From CNNs to Transformers: Top-k Image Retrieval in a Competition Setting

Silvia Bortoluzzi Diego Conti Sara Lammouchi Tereza Sásková
University of Trento

Abstract

This paper presents our approach to a top-k image retrieval competition involving the matching of real celebrity photographs to their synthetic counterparts. The task required retrieving the top 10 most similar gallery images per query and was evaluated using a weighted top-k accuracy metric (600 for Top-1, 300 for Top-5, 100 for Top-10). Our approach combined traditional CNNs (ResNet, EfficientNet, GoogLeNet) with transformer-based models (DINOv2, CLIP), leveraging both frozen and fine-tuned configurations. During the competition, CLIP ViT-B/16 achieved the highest performance (510.24), surpassing all CNN baselines. Post-competition, we fine-tuned CLIP ViT-L/14 using hybrid loss and memory optimization, achieving a score of 791.82. These results demonstrate that large pretrained models, when paired with the right training strategies, are highly effective for semantic retrieval tasks involving cross-domain variation.

1. Introduction

The competition task focused on top-k image retrieval, where the goal was to match real celebrity photographs (queries) with their corresponding synthetic versions (gallery). We were provided with a labeled training set and an unlabeled test set divided into two folders: `query/` and `gallery/`. For each query image, the system was required to retrieve the top-10 images most visually similar from the gallery. The submissions were formatted in JSON and evaluated using a weighted top-k accuracy metric, awarding 600 points for a correct top-1 match, 300 points for top-5 and 100 points for top-10.

To address this task, we explored a range of pre-trained image models that had already learned to recognize patterns from large image collections. Some of these models were based on traditional neural network architectures, like ResNet, EfficientNet, and GoogLeNet, while others, such as DINO and CLIP, used more recent transformer-based designs. These models had been trained on large-scale datasets, allowing them to extract rich and general-purpose features useful for comparing image similarity. We tested both frozen models, used only to extract image features, and lightly fine-tuned versions, where only the final layer was trained on the competition data to improve retrieval performance.

During the 2-hour competition, our best performing model was CLIP ViT-B/16, which achieved a weighted top-k accuracy score of 510.240, significantly outperforming all other models we tested. Due to memory constraints, we were unable to deploy the larger CLIP ViT-L/14 model during the event. However, in post-competition experiments, we successfully fine-tuned ViT-L/14 and obtained our best overall result of 791.82. This demonstrates the advantage of scaling up transformer architectures in image retrieval tasks that require robust semantic understanding across visual domains.

All code, evaluation scripts, and model configurations used in this study are available at: https://github.com/tercasaskova311/ml-project-intro#.

2. Models Selected

In the lead-up to the competition, we had no information about the types of images in the test set, whether they would include faces, everyday objects, natural photos, or synthetic visuals. To prepare for this uncertainty, we selected a diverse set of model architectures capable of handling various visual domains.

Once the test data were made available, it became clear that the task involved matching real celebrity photographs with synthetic images representing the same individuals. It was important to understand that synthetic images often differ in texture, lighting, and style, even when depicting the same identity. This cross-domain setting required models that could capture high-level semantic representations, rather than relying solely on low-level visual features such as color or texture. Traditional CNNs tend to be more sensitive to such superficial details, while transformer-based models, such as CLIP, are better at identifying conceptual similarity.

Our strategy, combining traditional convolutional networks with more recent transformer-based models, gave us the flexibility to adapt quickly and select the best-performing architecture based on the test set's actual char-

acteristics.

To ensure comparability across models, we consistently used cosine similarity to rank image embeddings. Given that different encoders produce features with varying scales, this metric allowed us to focus on their orientation in feature space rather than raw magnitude. Such an approach is well-suited to semantic retrieval tasks, where recognizing conceptual similarity is more important than matching low-level visual details. For instance, a real photograph and a stylized rendering of the same subject may differ in texture or color, yet still convey the same meaning. Cosine similarity helped capture this alignment, making our comparisons more robust across domains and model types, without requiring additional normalization procedures [1].

