Supplementary Materials

Kaiwen Cai[1]**, Chris Xiaoxuan Lu**[2]**, Xingyu Zhao**[3]**, Wei Huang**[4]**, Xiaowei Huang**[1]

[1]University of Liverpool, [2]University College London, [3]University of Warwick, [4]Purple Mountain Laboratories

1 Implementations

1.1 Network Architectures

Fig. 11 illustrates the network architectures of the three image retrieval models used in our experiments:

- The **Deterministic** model is composed of ResNet-50 (He et al. 2016) backbone, a GeM layer (Radenović, Tolias, and Chum 2018), a fully connected layer and a L2-normalization layer. The Deterministic model is trained with the triplet loss (Schroff, Kalenichenko, and Philbin 2015) (the loss function will be explained soon).
- The **MCD** model is the same as the Deterministic except that we apply dropout to all conventional layers of the backbone during both training and testing time.
- The **BTL** model has an additional variance head compared to the Deterministic model, and it is trained with the Bayesian triplet loss (Warburg et al. 2021) (the loss function will be explained soon).
- The **Deep Ensemble** is built with 5 Deterministic models that are trained from different initial weights.

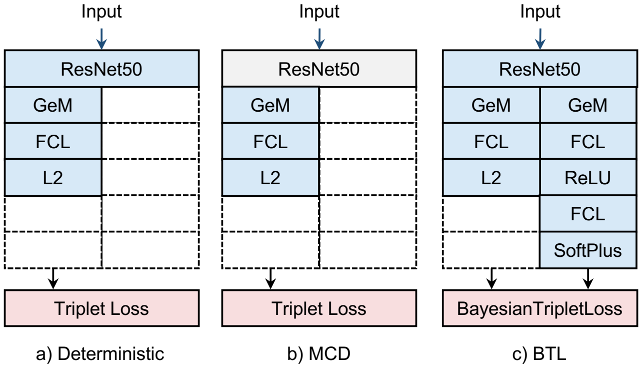

Figure 11: The network architectures of different image retrieval models.

1.2 Bayesian Triplet Loss

Bayesian triplet loss was proposed in (Warburg et al. 2021), and we include it here for completeness. In this setting, embeddings are modeled as a Gaussian distribution $\mathcal{N}(f_\phi(X), f^2_\sigma(X))$. Denote embeddings of query, positive, and negative images as a, p, and n, respectively. The probability that a query is closer to a positive than a negative is

$$P(\|a-p\|^2 - \|a-n\|^2 < -m), \tag{15}$$

where m is a predefined margin. eq. (15) is equivalent to

$$P(\tau < -m), \tag{16}$$

where $\tau = \sum_{d=1}^{D} (a_d - p_d)^2 - (a_d - n_d)^2$, and D is the dimension of the embedding. Since τ will approximate a Gaussian distribution when D is large, we can obtain the leading two moments of τ as

$$\begin{aligned}\mu &= \mu_p^2 + \sigma_p^2 - \mu_n^2 - \sigma_n^2 - 2\mu_a\left(\mu_p - \mu_n\right),\\ \sigma^2 &= 2[\sigma_p^4 + 2\mu_p^2\sigma_p^2 + 2\left(\sigma_a^2 + \mu_a^2\right)\left(\sigma_p^2 + \mu_p^2\right) - 2\mu_a^2\mu_p^2\\ &\quad - 4\mu_a\mu_p\sigma_p^2] + 2[\sigma_n^4 + 2\mu_n^2\sigma_n^2 + 2\left(\sigma_a^2 + \mu_a^2\right)\left(\sigma_n^2 + \mu_n^2\right)\\ &\quad - 2\mu_a^2\mu_n^2 - 4\mu_a\mu_n\sigma_n^2] - 8\mu_p\mu_n\sigma_a^2.\end{aligned} \tag{17}$$

The loss is obtained as

$$\mathcal{L}_{BTL} = -\frac{1}{T}\sum_{t=1}^{T} \log P(\tau < -m), \tag{18}$$

where T is the number of triplets in a batch.

1.3 Triplet Loss

In the triplet loss (Schroff, Kalenichenko, and Philbin 2015) setting, we denote embeddings of query, positive, and negative images as a, p, and n, respectively.

$$\mathcal{L}_{TL} = \frac{1}{T}\sum_{t=1}^{T} \max\left(0, \|a-p\|^2 - \|a-n\|^2 + m\right), \tag{19}$$

where T is the number of triplets in a batch, m is a predefined margin.

