Artificial Intelligence

Task 2: Evaluation of Visual Encoders for Image Retrieval

Beatrice Valdes Bretti
Universidad de los Andes

May 12, 2025

Abstract

This report presents an evaluation of image retrieval performance using deep learning models on multiple datasets. The goal was to assess the effectiveness of three models: *ResNet34, DINOv2* and *CLIP* in retrieving relevant images from the *Simple1K, VOC-Pascal* and *Paris* datasets. Implemented in Python using PyTorch and related libraries, extracting features, computing cosine similarities, and evaluating performance with mean Average Precision (mAP) and Precision-Recall curves. Results show that *DINOv2* achieved the highest mAP of 0.931 on *Simple1K*, followed by *CLIP* (0.832) and *ResNet34* (0.812). Similar trends were observed on *VOC-Pascal* and *Paris*, highlighting the strength of self-supervised learning in capturing robust visual features.

1 Introduction

Image retrieval is a fundamental task in computer vision that involves identifying and retrieving images from a large dataset that are most similar to a given query image. This task is critical in numerous applications, including *content-based* image search, digital asset management and recommendation systems. The core challenge lies in representing images in a feature space where semantic similarity can be accurately measured, enabling the ranking of images from most to least similar to the query. The effectiveness of image retrieval systems depends on the quality of the visual encoder, which transforms raw images into compact, meaningful feature vectors and the similarity metric used to compare these vectors. This reports evaluates the performance of three state of the art visual encoders: *ResNet18, ResNet34, DINOv2* and *CLIP on* the image retrieval task across three diverse datasets: *Simple1K, VOC-Pascal,* and*Paris*. The evaluation employs cosine similarity as the metric and computes metrics such as *Mean Average Precision* (mAP) and *Precision Recall* curves, providing insights into the encoder's ability to capture semantic relationships in varied image collections. The problem of image retrieval is complex because of the dimensionality and variability of visual data. This requires visual encoders to extract robust features that generalize across differet datasets.

The primary programming environment is *Python* with *PyTorch* as the deep learning framework for loading and processing the visual encoders. The `torchvision` library provides pretrained *ResNet18* and *ResNet34* models, while the `clip` library from OpenAI facilitates the use of *CLIP* ViT-B/32 visual encoder. For *DINOv2* we rely on the implementation provided by Meta AI `dinov2` repository, specifically the `vit_small` variant. Similarity computations are done using `NumPy` for numerical operations and `scikit-learn` for computing evaluation metrics. `Matplotlib` is used to generate visualizations, and `pandas` helps in handling dataset metadata.

Simple1K is a small dataset with $1,326$ images across 50 classes, used for simple tasks. *VOC-Pascal* is a more complex dataset that contains $5,823$ images in its validation set distributed in 20 categories and its known for its variability in object appearance and backgrounds. The *Paris* dataset, with $1,274$ images across 12 classes is directed for image retrieval, focusing on landmark recognition. Each dataset includes a `list_of_images.txt` file mapping image filenames to class labels which is used to define relevance.

The visual encoders evaluated are:

- **ResNet18 and ResNet34**: These are convolutional neural networks from the ResNet family, introduced by He et al [1]. They utilize residual connections to mitigate vanish gradient issues. ResNet18 has 18 layers while ResNet34 has 34. offering a trade off between depth and computational efficiency.
- **DINOv2**: A vision transformer (ViT) developed by Meta AI, trained on 142 million images using a self supervised strategy. The `vit_small` variant divides images into patches and produces 384-dimensional feature vectors.
- **CLIP (Visual Encoder)**: Introduced by Radford et al. [3], CLIP is a bimodal model aligning image and text representations. The ViT-B/32 visual encoder generates 512-dimensional feature vectors.

The evaluation involves extracting features for all images in each dataset, computing cosine similarity between query and catalog features and ranking images using a leave one out strategy. We report *mAP* and *Precision-Recall* curves for each encoder-dataset pair, along with five examples of the best and worst retrieval results to analyze performance qualitatively. This setup allows us to assess the encoders effectiveness in image retrieval and understanding their applicability in real world scenarios.

