IMAGE RETRIVAL PROJECT

Andrea Mauro Marco Rossi Walter Scafa
University of Trento
andrea.mauro@studenti.unitn.it marco.rossi-6@studenti.unitn.it walter.scafa@studenti.unitn.it

1. Introduction

We address the task of image retrieval, which consists in identifying visually similar images from a reference dataset given a query image. Our goal is to design a retrieval system that learns discriminative image embeddings where semantically related instances are close in the latent space.

We focus on improving retrieval precision in a structured, multi-class setting. Our evaluation relies on three key metrics: Top-k Accuracy (presence of the correct label among the top-k results), Top-k Ratio (share of relevant items in top-k), and Top-1 Accuracy (exact match in top rank). These metrics inform our design and training decisions throughout the pipeline.

1.1. Overview of approaches

The initial training dataset lacked the diversity and scale needed for building a robust, generalizable model. To address this limitation, we extended it with the widely-used Food-101 dataset, chosen for its semantic richness, visual variability, and fine-grained categorization challenges. This addition proved essential for refining fine-tuning strategies and tuning hyperparameters in the food recognition domain.

Our retrieval framework integrates large-scale pretrained visual encoders with targeted fine-tuning and metric learning. This results in highly discriminative, semantically coherent embeddings that significantly improve retrieval performance across visually diverse food categories.

1.2. Summary of results

While large-scale pretrained vision models like CLIP offer strong general-purpose embeddings, their performance in structured, multi-class image retrieval tasks often plateaus without further supervision. Particularly in fine-grained domains such as food recognition, the gap between pretraining objectives and retrieval-specific goals can result in suboptimal clustering of semantically similar images.

In our work, we adopt a supervised strategy that incrementally optimizes the retrieval pipeline through a sequence of targeted modifications—each aimed at enhancing the alignment between visual similarity and semantic structure in the embedding space. Starting from a minimal CLIP-based baseline, we explore variations in loss functions, data augmentation, backbone architecture, and fine-tuning strategies.

Our systematic approach progressed through three key architectural configurations, each representing a significant leap in performance. The progression from our minimal baseline (Model 1) through multitask training with partial fine-tuning (Model 2) to our fully optimized architecture (Model 3) demonstrates relative improvements exceeding +14% in Top-1 accuracy and +13% in mAP 10 over the initial setup.

We frame our contribution not as a single novel technique but as a systematic exploration and integration of multiple components, providing a practical guide for enhancing visual-only retrieval through supervised adaptation. **In the following sections, we detail these three progressive configurations that mark the most consequential steps in our optimization journey**, highlighting the context, key architectural changes, and design considerations that drove our performance gains from a basic Vision Transformer setup to a state-of-the-art retrieval system.

2. Related Work

Deep metric learning has emerged as a fundamental approach for learning discriminative image representations that preserve semantic similarity relationships. The core objective is to train neural networks to embed images into a feature space where visually or semantically similar images lie close together while dissimilar images are separated by large margins. This section reviews the evolution of metric learning approaches and positions our work within the current landscape.

2.1. Early Pair-based Methods

The foundation of deep metric learning was established through pair-based loss functions that directly optimize relationships between image pairs. The Contrastive Loss [1] was among the first to formalize this concept, encouraging similar pairs to have small distances while pushing dissimilar pairs apart. Building upon this, the Triplet Loss gained prominence through its application in face recognition by

Schroff et al. [2].

The Triplet Loss operates on triplets of images: an anchor x, a positive x^+ (same class), and a negative x^- (different class). It encourages the anchor to be closer to the positive than to the negative by a margin γ:

$$L_{\text{triplet}} = \left[d(f(x), f(x^+)) - d(f(x), f(x^-)) + \gamma\right]_+ \quad (1)$$

where $[\cdot]_+$ denotes the positive part and $d(\cdot, \cdot)$ represents a distance metric.

A critical component of triplet training is hard negative mining, where negatives are selected to maximize the training signal. However, this approach suffers from significant limitations: slow convergence due to the vast number of possible triplets ($O(N^3)$ combinations) and the difficulty of selecting informative triplets [2]. These challenges motivated the development of more efficient alternatives.