When training Deterministic, MCD and BTL models, we follow (Warburg et al. 2021) and use online hardest mining to generate triplet tuples. Specifically, we first randomly select a query image X_i^q. Then, we select a positive image X_i^p

from the same class that has the largest embedding distance to the query image as X_i^q. Finally, we select a batch of negative images $\{X_{i,1}^n, X_{i,2}^n, .., X_{i,J}^n\}$ from a different class that has the smallest embedding distance to the query. We refresh the embedding cache at the beginning of every epoch.

1.4 Metrics

- **Recall@1**: The Recall@1 is the percentage of queries whose top-1 retrieved sample is the true nearest neighbor of the query.
- **Reliability diagram**: Given finite query samples, the trained uncertainty estimator estimates the uncertainty for each query sample. These uncertainties are then normalized to fit within the range of $[0, 1.0]$ (Warburg et al. 2021). We organize the query samples into 10 bins based on their uncertainties, grouping them into intervals such as $[0, 0.1), [0.1, 0.2), .., [0.9, 1.0]$. By calculating the recall@1 for the query samples in each bin, we can evaluate how well the uncertainty estimator is calibrated. Ideally, we expect the recall@1 to decrease proportionally with the uncertainty, as depicted by the dashed line in the reliability diagram.

2 Experimental Results

2.1 Cosine Similarity as Heuristic Uncertainty

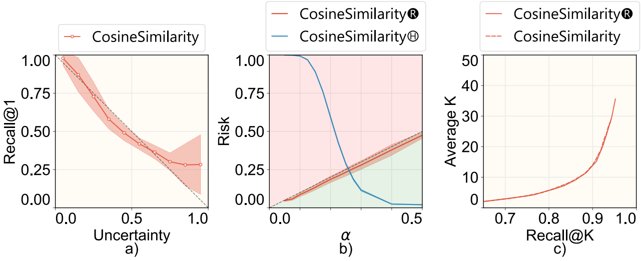

Figure 12: The reliability diagram, risk and average K of CosineSimilarity on the CUB-200 dataset

We consider using cosine similarity to measure the uncertainty between two embeddings, and we denote this comparing method as CosineSimilarity. Specifically, we take the cosine similarity between the query and its top-1 retrieved candidate as the heuristic uncertainty of the query. Similar to Fig. 5, Fig.6 and Fig.9 in the main paper, we show the reliability diagram, risk and the average K of CosineSimilarity on the CUB-200 test set in Fig. 12a), b) and c). Fig. 12a) shows that CosineSimilarity produces well-calibrated uncertainties, but Fig. 12b) demonstrates that theheuristic uncertainty-only methods, CosineSimilarity×Ⓗ, fail to control the risk. In comparison, CosineSimilarity×🅡 controls the risk effectively under different risk levels. Fig. 12c) shows that CosineSimilarity×🅡 does not rely on simply increasing K to control the risk.

2.2 Further Results on Varying Error Rates

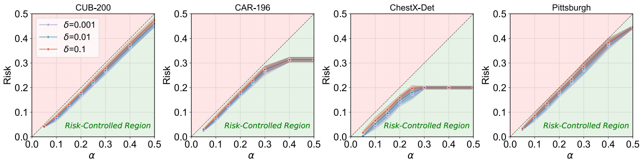

Figure 13: The risks on different test sets with different error rates by BTL🅡: smaller δ leads to more conservative retrievals (e.g., larger retrieval size). The colored shadows represent the standard deviation of the results of 10 trials.

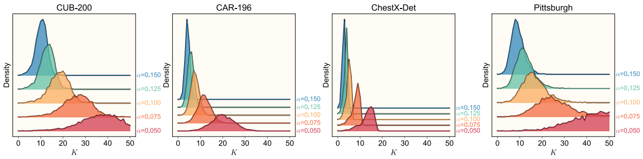

Figure 14: The retrieval size, K, on different test sets by BTL🅡: retrieval set size adapts to the risk level α.

Fig. 13 shows the empirical risk with different δ on the different test sets. With a smaller δ, RCIR would be more conservative (i.e., larger retrieval size) to ensure the risk is below the given α.

2.3 Further Results on Retrieval Size Distribution

Fig. 14 presents the distribution of retrieval size on different test sets. It is evident that the retrieval size varies with the risk level α: a smaller α results in a larger retrieval size, and vice versa. This helps end users save time on easy queries and focus on more difficult ones.

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Radenović, F.; Tolias, G.; and Chum, O. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668.

Schroff, F.; Kalenichenko, D.; and Philbin, w. c. d. t. r. . h. o. o. w. a. p. o. a. m. d. m., JaUsing (**??**). 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Warburg, F.; Jørgensen, M.; Civera, J.; and Hauberg, S. 2021. Bayesian triplet loss: Uncertainty quantification in image retrieval. In *Proceedings of the IEEE/CVF International conference on Computer Vision*, 12158–12168.