2 Methodology

To address the task of evaluating image retrieval performance across multiple visual encoder models and datasets a *Python* program was developed. The goal was to extract image features using pre trained visual encoders, compute similarity scores, evaluate retrieval performance using mean *Average Precision* (mAP) and visualize results through *Precision Recal*l curves and top/bottom retrieval examples. The methodology required dataset preparation, model setup, feature extraction, similarity computation and evaluation and visualization.

2.1 Dataset Loading

First the dataset preparation consisted in understanding that each dataset had a consistent structure, with an `images` subfolder, organized by class like in *Simple1K*. All of them had a `list_of_images.txt` file with image filenames and class labels. The `load_dataset()` function was implemented to read the file, and so image paths were constructed. The function returned list of image paths and labels.

2.2 Model Setup

We evaluated three models:

- **ResNet34**: Loaded using `torchvision.models.resnet34`. The fully connected layer was replaced with an identity layer to extract 512-dimensional features.

- **DINOv2**: Loaded using `torch.hub.load('facebookresearch/dinov2','dinov2_vits14')`, a vision transformer producing 384-dimensional features.
- **CLIP**: Loaded using `clip.load("ViT-B/32")`, a vision-language model outputting 512-dimensional features.

Each model required specific preprocessing. ResNet34 and DINOv2 used a standard transformation pipeline: resizing to 224×224, converting to tensors and normalizing with ImageNet mean and standard deviation. CLIP used its own preprocessing pipeline provided by the `clip` library.

2.3 Feature Extraction

For each image, features were extracted using the following steps:

1. Load and preprocess the image using the model-specific preprocessing pipeline.
2. Pass the preprocessed image through the model to obtain features (ex: 512 dimensions for ResNet34, 384 for DINOv2).
3. Normalize the features by dividing by their L2 norm to ensure unit length.
4. Handle edge cases: if features contained `inf` or were too large, we replaced them with a small-valued vector (zeros + 1e-5).

This process was applied to all images in each dataset, resulting in a feature matrix for each model-dataset pair.

2.4 Similarity Computation and Evaluation

Cosine similarities were computed between query and dataset images using the normalized features. For each query image:

1. Computed the similarity matrix using a dot product of the feature matrix with itself.
2. Clipped similarities to the range [-1, 1] to avoid numerical issues.
3. Sorted images by similarity scores to rank retrievals.
4. Excluded the query image by setting its similarity to -1.

Performance was evaluated using mean *Average Precision*. For each query the *Average Precision* was computed using `sklearn.metrics.average_precision_score` comparing predicted scores to ground truth labels. The mAP was the mean AP across all queries. Generated *Precision-Recall* curves for visualization.

2.5 Visualization

To provide qualitative insights the top 5 and worst 5 retrieved images were visualized for the first query in each dataset. Used `matplotlib` and `skimage` to create a grid: the top row showed the query and top 5 retrievals and the bottom row showed the worst 5 retrievals, resized to 64×64 pixels for display.

3 Experimental Results and Discussion

Here are presented the results for each model on the three datasets, including mAP scores, Precision-Recall curves, and retrieval visualizations.

3.1 Quantitative Results

Table 1 summarizes the mAP scores for each model-dataset pair.

Model	Simple1K	VOC-Pascal	Paris
ResNet34	0.812	0.540	0.263
DINOv2	0.931	0.554	0.399
CLIP	0.832	0.553	0.322

Table 1: mAP scores for each model on Simple1K, VOC-Pascal, and Paris datasets.

3.2 Precision-Recall Curves

Figures 1, 2, and 3 show the Precision-Recall curves for each dataset, comparing the three models.

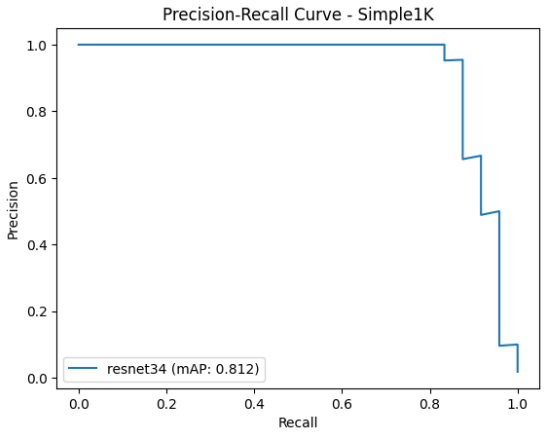

Figure 1: Precision-Recall curve for ResNet34 on Simple1K, mAP: 0.812.