2.2. Proxy-based Approaches

To address the computational inefficiency of pair-based methods, proxy-based metric learning was introduced. Instead of comparing image embeddings directly, these methods learn proxy vectors that represent entire classes, reducing the comparison complexity from $O(N^2)$ or $O(N^3)$ to $O(NC)$, where C is the number of classes.

Movshovitz-Attias et al. introduced Proxy-NCA [3], which assigns one learnable proxy per class and trains images to be close to their class proxy while being far from others. This approach significantly improved training efficiency and stability.

Building on this foundation, Kim et al. proposed Proxy-Anchor Loss [4], which treats proxies as anchors rather than targets. This key insight allows each proxy to simultaneously attract all positive samples and repel all negative samples within a batch, leading to richer gradient signals. The Proxy-Anchor loss is formulated as:

$$L_{\text{PA}} = \frac{1}{|P_+|} \sum_{p \in P_+} \left[\log \left(1 + \sum_{x \in X_p^+} e^{-\alpha(s(f(x),p) - \delta)} \right) + \log \left(1 + \sum_{x \in X_p^-} e^{\alpha(s(f(x),p) + \delta)} \right) \right] \quad (2)$$

where P_+ represents proxies for classes present in the batch, X_p^+ and X_p^- are positive and negative samples relative to proxy p, $s(\cdot, \cdot)$ denotes similarity (typically cosine), α is a scaling factor, and δ is the margin parameter.

The elegance of Proxy-Anchor lies in its automatic hard example mining through the log-sum-exp formulation, which naturally emphasizes difficult samples while down-weighting easy ones. Kim et al. demonstrated state-of-the-art results on multiple benchmarks, achieving 2-5% improvements in Recall@1 on CUB-200 and Cars196 datasets compared to previous methods [4].

2.3. Margin-based Classification Losses

A parallel development in metric learning emerged from face recognition, where classification losses with angular margins proved highly effective. ArcFace [5] exemplifies this approach by introducing an additive angular margin to the softmax loss. By normalizing both feature vectors and classifier weights to unit length, ArcFace operates on a hypersphere where the margin is defined in angular space.

The ArcFace loss computes logits as $s\cos(\theta_{y_i} + m)$ for the true class and $s\cos(\theta_j)$ for other classes, where θ_{y_i} is the angle between the feature and the true class weight, and m is the angular margin. This formulation naturally enforces intra-class compactness and inter-class separability on the hypersphere.

While formulated as classification objectives, these margin-based losses serve as effective metric learning methods by producing embedding spaces where same-class samples cluster tightly. ArcFace and related methods (CosFace, SphereFace) have shown excellent performance on both face recognition and general image retrieval tasks [5].

2.4. Recent Advances and Hybrid Approaches

Recent work has focused on more sophisticated pair weighting strategies. The Multi-Similarity Loss [6] addresses the limitation of considering only the hardest positives or negatives by incorporating multiple similarity relationships. Wang et al. achieved significant improvements, boosting Recall@1 from 60.6% to 65.7% on CUB-200 and from 80.9% to 88.0% on In-Shop Clothes datasets.

Other notable contributions include Lifted Structure Loss [7], N-Pair Loss [8], and various triplet variants with learnable margins. Many of these approaches can be unified under general pair weighting frameworks [6].

2.5. Performance Context on Food-101

To situate our work, we consider the Food-101 dataset, a challenging 101-class food recognition benchmark. Ghita and Ionescu [9] recently evaluated various losses on Food-101 for image retrieval, revealing striking performance differences. While standard cross-entropy classification achieved only 23-24% mAP@all with ResNet-18, their proposed Class Anchor Margin (CAM) loss reached approximately 75% mAP under identical conditions—a three-fold improvement that underscores the critical importance of metric learning objectives for fine-grained retrieval tasks.

2.6. Model Architecture and Training Formulation

2.6.1.

Network Architecture

Our image retrieval system consists of two main components: a convolutional backbone for feature extraction and a projection head for metric embedding. Let $f_\theta(x)$ denote the backbone network (e.g., ResNet) that maps an input image x to a feature vector $h = f_\theta(x) \in \mathbb{R}^{d_h}$, where d_h is the backbone's output dimension.

