query–region similarities:

$$s(I|q) = \mathrm{LSE}_k \left(\langle z, r_k \rangle \right) \tag{5}$$

3 Loss Function

The total loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{NCE}} + \lambda_2 \mathcal{L}_{\text{dense}} + \lambda_3 \mathcal{L}_{\text{trip}} + \lambda_4 \mathcal{L}_{\text{scale}} \tag{6}$$

where each term is:

3.1 Global Contrastive Loss (InfoNCE)

We compute global descriptors g_q for the query and $g(I)$ for each search image (using GeM pooling over a chosen feature map), then:

$$\mathcal{L}_{\text{NCE}} = -\log \frac{\exp(s(g_q, g^+)/\tau)}{\exp(s(g_q, g^+)/\tau) + \sum_j \exp(s(g_q, g_j^-)/\tau)} \tag{7}$$

where:

- g^+ is the global descriptor of a positive image.
- g_j^- are descriptors of negative images.
- τ is a temperature parameter.

3.2 Dense Matching Loss

We supervise H against a ground-truth mask (from bounding boxes) using focal loss + Dice loss to handle imbalance and encourage sharp peaks.

3.3 Triplet Region Loss

We enforce that the best-matching region in a positive image has higher similarity to the query than the best-matching region in a negative image:

$$\mathcal{L}_{\text{trip}} = \left[m + \max_k s(z, r_k^-) - \max_k s(z, r_k^+) \right]_+ \tag{8}$$

3.4 Scale Consistency Loss

We encourage per-level heatmaps to agree with the fused heatmap H via KL divergence.

4 Evaluation Strategy

- **Retrieval:** mean Average Precision (mAP), Recall@K.
- **Localization:** AP at IoU 0.5.
- **Stress Tests:** scale variation, occlusion, clutter.
- **Efficiency:** retrieval latency, index memory footprint.

5 Conclusion

We described a multi-scale dense matching architecture for robust instance retrieval, defined all variables explicitly, and proposed a multi-task loss that balances retrieval and localization objectives. The approach is scalable to large datasets when integrated with coarse-to-fine retrieval.