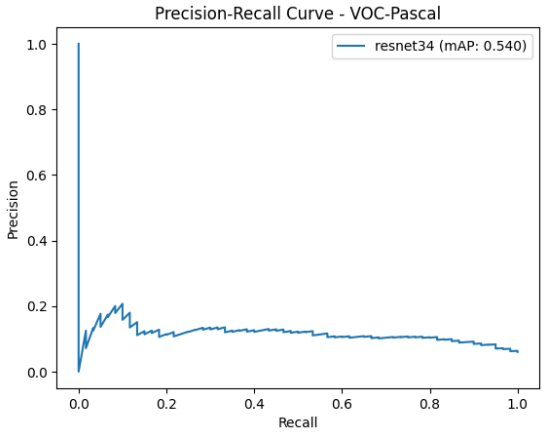

Figure 2: Precision-Recall curve for ResNet34 on VOC-Pascal, mAP: 0.540.

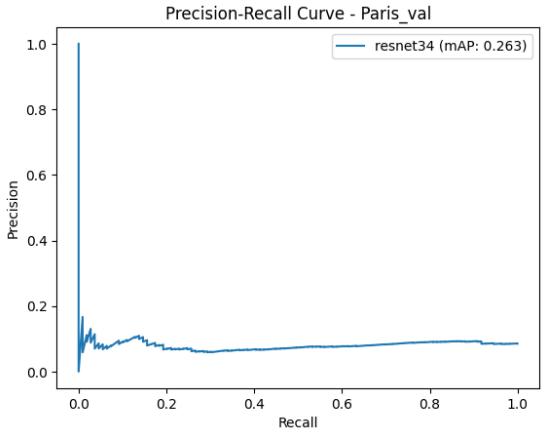

Figure 3: Precision-Recall curve for ResNet34 on Paris, mAP: 0.263.

3.3 Retrieval Visualizations

Figures 4, 5, and 6 display the top and worst retrievals for the first query in each dataset using ResNet34.

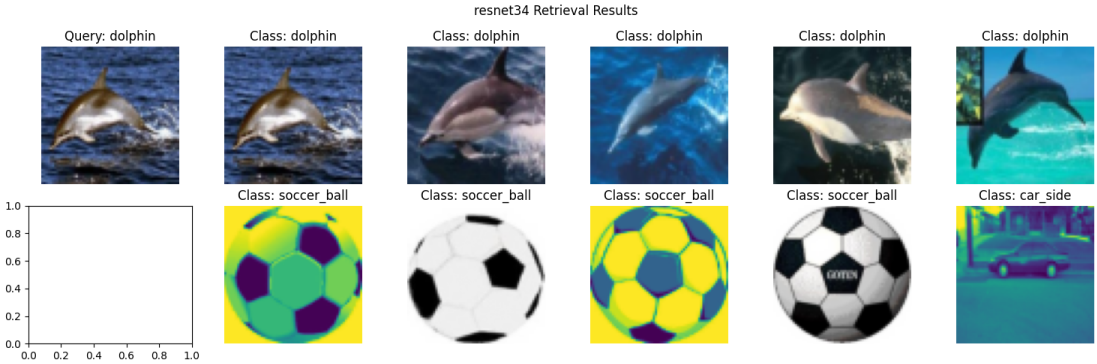

Figure 4: Top and worst retrievals for a dolphin query using ResNet34 on Simple1K.

Figure 5: Top and worst retrievals for a query using ResNet34 on VOC-Pascal.