The projection head $g_\phi(\cdot)$ is implemented as a multilayer perceptron (MLP) that transforms the backbone features into the final embedding space. For a two-layer MLP with hidden dimension d_{hid}:

$$z_i = \frac{W_2 \sigma(W_1 h_i)}{\|W_2 \sigma(W_1 h_i)\|_2} \tag{3}$$

where $W_1 \in \mathbb{R}^{d_{\text{hid}} \times d_h}$ and $W_2 \in \mathbb{R}^{d \times d_{\text{hid}}}$ are learnable parameters, $\sigma(\cdot)$ is the ReLU activation, and the denominator enforces ℓ_2 normalization.

The ℓ_2 normalization is crucial for several reasons: (1) it constrains all embeddings to lie on the unit hypersphere, providing geometric stability; (2) it enables the use of cosine similarity, which is more robust than Euclidean distance for high-dimensional spaces; and (3) it is required by certain loss functions like Proxy-Anchor and ArcFace [5].

2.7. Loss Function Design

We employ a combination of two complementary metric learning losses: Proxy-Anchor Loss and Triplet Margin Loss. This hybrid approach leverages the strengths of both proxy-based and pair-based methods.

2.7.1. Proxy-Anchor Loss

Let $\{p_c \mid c = 1, \ldots, C\}$ be learnable class proxy vectors of dimension d for the C training classes. For each training batch $\mathcal{B}$, let $P_+ \subseteq \{p_c\}$ be the set of proxies corresponding to classes present in the batch. For a given proxy p, we define:

$$X_p^+ = \{z_i \in \mathcal{B} \mid y_i = \text{class}(p)\} \tag{4}$$

$$X_p^- = \{z_i \in \mathcal{B} \mid y_i \neq \text{class}(p)\} \tag{5}$$

The Proxy-Anchor loss is then:

$$L_{\text{PA}} = \frac{1}{|P_+|} \sum_{p \in P_+} \left[\log\left(1 + \sum_{z_i \in X_p^+} e^{-\alpha(s(z_i,p)-\delta)}\right) + \log\left(1 + \sum_{z_j \in X_p^-} e^{\alpha(s(z_j,p)+\delta)}\right) \right] \tag{6}$$

where $s(z_i, p)$ denotes cosine similarity between embedding z_i and proxy p, α is a scaling factor, and δ is the margin parameter.

Intuitively, the first term forces each proxy to attract its hardest positive sample (the one with lowest similarity), while the second term repels the hardest negative sample. The log-sum-exp formulation provides automatic hard example mining: difficult samples contribute exponentially more to the gradient, while easy samples are naturally down-weighted.

2.7.2. Triplet Margin Loss

To complement the proxy-based approach with direct pairwise optimization, we incorporate Triplet Margin Loss. For each anchor z_i in the batch, we select a positive z_{i^+} (same class) and a hard negative z_{i^-} (different class with maximum similarity to the anchor):

$$L_{\text{triplet}} = \frac{1}{|\mathcal{T}|} \sum_{(i,i^+,i^-) \in \mathcal{T}} \left[\|z_i - z_{i^+}\|_2^2 - \|z_i - z_{i^-}\|_2^2 + \gamma \right]_+ \tag{7}$$

where $\mathcal{T}$ is the set of valid triplets, $\gamma > 0$ is the margin parameter, and $[\cdot]_+ = \max(\cdot, 0)$.

Hard negative mining is implemented by selecting, for each anchor, the negative sample that maximizes the similarity $s(z_i, z_{i^-})$ among all different-class samples in the batch. This ensures that the triplet loss focuses on the most challenging and informative examples.

2.7.3. Combined Objective

The total training objective combines both losses:

$$L_{\text{total}} = L_{\text{PA}} + \lambda L_{\text{triplet}} \tag{8}$$

where λ is a weighting coefficient. In our experiments, we set $\lambda = 1$ to give equal importance to both components, though this hyperparameter can be tuned based on the specific dataset and application.

This combination allows our model to benefit from both approaches: L_{PA} provides efficient class-level structure learning with automatic hard mining, while L_{triplet} directly optimizes individual sample relationships with explicit margin enforcement. The proxy-based component ensures fast convergence and global class organization, while the triplet component fine-tunes local neighborhood relationships.

2.8. Training Considerations

Several implementation details are crucial for successful training:

Batch Composition: We ensure each batch contains multiple samples from several classes to provide rich positive and negative pairs for both loss components.