2.1. ResNet

ResNet introduces skip connections to ease the training of deep networks by reformulating layers as learning residual functions with reference to layer inputs [2]. We adopted it as a classical baseline due to its simplicity, availability via `torchvision`, and strong performance on classification tasks.

We tested ResNet18, ResNet50, and ResNet152 by removing the classification head and extracting features from the global average pooling (GAP) layer. Fine-tuning experiments included unfreezing the last residual block (`layer4`) and training a lightweight MLP (Multi-Layer Perceptron) with ReLU and dropout, using cross-entropy loss to guide optimization.

Despite this, ResNet models consistently underperformed in both metrics we used. This finding is consistent with Li et al. [3], which demonstrates that ResNet features, optimized for classification, fail to capture the fine-grained similarities needed for instance-level retrieval. We thus focused our efforts on alternative architectures better suited to embedding learning, such as EfficientNet, DINOv2, and CLIP.

2.2. EfficientNet

EfficientNet optimizes both accuracy and efficiency through compound scaling, balancing width, depth, and resolution dimensions unlike traditional CNNs which typically scale one dimension at a time [4]. We used EfficientNet-B0 and B3 from the `timm` library with pre-trained ImageNet weights, comparing two pooling strategies: Global Average Pooling (GAP) and Generalized Mean Pooling (GeM), a trainable layer that interpolates between average and max pooling [5].

We implemented GeM as a custom PyTorch module in `EfficientNetWithGeM`, which extracts features via `forward_features()`, applies GeM pooling, and returns embeddings. An optional classification head enabled fine-tuning with cross-entropy loss while maintaining consistent embedding extraction through `get_embedding()`.

Both frozen and fine-tuned configurations were tested using the Adam optimizer. However, fine-tuning did not consistently improve retrieval—frozen models with GeM often outperformed fine-tuned counterparts, suggesting EfficientNet's pretrained features are well-suited to cross-domain retrieval. Overall, EfficientNet proved to be a strong CNN baseline, highlighting the impact of pooling strategies on retrieval quality.

2.3. GoogLeNet

GoogLeNet uses Inception modules and each one applies multiple convolutional filters in parallel, allowing the model to capture information at different spatial scales and aggregate them efficiently. [6]. We selected GoogLeNet as a lightweight CNN baseline to evaluate whether older architectures could compete with modern approaches when enhanced with updated training strategies. We used the pretrained PyTorch implementation with ImageNet weights, removing auxiliary classifiers (`aux1` and `aux2`) and replacing the global pooling layer with either GAP or GeM pooling. The classification head was replaced with `Identity()`, allowing direct embedding extraction from convolutional features. These embeddings were extracted using a custom `extract_embeddings()` function and compared via cosine similarity for top-k retrieval. We tested fine-tuning strategies: classifier-only updates versus full network training, using Adam optimizer with cross-entropy loss. GoogLeNet proved fast and stable but showed limited performance improvements from fine-tuning, with GAP outperforming GeM. Since GoogLeNet was originally optimized for classification, its intermediate representations lack the structured embedding space typical of modern retrieval-oriented architectures. Nonetheless, it provided a valuable baseline for comparison.

2.4. DINOv2

DINOv2 is a self-supervised vision transformer developed by Meta AI [7] that learns by grouping similar images based on internal patterns [8], unlike traditional models trained with labels. Its self-supervised training uses a teacher–student setup, encouraging the model to produce consistent embeddings across augmented views of the same image.

While larger variants of DINOv2 offer state-of-the-art performance in several benchmarks, we selected the base version to maintain a balance between performance and resource usage. In particular, we used the `facebook/dinov2-base` model from HuggingFace and tested two configurations: one using the frozen model (zero-shot), and one with a lightweight classification head trained on our labeled dataset.

In the frozen setup, images were processed through the transformer backbone, extracting the mean of final hidden states across spatial patches as embeddings. For fine-tuning, we added a trainable linear classification head on top of the frozen backbone, trained using cross-entropy loss with Adam optimizer. All DINOv2 weights remained frozen during training.

Embeddings were extracted using a custom `extract_features()` function, normalized to unit length, and compared via cosine similarity for top-k retrieval. The fine-tuned version generated embeddings by passing images through both the frozen backbone and trained head before normalization.