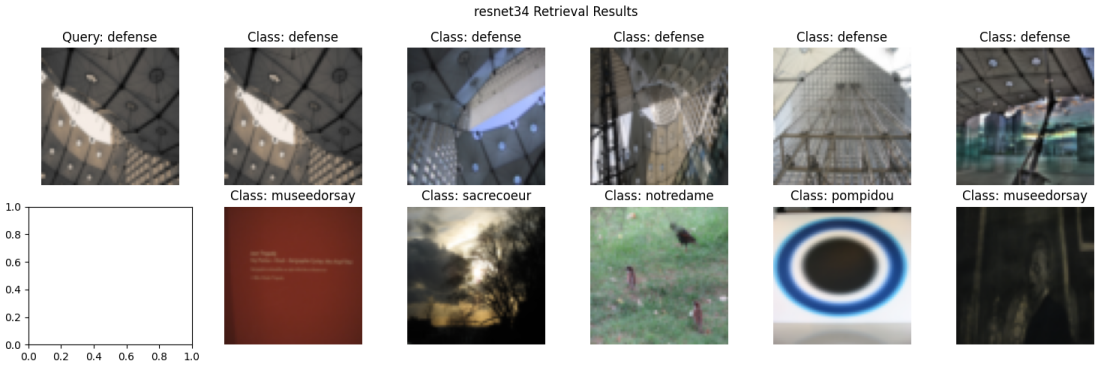

Figure 6: Top and worst retrievals for a query using ResNet34 on Paris.

3.4 Discussion

The results indicate that *DINOv2* consistently outperformed *ResNet34* and *CLIP* across all datasets, achieving the highest mAP scores of 0.931 (Simple1K), 0.554 (VOC-Pascal), and 0.399 (Paris). This superior performance is likely due to DINOv2s self supervised learning approach, which enables it to capture robust and generalized visual features, making it highly effective across diverse datasets, including object-centric (VOC-Pascal) and landmark-based (Paris) images. CLIP followed with mAP scores of 0.832 (Simple1K), 0.553 (VOC-Pascal), and 0.322 (Paris), performing well on Simple1K but showing reduced effectiveness on Paris, possibly because its multimodal training. ResNet34 had the lowest performance, with mAP scores of 0.812 (Simple1K), 0.540 (VOC-Pascal), and 0.263 (Paris), probably due to its reliance on supervised ImageNet pre-training, which may not generalize as well to datasets with specific visual characteristics.

The Precision-Recall curves reflect these trends, with DINOv2 showing the slowest precision drop as recall increases, indicating its ability to maintain high precision across a wider range of recall levels. Retrieval visualizations confirm that all models correctly retrieve same-class images in the top 5, while worst retrievals are from irrelevant classes, demonstrating effective class discrimination. However ResNet34 occasionally retrieved visually similar but incorrect classes, highlighting limitations in its feature representation.

4 Conclusions

This work evaluated image retrieval performance using *ResNet34, DINOv2* and *CLIP* on the *Simple1K, VOC-Pascal* and *Paris* datasets. DINOv2 achieved the best performance, with mAP scores of 0.931 (Simple1K), 0.554 (VOC-Pascal) and 0.399 (Paris) underscoring the advantage of self-supervised learning in capturing robust visual features across diverse visual tasks. CLIP and ResNet34 followed, with CLIP performing better on general datasets like Simple1K, while ResNet34 struggled, particularly on the Paris dataset. The pipeline successfully handled numerical issues and provided qualitative insights through visualizations. Future work could explore tuning models on specific datasets like Paris to improve performance or incorporating query augmentation to further enhance retrieval accuracy.