Proxy Initialization: Class proxies are initialized using Xavier initialization and updated through standard backpropagation alongside the main network parameters.

Hard Mining Strategy: For triplet loss, we implement online hard negative mining within each batch, selecting the most challenging negatives that violate the margin constraint.

Embedding Dimension: We set the final embedding dimension d to 128 or 256, balancing representational capacity with computational efficiency.

After the main training phase, we freeze the backbone parameters and perform an additional fine-tuning step on the projection head and classification layer only. This refinement allows the embedding head to better adapt to the retrieval objective, while preserving the strong visual priors captured by the pretrained CLIP backbone.

This architecture and training formulation creates an embedding space where images from the same class form tight clusters around their respective proxies, while maintaining clear margins between different classes—precisely the objective of effective metric learning for image retrieval.

3. Evaluation

3.1. Model Evolution

Our development process followed a gradual progression from a minimal configuration to a fully optimized Vision Transformer (ViT) architecture. We explored twelve distinct variants, each introducing targeted modifications to the network architecture or preprocessing pipeline to improve retrieval performance. Below, we present the three configurations that marked significant turning points relative to their immediate predecessors, highlighting context, key innovations, and their immediate impact.

Model 1 — Minimal Starting Point The initial setup employed a pretrained ViT-B/32 (QuickGELU) backbone with no layer freezing or data augmentations, trained in mixed precision for computational stability and efficiency. This configuration established a reasonable baseline, demonstrating that a solid backbone and proper mixed-precision training are sufficient starting points for further refinements.

Model 2 — Multitask Training and Partial Fine-Tuning To recover performance and introduce richer supervisory signals, we adopted a multitask training scheme on ViT-B/16, combining ProxyAnchorLoss with CrossEntropyLoss and unfreezing the last four transformer layers. The classification head was enhanced with BatchNorm and 15% Dropout, and a CosineAnnealingWarmRestarts scheduler managed the learning rate. This phase produced a noticeable performance gain over the minimal setup, confirming the effectiveness of partial fine-tuning.

Model 3 — Fully Optimized Architecture In the final variant, we integrated all lessons learned: a ViT-L/14 backbone (representing a larger model with smaller patch sizes for finer granularity) with a 2048-dimensional embedding; a triple-loss objective (ProxyAnchor + CrossEntropy + TripletMargin); a two-layer MLP head (Linear → BatchNorm → ReLU → Dropout → Linear); a CosineAnnealingWarmRestarts scheduler; and a structured augmentation pipeline (RandomResizedCrop, ColorJitter, RandomErasing). This holistic combination yielded the most substantial performance improvements, demonstrating that synergistic integration of model capacity, multi-loss training, and robust augmentations is essential for state-of-the-art supervised image retrieval.

3.2. Quantitative Results

Table 1 summarizes the retrieval performance of three representative models on our evaluation set. Metrics include Top-1, Top-5/Top-10 accuracy as well as mean Average Precision at corresponding cutoffs.

3.3. Ablation Study

In the interest of assessing the true robustness of our design, we deliberately did not fix a random seed across experiments. While this may introduce minor run-to-run variance, it more faithfully reflects real-world deployment conditions and confirms that, despite light architectural or hyperparameter tweaks, the retrieval performance remains consistently high.

In the following tables, we present a series of ablations where each component is varied in turn. The modest fluctuations in precision underscore that the final model's accuracy stems from a stable synergy of its parts, rather than from lucky initializations.

In this section, we systematically isolate the impact of individual components in the final model (Model 3). We present several ablation tables, each varying one key factor while keeping all others fixed.

Table 1 compares the retrieval performance of three models with increasing architectural complexity and training strategies. Model 3 (ViT-L/14 + HNM) significantly outperforms the others, reaching 0.882 Top-1 and 0.8972 mAP@10, highlighting the benefit of using a larger backbone with multitask loss and hard negative mining. Model 2 shows consistent improvement over the minimal baseline (Model 1), especially in mAP metrics. These results demonstrate that both model capacity and loss design contribute substantially to retrieval accuracy.

Table 1. Retrieval performance for representative models.