Results showed strong performance even without fine-tuning, with the classification head providing moderate but consistent improvements. This demonstrates that pretrained features were already semantically rich, with embeddings naturally organizing conceptually similar images close together, making DINOv2 a robust and generalizable reference model.

2.5. CLIP

CLIP (Contrastive Language–Image Pre-training) is a multimodal model developed by OpenAI that jointly trains image and text encoders to align visual and textual concepts in a shared embedding space using contrastive learning [9]. Its multimodal training gives it a strong inductive bias for semantic alignment, even in purely visual tasks.

We used the `ViT-L/14` variant, evaluating both frozen and fine-tuned modes for image retrieval. In the frozen setup, we used CLIP's `encode_image()` function to extract feature vectors from query and gallery images. These vectors were normalized to unit length and compared via cosine similarity for top-k retrieval.

To explore fine-tuning, we wrapped the visual encoder in a custom `ClipClassifier` module, adding a trainable linear head optimized using cross-entropy loss. Depending on configuration, we either froze the backbone or allowed partial/full weight updates. During inference, embeddings were extracted by passing images through both the encoder and trained head, followed by normalization.

We initially tested lighter variants such as `ViT-B/32` and `ViT-B/16`, which offered lower latency but also reduced accuracy. Based on empirical results, we selected `ViT-L/14` as our primary model due to its superior trade-off between precision and computational cost.

CLIP's pretrained features generalized well across our cross-domain setting, where real and synthetic face images needed to be matched semantically. Fine-tuning the projection head yielded consistent improvements, confirming CLIP's strength as a flexible and high-performing baseline for semantic retrieval.

3. Evaluation

This section provides an in-depth analysis of the models explored in our study, organized into three chronological phases: pre-competition, competition-day, and post-competition. To assess model performance across these phases, we employed two different datasets.

The first, used during pre-competition testing, is the publicly available *Animals* dataset sourced from Kaggle[1]. We used a custom script to split the dataset into `training`, `query` and `gallery` sets, with test images renamed to simplify later evaluation, using a standardized format `<class>_<number>.jpg`. Class names were derived from the original folder structure, where each subfolder corresponded to a specific animal species. Since the dataset contains only 9 images per class, we retrieved the top-9 most similar images for each query, instead of the top-10 as required during the competition.

The second dataset, used on the competition day, consisted of approximately 5000 training images and 3000 test images. Training images were organized by celebrity identity, with each identity subfolder containing both real (natural) and synthetic images. The test set was structured into a `query` set of real celebrity images and a `gallery` set containing synthetic images depicting the same celebrities in different visual styles. The primary objective of the competition was, for each query image, to retrieve the top 10 most similar gallery images.

Performance evaluation was based primarily on two metrics: Precision@K and top-k accuracy. Precision@K was our main metric during the pre-competition phase, measuring the proportion of correct images retrieved among the top-k results. We also recorded the runtime and the final training loss to analyze computational efficiency and convergence behavior.

It is important to note that Precision@K is a per-query metric, computed individually and then averaged across queries, whereas the official competition evaluation utilized a weighted top-k accuracy, specifically designed for the retrieval scenario. This metric assigned points based on the position of the correct images among retrieved results: 600 points for top-1 accuracy (correct identity at rank 1), 300 points for top-5 accuracy (at least one correct identity within the top-5 retrieved images), and 100 points for top-10 accuracy (at least one correct identity within the top-10 retrieved images).

3.1. Pre-Competition Evaluation

The pre-competition evaluation focused on systematically benchmarking multiple pre-trained models using the *Animals* dataset. Our aim was to identify suitable candidates

[1] https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals

for the subsequent competition and validate different training strategies and hyperparameters. Precision@K was our primary metric, complemented by runtime analysis and final training loss.

Among CNNs, EfficientNet-B3, with GAP and fine-tuning, achieved the highest performance with a Precision@K of 0.8513, with GoogLeNet, using GAP and fine-tuning, showing a promising Precision@K of 0.5575. In contrast, ResNet architectures consistently underperformed, with Precision@K rarely above 0.06, demonstrating a limited semantic retrieval capability, necessary for effective image retrieval tasks. All CNN models were trained using the standard Cross-Entropy loss function.