5 Appendix

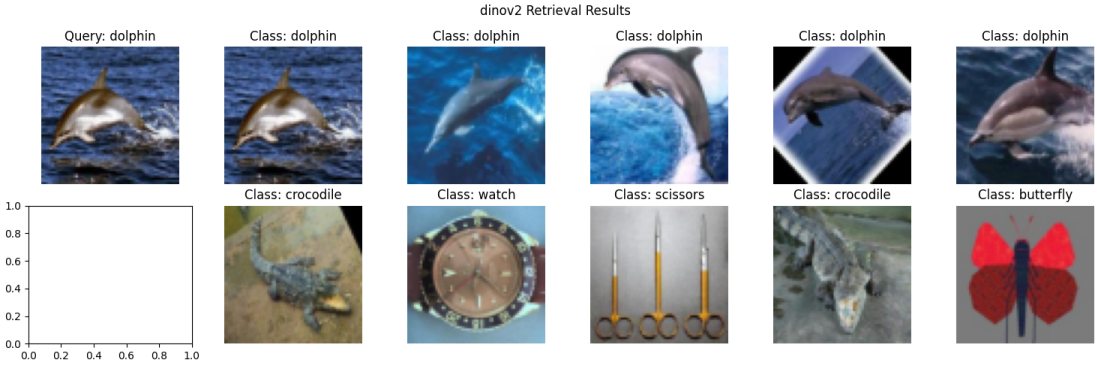

Figure 7: Top and worst retrievals for a dolphin query using DINOv2 on Simple1K.

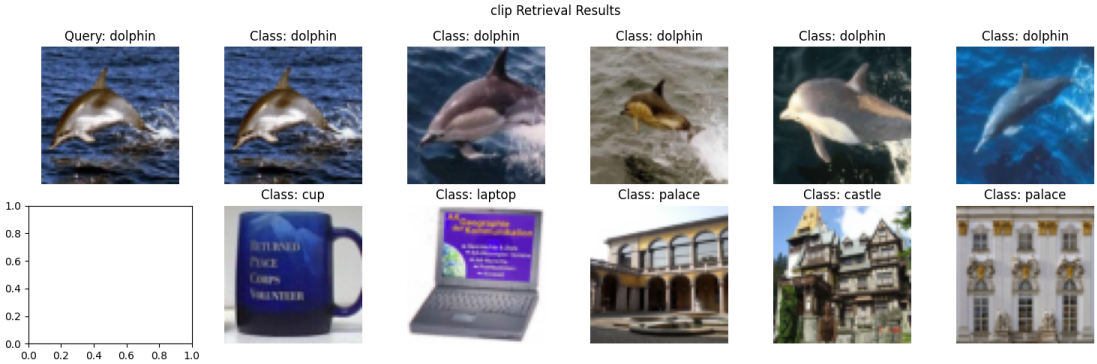

Figure 8: Top and worst retrievals for a dolphin query using CLIP on Simple1K.

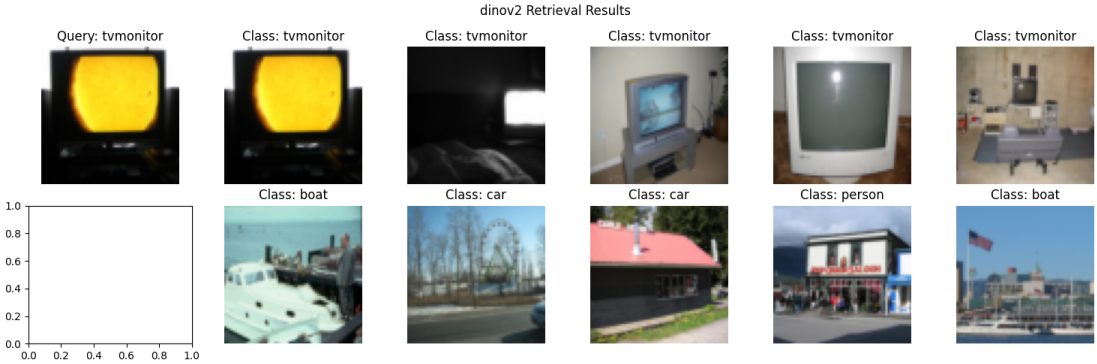

Figure 9: Top and worst retrievals for a query using DINOv2 on VOC-Pascal.