Model	Top-1	mAP@1	Top-5	mAP@5	Top-10	mAP@10
Model 1 (ViT-B/32 Minimal)	0.7404	0.7404	0.8880	0.7811	0.9240	0.7627
Model 2 (ViT-B/16 Multitask)	0.8004	0.8004	0.8656	0.8223	0.8760	0.8191
Model 3 (ViT-L/14 HNM 2048)	0.8820	0.8820	0.9344	0.9006	0.9420	0.8972

Table 2. **Ablation of loss function configurations.** This table presents the results of varying the loss function components. The combination of ProxyAnchor, Triplet Margin, and CrossEntropy losses (**”All three”**) yields the highest retrieval performance across all metrics (e.g., Top-1: 0.8880, mAP@10: 0.897205), confirming a strong synergistic effect. Notably:

- Removing the ProxyAnchor loss causes the largest performance drop (e.g., Top-1: 0.8308 , a decrease of approximately -5.1% relative to the ”All three” configuration), emphasizing its central role in structuring the embedding space.
- ProxyAnchor alone (Top-1: 0.8820) performs remarkably well, nearly on par with the full combination, suggesting it is the most effective standalone loss.
- Triplet loss on its own (Top-1: 0.8040) shows the weakest results, indicating it contributes primarily to local sample refinement rather than global structure, and benefits significantly from combination with other losses.

Loss Config	**Top-1**	**mAP@1**	**Top-5**	**mAP@5**	**Top-10**	**mAP@10**
All three	0.8820	0.8820	0.9344	0.9006	0.9420	0.8972
- Triplet	0.8780	0.8780	0.9260	0.8943	0.9348	0.8904
- CE	0.8756	0.8756	0.9316	0.8932	0.9420	0.8903
- Proxy	0.8308	0.8308	0.9212	0.8573	0.9372	0.8437
Only Proxy	0.8820	0.8820	0.9284	0.8982	0.9388	0.8963
Only CE	0.8584	0.8584	0.9132	0.8763	0.9232	0.8730
Only Triplet	0.8040	0.8040	0.9168	0.8311	0.9388	0.8118

Table 3. **Ablation of embedding dimension.** This table illustrates the impact of varying the embedding dimensionality on performance. Increasing the embedding dimensionality generally improves performance up to a certain threshold:

- **2048 dimensions** offer the best overall results (e.g., Top-1: 0.8864, mAP@10: 0.8981), balancing precision and computational efficiency.
- Performance degrades slightly at **3072 dimensions** (Top-1: 0.8816), hinting at possible overfitting due to increased model capacity without sufficient data, or diminishing returns where the additional complexity offers no further discriminative power.

Embed Dim	**Top-1**	**mAP@1**	**Top-5**	**mAP@5**	**Top-10**	**mAP@10**
512	0.8792	0.8792	0.9344	0.8973	0.9440	0.8923
1024	0.8740	0.8740	0.9228	0.8905	0.9332	0.8875
2048	0.8864	0.8864	0.9324	0.9027	0.9420	0.8981
3072	0.8816	0.8816	0.9272	0.8981	0.9356	0.8957

Table 4. **Ablation of the alpha parameter in ProxyAnchorLoss.** The scale parameter α critically influences the behavior of ProxyAnchor loss:

- An α of **64** achieves the best trade-off across metrics (e.g., Top-1: 0.8832, mAP@10: 0.8922), maximizing the balance between pulling anchors towards their proxies and pushing negatives away.
- Both smaller ($\alpha = 16$) and larger ($\alpha = 128$) values lead to suboptimal results (e.g., Top-1: 0.8756 and 0.8688 respectively), likely due to under- or over-emphasis on hard examples, which can destabilize training or hinder generalization.

Alpha	**Top-1**	**mAP@1**	**Top-5**	**mAP@5**	**Top-10**	**mAP@10**
16	0.8756	0.8756	0.9252	0.8892	0.9364	0.8855
32	0.8800	0.8800	0.9300	0.8953	0.9400	0.8918
64	0.8832	0.8832	0.9272	0.8966	0.9356	0.8922
128	0.8688	0.8688	0.9212	0.8843	0.9304	0.8817

Table 5. **Ablation of the triplet miner strategy.** The strategy for mining negative examples in triplets has a significant impact on performance:

- **Semi-hard mining** consistently performs best (e.g., Top-1: 0.8852), avoiding the instability of hard mining while still providing informative contrastive signals for effective learning.
- **Hard mining** results in lower performance (Top-1: 0.8752), likely due to the inclusion of noisy or ambiguous hard negatives that can lead to local minima or degrade the quality of the embedding space.