Among the models evaluated, the transformer-based CLIP ViT-B/32 consistently outperformed all CNN baselines, confirming the effectiveness of vision transformers for semantic image retrieval. Despite longer runtimes due to higher computational overhead, CLIP ViT-B/32 achieved a strong Precision@K of 0.6085 even without fine-tuning, significantly surpassed most CNN-based models. This validated our expectation that vision-language models like CLIP leverage richer semantic representations, strongly justifying our decision to prioritize CLIP and transformer-based models on competition day.

All results reported in this section are averaged over multiple runs per model configuration. We experimented with different batch sizes (16-64), training durations (10-20 epochs), and learning rates (1e-5, 5e-5, 1e-4), to assess model robustness and mitigate the effect of stochastic fluctuations. The choice of hyperparameters was guided by both empirical heuristics and practical constraints, considering GPU memory limitations and gradient estimation stability. For instance, smaller batch sizes allowed training on larger models, while higher learning rates were used for last-layer tuning, whereas lower values were adopted for full fine-tuning.

Figure 1 summarizes these results, comparing Precision@K of the best-performing configuration for each model family on the *Animals* dataset. CNNs, such as EfficientNet-B3 and GoogLeNet, showed strong performances, transformer-based models (CLIP, DINOv2) performed better than most convolutional architectures. EfficientNet with GAP and fine-tuned had the best performance on *Animals* dataset

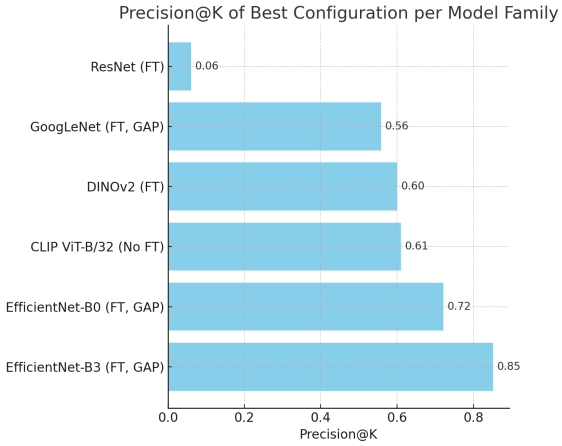

Figure 1. Precision@K of the best-performing configuration for each model family on the *Animals* dataset

3.2. Competition-Day Evaluation

On competition day, due to the limited time available, we strategically prioritized our testing sequence based on expected performance and computational efficiency. All models were initially evaluated without fine-tuning to quickly assess their baseline retrieval performance. The inference time for these preliminary tests varied according to model complexity, ranging from approximately 30 seconds for simpler CNNs, up to a couple of minutes for transformer-based models. Subsequently, models that demonstrated promising results were further explored with fine-tuning.

As anticipated, the transformer-based CLIP models delivered the strongest overall performance, significantly outperforming convolutional baselines. Among all configurations tested, CLIP ViT-B/16 with fine-tuning (batch size 32, learning rate 1e-4, 15 epoch, training only the last layer) achieved the highest weighted top-k accuracy of 510.24 points (see Table 1), confirming our expectation of its superior semantic retrieval capabilities due to large-scale pre-training.

As shown in Figure 2, the top-10 gallery images retrieved by CLIP ViT-B/16 include semantically consistent matches, even across stylistic variations.

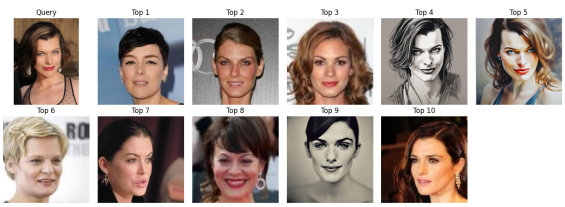

Figure 2. Qualitative example of retrieval results ranked by similarity, using CLIP ViT-B/16 on the competition dataset.

