Introduction to machine learning competition project

Tommaso Grotto
tommaso.grotto@studenti.unitn.it

Stefano Murtas
stefano.murtas@studenti.unitn.it

1. Introduction

1.1. Task Description

The goal of this task was to implement an image retrieval system that, given a query image, returns the top-k most similar images from a separate gallery set. This type of similarity search is widely used in applications such as recommendation systems, content-based image retrieval, and facial recognition. The challenge required designing an approach that performs well under strict time constraints, using only the training and test sets provided with no external labels for the test set.

1.2. Overview of Approaches

To solve this task, we explored a wide range of deep learning models, from classical CNN-based feature extractors to modern contrastive learning frameworks and fine-tuned multimodal models. Our pipeline generally followed this structure:

1. Feature Extraction: Use a pre-trained or fine-tuned model to extract embeddings from gallery and query images.
2. Normalization & Similarity: Normalize embeddings and compute cosine similarity between query and gallery features.
3. Top-k Retrieval: Return the most similar gallery images based on similarity scores.

The main models we considered included the following:

- ResNet (both pre-trained and fine-tuned)
- CLIP (zero-shot and fine-tuned variants)
- Triplet Network
- ArcFace
- SimCLR (self-supervised contrastive learning)
- Siamese Networks
- HuggingFace vision models
- Timm models
- FAISS for scalable similarity search

1.3. Summary of Results

Across our experiments, we found that CLIP offered the most consistent performance in top-k retrieval accuracy, with a good speed of processing. The other models performed poorly in the accuracy part when compared to CLIP, or they took too much time for the training aspect. Results were evaluated both quantitatively (top-k accuracy) and qualitatively (visual inspection of results).

2. Models Considered

In this section, we describe the models we explored for the image retrieval task, including their theoretical underpinnings and relevant literature references. The selection spans classical CNNs, self-supervised learning, metric learning techniques, and state-of-the-art multimodal models.

2.1. ResNet (Pretrained & Fine-Tuned)

Description: ResNet (Residual Networks) introduced by He et al. (2016) is a deep convolutional architecture that uses skip connections to alleviate the vanishing gradient problem in deep networks. We employed the ResNet50 variant, which contains 50 layers, and is widely used for transfer learning, and the ResNet18 as a lightweight CNN baseline. We used both:

- Pretrained: Used directly as a feature extractor by removing the final classification layer.
- Fine-Tuned: Retrained on the labeled training set to adapt feature representations to the specific domain.

Reference: He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

2.2. CLIP (Contrastive Language–Image Pretraining)

Description: CLIP (Radford et al., 2021) is a multimodal model trained to align image and text representations using contrastive learning. It uses a ViT (Vision Transformer) or ResNet backbone and is trained on 400M (image, text) pairs. We used both:

- Zero-shot CLIP (pretrained without modification)
- Fine-Tuned CLIP using image–label text pairs from our dataset.

Reference: Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PmLR.

2.3. Triplet Network

Description: Triplet networks are trained using a triplet loss that pulls an anchor image closer to a positive (same class) and farther from a negative (different class). This encourages the model to learn an embedding space where semantic similarity translates to spatial proximity.

Reference: Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

2.4. ArcFace

Description: ArcFace introduces an additive angular margin loss to improve inter-class separability and intra-class compactness. Instead of using standard softmax, it operates on the angle between embeddings and class weights.

Reference: Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4690-4699).

2.5. SimCLR (Simple Framework for Contrastive Learning)

Description: SimCLR is a self-supervised contrastive learning framework where models are trained to bring augmentations of the same image closer and push apart different images. No labels are required – just strong data augmentation and a contrastive loss.

Reference: Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PmLR.

2.6. Siamese Networks

Description: Siamese networks use two identical subnetworks to encode a pair of images and apply a contrastive loss to minimize the distance for similar pairs and maximize it for dissimilar ones. We trained both single and double fine-tuned Siamese networks.

Reference: Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2, No. 1, pp. 1-30).

2.7. FAISS (Facebook AI Similarity Search)

Description: FAISS is a library for efficient similarity search and clustering of dense vectors. It is used post-feature-extraction to retrieve top-k similar images quickly, especially when scaling to large gallery sets.

Reference: Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535-547.