Miner Type	Top-1	mAP@1	Top-5	mAP@5	Top-10	mAP@10
Hard	0.8752	0.8752	0.9192	0.8871	0.9320	0.8845
Semi-hard	0.8852	0.8852	0.9296	0.8988	0.9388	0.8954
All	0.8840	0.8840	0.9280	0.8951	0.9376	0.8905

Table 6. **Ablation of the margin parameter in TripletMarginLoss.** The margin in Triplet Loss affects how much separation is enforced between anchor-positive and anchor-negative pairs:

- Margins of **0.1** and **0.5** perform best overall (both yielding a Top-1 of 0.8784), showing the model is robust to this hyperparameter within a reasonable range.
- Intermediate values (0.2-0.3) do not provide further benefits (e.g., Top-1: 0.8752 for 0.2, 0.8764 for 0.3), suggesting a relatively shallow U-shaped relationship between margin value and retrieval effectiveness, where very small or very large margins might be less optimal.

Margin	Top-1	mAP@1	Top-5	mAP@5	Top-10	mAP@10
0.1	0.8784	0.8784	0.9260	0.8953	0.9376	0.8929
0.2	0.8752	0.8752	0.9228	0.8909	0.9320	0.8869
0.3	0.8764	0.8764	0.9240	0.8932	0.9380	0.8916
0.5	0.8784	0.8784	0.9268	0.8927	0.9404	0.8892

Final Remarks

Across all configurations, the ProxyAnchor loss emerges as the key contributor to embedding quality, demonstrating its fundamental role in structuring the representation space. However, it is the balanced integration of multiple loss functions and judicious architectural and hyperparameter choices that collectively lead to state-of-the-art retrieval performance. The model also demonstrates strong robustness, with modest performance variation across a wide range of hyperparameter settings, confirming the value of a principled and compositional design approach for image retrieval tasks.

3.4. Retrieval Visualization

To complement the quantitative evaluation, we now present two qualitative retrieval examples. The figures below show a query image on the left alongside its ten most similar images retrieved from the gallery set on the right. This visualization demonstrates how the learned embedding space groups visually and semantically related items—for instance, different presentations of the same dish or images sharing similar color palettes. Of particular interest are instances where the model correctly matches plates with the same primary ingredient despite variations in composition, lighting, or viewpoint.

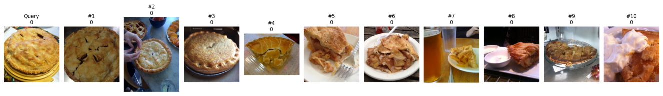

Figure 1. Top-10 retrieval results for the query *English pie*.

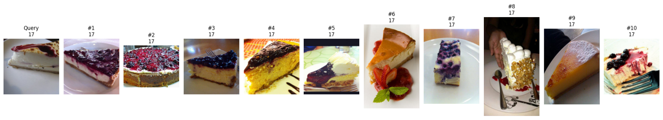

Figure 2. Top-10 retrieval results for the query *Cheesecake*.

To further analyze retrieval performance, we visualize the Recall@K and Precision@K curves across the top 10 retrieved items. These plots provide insight into how well the model maintains relevance and accuracy as the number of returned results increases. High and stable values across K indicate that the learned embedding space effectively ranks semantically similar images near the top of the list, ensuring both precision and coverage in retrieval tasks.