We also explored other variants of the CLIP family,

such as ViT-B/32 and ViT-L/14. The performance of the ViT-B/32 variant peaked at approximately 346 points. In contrast, the larger ViT-L/14 model encountered significant memory constraints on our hardware setup, resulting in repeated out-of-memory (OOM) errors, and forcing us to defer its evaluation and resolution strategies to the post-competition analysis described in Section 3.3.

In parallel, we briefly evaluated several convolutional architectures, primarily as comparative baselines. As expected, ResNet models consistently showed poor retrieval performance, with weighted top-k accuracy consistently below 10 points, confirming our previous findings and validating our decision not to focus on ResNet-based models.

GoogLeNet (CNN) and DINOv2 (transformer-based) achieved moderate performance. DINOv2 with fine-tuning (batch size 64, learning rate 5e-5, 7 epochs), obtained an accuracy of 68.38 points, marginally improving upon its baseline (67.56 points). GoogLeNet, tested with both GAP and GeM pooling, reached a maximum score of 35 points, reinforcing our expectation of limited effectiveness for complex semantic retrieval scenarios.

EfficientNet architectures provided somewhat unexpected results. EfficientNet-B3 (no fine-tuning, GAP) achieved a top score of only 38 points, while EfficientNet-B0 (no fine-tuning, GeM pooling) variant slightly outperformed it, reaching up to 44 points. Fine-tuning led to minimal improvements and in some cases slight performance degradation, indicating the potential limitations of fine-tuning smaller convolutional architectures on this specific task and dataset.

Ultimately, given the observed results, we focused our efforts on CLIP ViT-B/16 for the final submission, driven by its significantly superior accuracy and balanced computational requirements.

Table 1. Best competition-day results per model family. Fine-tuning was applied when epochs are specified; otherwise, results refer to non-finetuned configurations. Pooling is reported only where explicitly modified in the code.

Model	Batch	Epochs	LR	Pooling	Top-k
CLIP ViT-B/16	32	15	1e-4	-	510.24
CLIP ViT-B/32	32	7	5e-5	-	346.05
EfficientNet-B0	32	-	-	GeM	44
EfficientNet-B3	32	-	-	GAP	38
DINOv2	64	7	5e-5	-	68.38
GoogLeNet	32	-	-	GAP	35
ResNet18	32	10	-	-	5.42
ResNet34	32	10	-	-	5.42
ResNet50	32	10	5e-5	-	9.03
ResNet101	32	10	-	-	6.8
ResNet152	32	10	-	-	8

3.3. Post-Competition Evaluation

In the post-competition phase, we focused our evaluation on transformer-based models, with the aim of correcting implementation errors identified during the event, scaling to larger architectures, and enhancing model training through improved loss design. Building on CLIP's superior performance, we revisited our pipeline with a focus on correct fine-tuning, hybrid loss functions, and GPU memory optimizations.

We identified that the fine-tuning mechanism used during the competition did not work as intended. In particular, parameter updates were not correctly applied to the designated layers, limiting the model's ability to adapt to the task. To address this, we redesigned our training logic to explicitly support both full-model and final projection layer fine-tuning in a new pipeline, `clip_test.py` (available in our GitHub repository), which also incorporated memory management improvements, such as `torch.cuda.empty_cache()` and gradient accumulation, allowing us to finally run the larger ViT-L/14 model without OOM errors.

Another significant change was implementing a hybrid loss function that integrates standard cross-entropy with a contrastive objective applied to normalized embeddings. This facilitates the development of semantically meaningful representations while preserving the discriminative power necessary for classification, a critical aspect for cross-domain retrieval.

All CLIP variants demonstrated substantial performance gains under the updated fine-tuning approach. ViT-L/14 achieved a top-k accuracy of 791.82 with full-model fine-tuning and 747.70 with projection layer tuning. Interestingly, its baseline performance (603.85, pre-trained, no fine-tuning) under the new pipeline was lower than the competition-day baseline (637.66), suggesting that the revised training setup significantly enhances results when fine-tuning is applied, but does not inherently improve zero-shot retrieval performance.

Similar trends were also observed in smaller CLIP variants. ViT-B/16 reached 600.21 (projection layer tuning) and 599.86 (full fine-tuning), both outperforming its updated baseline (422.54) but still slightly below the original competition-day baseline (510.24). ViT-B/32 followed a comparable pattern, improving from 317.66 (pre-trained) to 520.89 (full fine-tuning).

These findings underscore the importance of correct fine-tuning implementation and careful loss design in extracting the full potential of large-scale pre-trained models. They also highlight that performance improvements in this phase stem primarily from training configuration and optimization, rather than inherent changes to model architecture or inference logic. Figures 3 and 4 visualize the comparative performance of CLIP models evaluated during and after the

competition. The first shows gains on the actual competition dataset in terms of weighted top-k accuracy, while the second confirms consistent improvements on the *Animals* dataset using Precision@K, thus validating the generalization of our updated training pipeline.

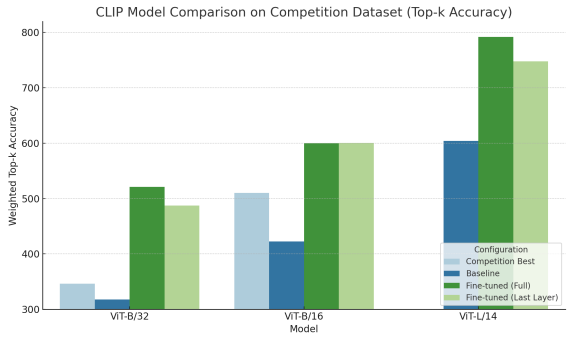

Figure 3. Comparison of CLIP variants on the competition dataset using the official weighted top-k accuracy. We compared the best-performing configurations from competition day (Competition Best) with baseline and fine-tuned results obtained using our updated codes. ViT-L/14 was not tested during the competition due to out-of-memory issues.

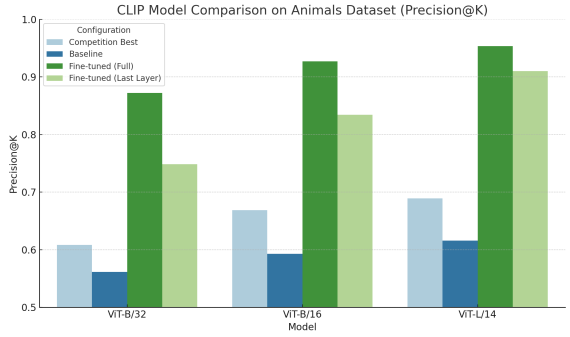

Figure 4. Precision@K results on the *Animals* dataset comparing the best-performing model tested during the pre-competition phase (ViT-B/32, Competition Best) with baseline and fine-tuned configurations obtained using the updated pipeline. Note that ViT-B/16 and ViT-L/14 were introduced post-competition for additional analysis. Fine-tuning strategies consistently improved retrieval performance over pre-competition baselines across all tested CLIP variants.

4. Final Discussion

The results of our experiments reveal a consistent theme: when semantic similarity is more important than pixel-level matching, transformer-based models, particularly the CLIP variants, systematically outperform convolutional baselines. This mirrors a broader shift in computer vision toward models that leverage large-scale multimodal pre-training, enabling stronger generalization across domains. In our case, CLIP's ability to retrieve semantically consistent results across stylistic variations proved critical in tasks such as celebrity identity retrieval.

An equally important insight is that architectural strength alone does not ensure performance. A powerful model can still underperform without a carefully engineered training pipeline. Only after addressing the flaws in the implementation of the fine-tuning process of ViT-L/14, we were able to fully express its potential, surpassing our competition day results by a substantial margin.

A particularly illustrative example is shown in Figure 2, where CLIP ViT-B/16 retrieves top-10 gallery images that are not only visually similar but semantically aligned with the query image, despite stylistic variation. This highlights what is perhaps CLIP's most defining feature: it does not simply match images at the pixel or texture level, but instead, pretrained on image–text pairs, CLIP learns to embed meaning itself. By operating in a joint vision-language embedding space, it is capable of abstract reasoning, recognizing identity and context in a manner closer to human perception. In contrast, traditional CNNs often rely on lower-level cues such as edges, textures, and colors, which limits their ability to generalize in cross-domain or semantically complex retrieval tasks.

Although we initially assumed that fine-tuning was being correctly applied during the competition, we later discovered that parameter updates were not propagated to the intended layers due to a flaw in the training logic. While CLIP provides strong zero-shot performance due to its large-scale pre-training, the absence of effective fine-tuning limits its ability to adapt to task-specific distinctions, such as discriminating between real and synthetic faces.

Initially, our training relied solely on the standard cross-entropy loss, which is effective in reinforcing class separation but does not preserve the relational structure of the embedding space. As discussed by Malik (2020) [10], cross-entropy minimization can be viewed as approximate bound optimization and relates to maximizing mutual information. However, in our experiments, using CE loss alone resulted in embeddings that were discriminative but poorly structured for generalization, confirming its limitations for retrieval tasks.

After the competition, we redesigned our training procedure by combining cross-entropy with a contrastive loss applied to *normalized embeddings*. This hybrid formulation aimed to balance discriminative classification with semantically meaningful representations. Inspired by recent work such as CLCE [11], we incorporated contrastive learning guided by class labels to complement the strengths of cross-entropy. While CE effectively enforces class separation, it does not structure the embedding space to reflect semantic similarity. In contrast, contrastive loss brings embeddings of the same class closer together while pushing apart those

from different classes, reinforcing intra-class cohesion and inter-class separation.

The CLCE framework extends this idea by leveraging label information to focus learning on informative negatives (visually similar but semantically different examples), without requiring large batch sizes. This improves training efficiency and makes contrastive learning more accessible, even in resource-constrained settings.

Motivated by this approach, we adopted a similar formulation that jointly optimizes for both classification accuracy and semantic consistency in the embedding space:

- Cross-entropy encouraged sharp decision boundaries.
- Contrastive loss preserved semantic continuity across samples.

This dual objective proved especially effective in our setting, where domain shifts and stylistic variations complicated direct matching. By enforcing both discriminative and structural constraints, the model successfully learned to associate different visual representations of the same identity. These empirical results align with findings from CLCE [11], which reports +3.52% gains in few-shot and +3.41% in transfer learning on BEiT-3, achieved without the need for batch sizes larger than 4096.

Equally important was addressing the resource bottleneck that had previously prevented us from training ViT-L/14. As mentioned, we repeatedly run into Out Of Memory errors during the competition. Cherti *et al.* demonstrated that CLIP models such as ViT-L/14 follow predictable *power-law scaling*: as model size, dataset scale, and compute increase, performance steadily improves across tasks like zero-shot retrieval and classification [12].

Our own results with ViT-L/14 closely reflect this trend. After resolving memory issues and fine-tuning the entire model (full fine-tuning), unfreezing all transformer blocks and optimizing with a small learning rate, we observed a significant performance jump, from 510.24 to 791.82 top-k accuracy. This reinforces Cherti *et al.*'s core insight [12]: while large-scale models hold strong potential, realizing that potential requires sufficient data and memory processing, along with a well-engineered training pipeline. Their study also emphasizes the importance of dataset composition. Different pretraining datasets (e.g., WIT-400M vs. LAION-2B) yield varying scaling curves. This parallels our observation that domain-specific fine-tuning on the synthetic–real celebrity dataset was essential to adapt the model to our retrieval task.

Finally, Cherti *et al.* [12] stressed the need for reproducibility and infrastructure reliability in scaling experiments. Our post-competition improvements, memory optimization, hybrid loss integration, and correct gradient flow, mirror this philosophy. Together, both works highlight that performance at scale is not just a function of architecture, but also of training design and computational strategy. Our post-competition improvements included explicit memory management (e.g.,`torch.cuda.empty_cache()`) and gradient accumulation, a technique that enables training with effectively larger batch sizes by accumulating gradients over multiple forward–backward passes before updating model weights. This allowed us to simulate large batches without exceeding GPU memory limits, improving stability and convergence. These changes enabled us to successfully train ViT-L/14, unlocking its full capacity. The resulting top-k accuracy of 791.82, compared to its baseline of 603.85 and the competition-day best of 510.24, demonstrates how critical infrastructure and training design are to performance.

In conclusion, our paper demonstrates that unlocking the potential of modern vision models is a function of both architectural selection and engineering. While Transformer-based architectures like CLIP proved superior to CNNs for semantic retrieval under domain shift, their advantage was fully realized only through a carefully designed training pipeline. The integration of a hybrid cross-entropy and contrastive loss was critical for building an embedding space that was both discriminative and semantically structured. Concurrently, infrastructure-level optimizations, namely gradient accumulation and explicit memory management, were essential to overcome resource constraints and successfully train the large-scale ViT-L/14 model. This unlocked a +281.6 improvement in top-k accuracy over our competition-day score, empirically confirming recent scaling laws: the performance of large models is fundamentally gated by the robustness of the pipeline used to train them.

Taken together, our findings highlights that achieving high performance at scale requires both selecting a model well suited to the task at hand, as in our case, where transformer-based architectures outperformed CNNs, and designing a training pipeline that enables the model to reach its full potential. Our results show that precision in optimization strategy, supervision, and resource management can make a decisive difference in semantic retrieval tasks.

Workload Table

Team Member	Contribution
Bortoluzzi Silvia	– Implemented EfficientNet (B0, B3) and added GAP and GeM pooling to the models. – Ran inference and top-k retrieval using cosine similarity. – Participated in competition testing. – Cleaned GitHub repo and finalized README. – Edited the final report as a whole.
Conti Diego	– Developed the GoogLeNet pipeline and performed ablation studies on pooling strategies and training configurations. – Evaluated Top-k retrieval accuracy under varying settings. – Wrote Section 3 (Evaluation). – Participated in competition testing and conducted post-competition experiments on CLIP. – Contributed to report editing.
Lammouchi Sara	– Implemented and tested the DINOv2 pipeline, including embedding extraction and optional fine-tuning. – Wrote Abstract, Introduction, and Sections 2 and 5 of the report. – Recorded all model outcomes during the competition to track progress and guide next steps. – Contributed to report editing.
Sásková Tereza	– Created GitHub repository. – Developed CLIP and ResNet (fine-tuned and evaluated). – Wrote Section 4 (Final Discussion). – Participated in model testing during competition. – Contributed to report editing.

Table 2. Summary of each team member's contributions.

References

[1] M. Kryszkiewicz, "The cosine similarity in terms of the euclidean distance," in *Encyclopedia of Business Analytics and Optimization* (J. Wang, ed.), pp. 2498–2508, IGI Global, 2014. 2

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 2

[3] L. Li, S. Zhang, and J. Wu, "Does ResNet learn good general purpose features?," in *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies (AIACT '17)*, pp. 1–6, ACM, 2017. 2

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6113, PMLR, 2019. 2

[5] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018. 2

[6] C. Szegedy and et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015. 2

[7] M. Oquab, T. Darcet, T. Moutakanni, and et al., "DINOv2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023. arXiv:2304.07193 [cs.CV]. 2

[8] Meta AI, "DINOv2: State-of-the-art computer vision models with self-supervised learning." https://ai.meta.com/blog/dino-v2-computer-vision-self-supervised-learning/, 2023. 2

[9] A. Radford, J. W. Kim, C. Hallacy, and et al., "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021. arXiv:2103.00020 [cs.CV]. 3

[10] B. Malik, T. Zhang, and A. Anandkumar, "Revisiting cross-entropy and mutual information in representation learning," in *Proceedings of the NeurIPS Workshop on Self-Supervised Learning*, 2020. 6

[11] Z. Long, G. Killick, L. Zhuang, G. Aragon-Camarasa, Z. Meng, and R. Mccreadie, "CLCE: An approach to refining cross-entropy and contrastive learning for optimized learning fusion," *arXiv preprint arXiv:2402.14551*, 2024. arXiv:2402.14551 [cs.CV]. 6, 7

[12] M. Cherti, R. Beaumont, R. Wightman, and et al., "Reproducible scaling laws for contrastive language–image learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, 2023. 7