2.8. Other Models (HuggingFace ViT & Timm)

We briefly experimented with vision transformer models from the Hugging Face transformers library and timm (PyTorch Image Models), using them primarily as pretrained feature extractors. While these models were not fine-tuned, they offered competitive baseline embeddings.

3. Evaluation

We evaluated our models using top-k accuracy, which measures the proportion of query images whose correct match appears among the top-k retrieved gallery images. Below are the results obtained during the competition day.

Table 1. Models' accuracy in the image retrieval task. ND: not deployed in the competition.

Model	Accuracy
CLIP (Zero-Shot)	345.57
Siamese Networks	46.52
HuggingFace ViT	43.37
ResNet18	30.72
ResNet50 (Pretrained)	30.58
FAISS	22.75
ResNet50 (Fine-Tuned)	ND
CLIP (Fine-Tuned)	ND
Triplet Network	ND
ArcFace	ND
SimCLR	ND
Timm	ND

From a quantitative perspective, the CLIP (Zero-Shot) model clearly stood out. It demonstrated exceptional ability to retrieve relevant matches, achieving the highest score without any task-specific training. This highlights the power of large-scale multimodal pretraining for zero-shot generalization.

In contrast, the other models performed notably worse. In order of accuracy, we have Siamese Networks, HuggingFace ViT, ResNet-based models, and FAISS. A key challenge we faced was that several of our promising models (e.g., ArcFace, Triplet Network, SimCLR) were not deployable within the competition time window, due to their computational demands or longer training requirements.

4. Discussion

In this section, we reflect on our overall experience in the competition, outlining the lessons learned regarding the practical deployment of image retrieval models under strict time and resource constraints. This image similarity task provided a valuable opportunity to experiment with a wide spectrum of models under realistic constraints, such as limited training time and compute availability. Through this experience, we gained key insights into model performance, efficiency, and robustness in the context of content-based image retrieval.

One of the clearest takeaways was the outstanding performance of the CLIP (Zero-Shot) model. Without any task-specific fine-tuning, CLIP significantly outperformed all other approaches. Its success illustrates the practical benefits of large-scale pre-training on diverse multimodal data and highlights how foundation models can generalize remarkably well to downstream retrieval tasks.

Other models, such as Siamese Networks and HuggingFace ViT, demonstrated lower retrieval performance compared to CLIP. While HuggingFace ViT benefited from lower training overhead, Siamese Networks required longer training times due to their pair-based structure. These models may still be considered in scenarios where task-specific fine-tuning is acceptable and computational budget allows.

Several of our most theoretically promising models, including ArcFace, Triplet Networks, and SimCLR, were not deployable within the competition's two-hour time frame. This reflects an important trade-off between model expressiveness and practical usability. Although these models often achieve excellent results in offline experiments, their training time and tuning requirements make them less suitable for time-critical environments unless prepared in advance.

The integration of FAISS for similarity search highlighted another important consideration: even with efficient indexing, the overall system performance is still limited by the quality of the underlying embeddings. This reinforced the importance of focusing on robust and generalizable feature extraction as a foundation for retrieval performance.

This challenge emphasized not only the accuracy of the model, but also the importance of efficiency and strategic model selection in applied machine learning.

Future improvements could include offline embedding computation, exploring hybrid retrieval pipelines, or incorporating lightweight self-supervised fine-tuning strategies. These adjustments may help balance performance and feasibility under real-time constraints.

5. Workload Table

In the following table, we report the contributions of each member. The 'Models' row reports the models each member created that were used on competition day, the 'Models ND' row reports the models each member created but did not use, the 'Report' row represents the contribution to the writing of the report, and the 'Repository' row indicates the contribution to the creation of the GitHub repository for this project.

Table 2. Models' accuracy in the image retrieval task. ND: not deployed in the competition.

Member	**Contributions**
Tommaso Grotto	Models: CLIP, HuggingFace ViT, ResNet50 Models ND: Triplet Network, ArcFace, SimCLR Report: 70% Repository: 30%
Stefano Murtas	Models: ResNet18, FAISS, Siamese Networks Models ND: Timm Report: 30% Repository: 70%

The code used for this project is available at:
https://github.com/smurtas/ML-STF.git