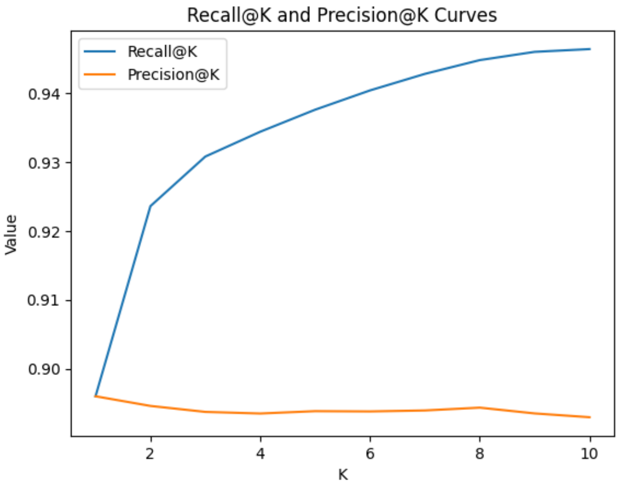

Figure 3. Recall@K and Precision@K for the retrieval model

These plots provide insight into how well the model maintains relevance and accuracy as the number of returned results increases. Recall@K shows a consistent upward trend, reaching approximately 0.947 at K=10, which indicates that nearly all relevant images are retrieved when the list is long enough. Meanwhile, Precision@K remains relatively stable around 0.895 across values of K, highlighting that the top results retain high semantic relevance. The combination of high recall and steady precision confirms that the learned embedding space effectively ranks similar items at the top, ensuring strong retrieval quality in terms of both coverage and accuracy.

4. Conclusions

In this work, we presented a comprehensive and methodical approach to supervised image retrieval in fine-grained, multi-class settings, with a specific focus on the challenging Food-101 dataset. Our contribution lies not in proposing a single novel component, but in the principled integration of established techniques—ranging from foundation model fine-tuning to hybrid loss function design and strategic architecture scaling.

Through a staged evolution from a minimal baseline (Model 1) to a fully optimized ViT-L/14 model (Model 3), we demonstrated substantial performance gains. Specifically, we achieved over 88% Top-1 accuracy and over 89% mAP@10 with our optimal configuration combining all three loss functions (Top-1: 0.8880, mAP@10: 0.897205). Ablation studies further confirmed that these improvements stem from the synergistic effect of combining Proxy-Anchor loss, Triplet Margin loss, and classification signals, along with carefully tuned architectural and hyperparameter choices.

Our findings validate the central role of Proxy-Anchor loss in structuring the embedding space, demonstrating its effectiveness even as a standalone loss. We also highlight the complementary benefits of triplet-based local refinement and classification-based global supervision. Furthermore, the robustness of our final model across configurations, showing modest performance variation despite various tweaks, underlines the strength of our compositional design strategy and its faithful reflection of real-world deployment conditions.

Overall, this work provides a solid blueprint for practitioners seeking to build high-performance, supervised image retrieval systems, especially in fine-grained domains where visual similarity aligns tightly with semantic class structure. Future work will explore generalization to multimodal retrieval and zero-shot adaptation, further leveraging the power of foundation models like CLIP and DINOv2 in broader retrieval contexts.

5. Work Distribution

The project tasks were distributed among team members as follows:

Andrea (A): Algorithmic choices (loss function, freezing layers, etc.), Dataset selection and analysis, Literature review and scientific papers reading (led the theoretical foundation of the project), Model testing (DINO and CLIP), Model training, Problem identification (image recognition), Theoretical part writing.

Marco (M): Algorithmic choices (loss function, freezing layers, etc.), Dataset selection and analysis, Literature review and scientific papers reading, Model testing (DINO and CLIP), Model training, Problem identification (image recognition), Theoretical part writing.

Walter (W): Dataset selection and analysis, Literature review and scientific papers reading, Model training, Problem identification (image recognition), Theoretical part writing.

All team members collaborated on almost every aspect of the project, with each member taking primary responsibility for their assigned tasks while contributing to the overall development and implementation.

References

[1] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *CVPR*, 2006, pp. 1735–1742.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *CVPR*, 2015, pp. 815–823.

[3] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, “No fuss distance metric learning using proxies,” in *ICCV*, 2017, pp. 360–368.

[4] S. Kim, D. Kim, M. Cho, and S. Kwak, “Proxy anchor loss for deep metric learning,” in *CVPR*, 2020, pp. 3238–3247.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” pp. 4690–4699, 2019.

[6] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, “Multi-similarity loss with general pair weighting for deep metric learning,” in *CVPR*, 2019, pp. 5022–5030.

[7] H. Oh, Y. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *CVPR*, 2016, pp. 4004–4012.

[8] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *NIPS*, 2016, pp. 1857–1865.

[9] A. Ghita and B. A. Ionescu, “Content-based image retrieval with class anchor margin loss,” *IEEE Access*, vol. 9, pp. 67 356–67 368, 2021.

The full implementation of our project, including code, training scripts, and experiments, is available at: `https://github.com/marcoRossi27/image-retrieval-project`