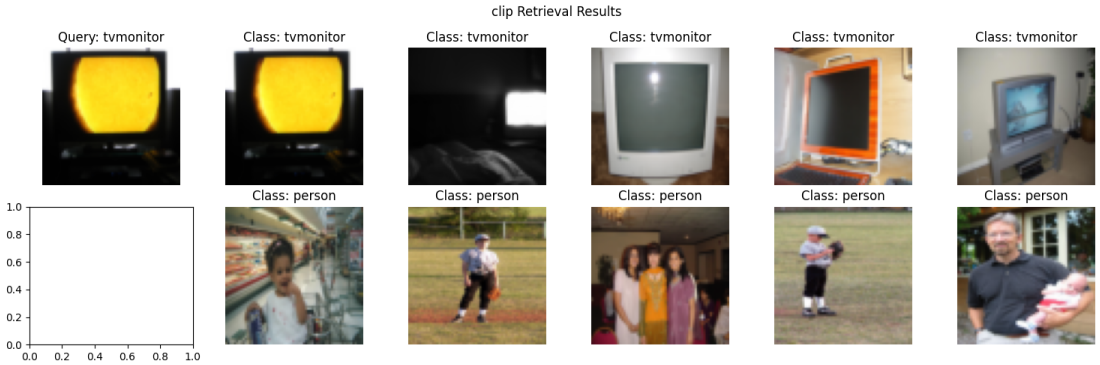

Figure 10: Top and worst retrievals for a query using CLIP on VOC-Pascal.

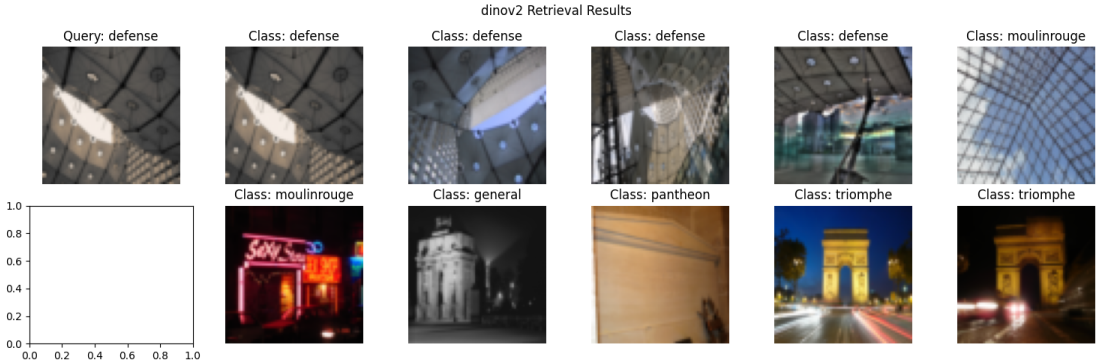

Figure 11: Top and worst retrievals for a query using DINOv2 on Paris.

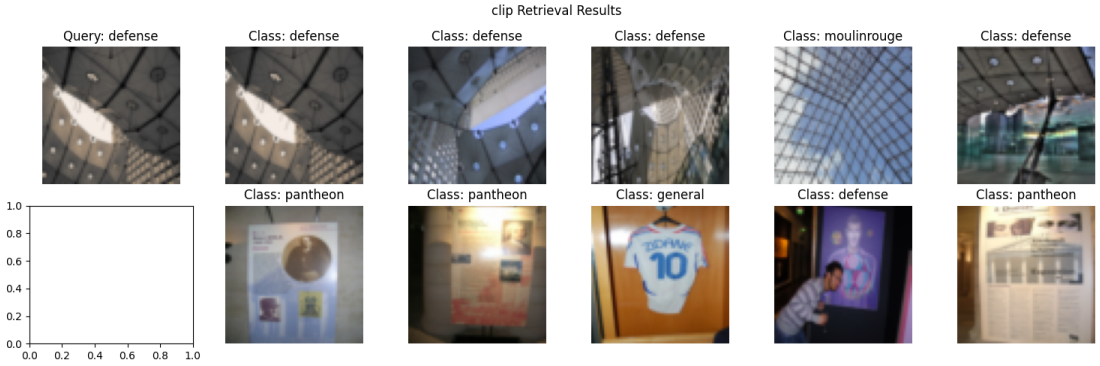

Figure 12: Top and worst retrievals for a query using CLIP on Paris.

6 Bibliography

References

[1] He, K., et al. (2015). Deep Residual Learning for Image Recognition.

[2] Oquab, M., et al. (2023). DINOv2: Learning Robust Visual Features without Supervision.

[3] Radford, A., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision.