VISION TRANSFORMER MEMORY AS A DIFFERENTIABLE SEARCH INDEX FOR IMAGE RETRIEVAL

Valerio Di Stefano
Sapienza University of Rome
distefano.1898728@studenti.uniroma1.it

ABSTRACT

In 2022, a paper titled "Transformer Memory as a Differentiable Search index" demonstrated that information retrieval can be accomplished with a single Transformer, in which all information about the document corpus is encoded in the parameters of the model, which acts as a Differentiable Search Index (DSI): the proposed text-to-text model maps string queries directly to relevant document IDs, answering queries directly by using only its parameters, which simplifies the retrieval process. We present results of a similar approach applied to image retrieval, where a Vision Transformer is used as a DSI for an indexed image database, and which, given an image as an input query, returns a list of relevant image IDs from the database.

1 Introduction

This paper explores a novel image retrieval (IR) framework that utilizes a differentiable function to generate a sorted list of image identifiers in response to a given image query: this approach is called "Differentiable Search Index" (DSI), and was originally proposed in the paper "Transformer Memory as a Differentiable Search Index" by researchers at Google Research [1]. DSI aims at both encompassing all document corpus information and at executing retrieval within a single Transformer model [2], instead of adopting the traditional "index-then-retrieve" pipeline used in most modern IR systems.

The paper presents an alternative implementation of the DSI approach adapted for the image retrieval task, where a Sequence to Sequence Vision Transformer model (ViT) [3], given an RGB image query as input, directly returns a list of IDs of images in an indexed database, ranked by relevance to the given image query.

In the proposed solution, we explore different approaches for a Vision Transformer model, varying both traditional machine learning models' training parameters (such as dropout rate, training epochs, learning rates or batch size) and parameters that are specific to the Transformer and ViT architectures (such as learned/fixed positional encodings, dimensions and configurations of "Multi-Head Attention "layers or fully connected layers, or image patch size), as well as varying datasets and task-specific paremeters too (indexing/retrieval data splits, datasets and datasets sizes, clustered/random image IDs and number of classes or relevant images).

The performances of the proposed models' variations are evaluated using three metrics, considering a traditional "index-then-retrieve" approach (using a "Bag of Visual Words") as a baseline, and computed on different versions of the Microsoft Common Objects in Context (MS COCO) [4] and ImageNet [5] datasets: "Indexing Accuracy" (IA), "Mean Average Precision" (MAP) and the "Recall at K". We ultimately compare the results of the models, as well as using a Bag of Visual Words model using a traditional index-then-retrieve approach as a baseline.

2 Data

For the training of the Vision Transformer model variations, as well as the initialization of the BoVW model (see Section 3), different variations and sizes of the MS COCO [4] and ImageNet [5] datasets were considered.

We also tested different ways of indexing the images in the final database, and different percentages of images to populate the indexed images database and a retrieval dataset for testing and fine-tuning the models (see Section 3.2.1).

2.1 Datasets and Variations

The MS COCO and ImageNet datasets were used to train and test the various ViT models variations, but only after simplifying, resizing and preprocessing their images and corresponding contextual information.
In particular, the images of both datasets were selected to only include squared images: only images with either a square aspect ratio or an aspect ratio inside a fixed tolerance range (set to [0.9, 1.1]) were considered.
All images were then cropped to a proper square aspect ratio and resized to 3 different sizes: 256x256, 160x160 and 64x64. All images also use 3 color channels (3 dimensional tensors) representing RGB color values, and were therefore represented as 3D RGB values arrays.

The datasets were then split into an "Indexing Dataset" and a "Retrieval Dataset" for the training of the ViT models.
For the "Indexing Dataset", a fixed number of images were chosen, and to each of them an ID was assigned, to finally constitute an indexed database (IDB) for the image retrieval task: these images were used to populate the dataset with a list of <image, ID> tuples from the IDB, representing images mapped to their image IDs.
For the "Retrieval Dataset", instead, a different number of images were chosen to constitute image queries, represented once again as <image, relevant_ID> tuples where the relevant ID is that of an image of the IDB (thus the "Indexing Dataset") that is considered relevant to the given image query.

These datasets were created by taking into account either different object instances found inside the image for the MS COCO dataset, or the class of the image for the ImageNet datasets. In particular, to define the relevant images of the "Retrieval Datasets" for the MS COCO dataset, all indexed images containing an instance of an object found also in a query image are considered relevant to that query image. For the ImageNet dataset, instead, we consider as relevant to an indexed image all query images that have the same class (or label) of the indexed image.

For indexing the images of the IDB, thus to compute the IDs of images found in the "Indexing Dataset", we considered both a dense random approach, where we simply shuffle the images in the DB and then assign them consecutive numeric IDs, and a dense clustered approach, where images containing the same object instances (for MS COCO images) or having the same class label (for ImageNet images) were clustered together and thus were assigned similar, consecutive numeric IDs.

2.2 Transformer Data

The "Indexing" and "Retrieval" datasets used for training the ViT model contain encoded images and encoded image IDs: for the encoding of images, we simply consider a 3 dimensional tensor containing the RGB color values of pixels of each image (later divided into patches of fixed sizes, see section 3.2), while for image IDs, we encode each digit of the transformer model as a token from 0 to 9, and then use special tokens (11 for "beginning of sequence", 12 for "end of sequence", and 13 for "padding") to complete the sequence representing the ID, then we also consider a vector representation of the digit token, compatible with the expected size of the Transformer model's inputs.

3 Models

To evaluate the final ViT models with respect to each other and to a baseline, an additional "Bag of Visual Words" model was also considered, as a a "no machine learning" approach using the traditional "index-then-retrieve" image retrieval pipeline.
Additionally, various Vision Transformer model architectures and configurations were considered, varying multiple Transformer-specific and task-specific parameters.

3.1 Bag of Visual Words

The model consists of a simple Bag of Visual Words vectorizer (BoVW), which uses no Machine Learning for its training, but rather simply computes the BoVW histograms, represented as vectors, for each image in the IDB.

For the indexing of images in the IDB, the BoVW model is initialized by first extracting image features and descriptors from all the (gray-scale) indexed images using a SIFT detector (Scale Invariant Feature Transform), then performing a k-means clustering (considering a fixed number of clusters K=200) to obtain cluster centroids that will represent the "visual words" to use in our "dictionary", and ultimately computing the BoVW representation for each image in the IDB, mapping the extracted SIFT features of each image to the nearest centroid and then counting the number of features belonging to each cluster to obtain the final BoVW histogram values (represented as a vector).

At inference time, the BoVW model computes, for a given query image, its BoVW representation, by once again extracting SIFT features and mapping them to the already initialized cluster centroids (hence the words in our dictionary), and compares the obtained BoVW representation with the corresponding BoVW representation of all the images in our IDB, to then return the top K results based on cosine similarity.

The approach adopted by the BoVW model is therefore that of an "index-then-retrieve" pipeline, in which we first index the images in our database, and then, at inference time, we retrieve all relevant results by iterating through the whole indexed database to compute the similarity scores (cosine similarity) between the given query image and each other single image in the database.

3.2 Vision Transformer

The implemented Vision Transformer model architecture consists of several "Multi-Head Attention" layers in sequence for both the encoder and decoder modules, with varying number of heads and varying embedding sizes for both the attention modules and the fully connected layers.
The Vision Transformer module takes as input an image as a 3D RGB values tensor, which is divided into patches of a fixed size (multiple approaches were tested, considering 8x8, 16x16 and 32x32 patch sizes see Section 3.2.1) that will represent the sequence tokens given as input to the ViT, and to which additional tokens are appended, which represent the digits of an image ID along with the special tokens for the "beginning of sequence" (BOS), "end of sequence" (EOS) and "padding" tokens, once again represented as vectors.
Furthermore, the transformer model has 12 output heads representing the "logits" (corresponding to probabilities) of the 10 possible tokens of the image IDs (digits from 0 to 9) and the special "end of sequence" and "padding" tokens.

During training (performed with variable training parameters, most notably early stopping training epochs and dropout), for both the "indexing" and "retrieval" phase (using the corresponding datasets defined in section 2), the transformer is given as input vector corresponding to patches of an image, to which a vector corresponding to the BOS token is appended along with the remaining of the image ID tokens, masked using the traditional "masked attention" approach: the Transformer model thus learns to output a sequence of tokens of the image ID ("teacher forcing" approach) in response to given image queries (DSI image retrieval).
For the training of the "indexing" task, in particular, no split of the training data was used, but instead the entirety of the training data, hence all of the <image, ID> pairs (described in section 2), was used for the training set: this ensured that the model could learn to map images to image IDs for all of the images in our IDB.
For the training of the "retrieval" task, the usual train-validation-test dataset split was used (0.8, 0.1, 0.1), with then the "test" data used to compute the IA@K, MAP@K and Recall@K evaluation metrics.

During inference, the Transformer model takes as input an image (divided in patches and vectorized, as for the indexing task), and the BOS token vector, and predicts the next image ID token using an "autoregressive" approach, outputting logits (probabilities) for the next token to append to the image ID tokens sequence.

An important note for the training of the Transformer models is that, unlike the usual machine learning or deep learning models' training settings, during the "indexing" task, we try to "overfit" our model to the training data: in this setting, in fact, we want to memorize the training data (the indexed images database) in the weights of the Transformer model's network, to then be able to perform inference (generate image IDs from image queries) without accessing the indexed database again.

3.2.1 Transformers Variations

The Vision Transformer model was tested on different versions and sizes of the datasets, and considering different configurations and training parameters.
Because the number of possible combinations of configurations and parameters was high, and the amount of computing resources available was limited, a "training/performance trade-off" was considered while evaluating different model configurations and approaches: we aimed at keeping the needed training time and resources at a minimum, while trying to obtain good resources on the 3 considered evaluation metrics ("indexing accuracy", "mean average precision" and "recall").

For the different versions and size of the MS COCO and ImageNet datasets used, the best "training/performance" trade-off was obtained for the ImageNet dataset using 160x160 images: in this case, the size of images was not high enough to considerably slow down training, while the amount of retained detail for each image was still high enough to allow the ViT model to compute appropriate feature representations for each image.
The use of the ImageNet dataset also forced a mutually exclusive grouping for relevant images in our indexed database: every ImageNet image has one and only one associated class/label. This means that for the retrieval phase training, each query image also has one and only one associated class/label, and is thus relevant to only a small amount of images of the indexing database (belonging to the same class), with no cross-class image relevance possible.
In a real-life image retrieval task scenario, a database would usually contain images that have multiple features (instead

of a single associated class/label), and therefore the MS COCO dataset would be a better fit for out ViT model: for this dataset, however, in order to obtain good performances, the ViT model would have to be trained for a far larger number of iterations and would need a far more complex architecture, which would in turn inflate training time and possibly hinder the possibility of training the model altogether (because of limits on the amount of memory we can allocate during training).

We also tested both a "dense random IDs" approach and a "dense clustered IDs" approach (see Section 2). The clustered approach led to higher performances than the random approach on both indexing and training (since the Transformer model learned to map similar image IDs to similar images).
In a real-life image retrieval task scenario, however, using a "dense clustered IDs" approach means that, at every update of the database, the IDs of all the images in the entire dataset would have to be re-computed, and the whole ViT DSI would have to be re-trained on the new IDs. A "dense random IDs" approach, however, would simply require new images to be assigned the next available ID in the database, and image deletions would simply require the ID to be deleted from the database: we can then simply fine-tune the Transformer model on the added images' IDs. Another approach would be to use "sparse semantic IDs" with a similar approach explored in the original DSI paper by Tay et al. [1] (not tested nor reported here, for brevity) this approach, however, leads to longer image IDs sequences to be generated by the Transformer DSI with its sequence-to-sequence approach (because we have to account for new images possibly being added to the database, thus leave empty spaces inside the dataset), and therefore to a higher error probability.
The considered ViT DSI specialized on clustered IDs would therefore be optimal only in a setting where either database updates would be rare or absent, or a setting where re-computing image IDs and re-training the Transformer model entirely would be feasible at a rate equal to the rate at which database updates occur.

Considering the aforementioned ImageNet 160x160 dataset variation, out of the tested 8x8, 16x16 and 32x32 possible patch size, the 16x16 solution (with thus 10x10=100 patches per image) was found to lead to better model performances, along with a total of 10 classes in the database.
In this setting, for the Vision Transformer model, both a "learned" and "non-learned" approach for the positional encodings of image patches were tested: the non-learned approach (using the cosine positional encoding solution presented in the original Transformers architecture paper [2]) showed better performances overall for the 3 main measured metrics.

The size of the indexing and retrieval dataset was also a crucial parameter to tune for the "training/performance trade-off", as well as for the trade-off between indexing accuracy and retrieval accuracy: because we first learn to index the images in our database by means of the "indexing" training phase, a large size of the retrieval dataset for the "retrieval" training phase would lead to a better generalization, and thus a higher retrieval accuracy, but would also skew the learned weights for the indexing phase of the transformer away from the optimal learned weights that could encompass all the information of our indexed database, thus leading to a lower indexing accuracy.
We tested several dataset sizes (512, 1024 and 2048 images), and for each of them, the best evaluation results were obtained by considering a 25% / 75% split for the "Indexing Dataset" and the "Retrieval Dataset" respectively (see Table 1).

Moving onto training parameters, varying the number of max training epochs for the early stopping approach for both the indexing and training phases showed radically different results for the evaluated performance metrics.
As for the number of training epochs of the "indexing" phase, we chose, for each tested ViT architecture and dataset, a value that would lead to a proper overfitting of the training data, thus a training accuracy of 0.98 or above, with usually 700 to 1000 indexing training epochs needed.
For the "retrieval" phase, instead, we considered different early stopping max epochs: as the number of max retrieval epochs grew (usually from 0 to 50), the retrieval accuracy also grew, while the indexing accuracy got close to 0. A "zero shot learning" approach was also tested, where no fine-tuning for the retrieval training phase was performed: this obviously led to high indexing accuracy results but also low retrieval results (see Table 1).

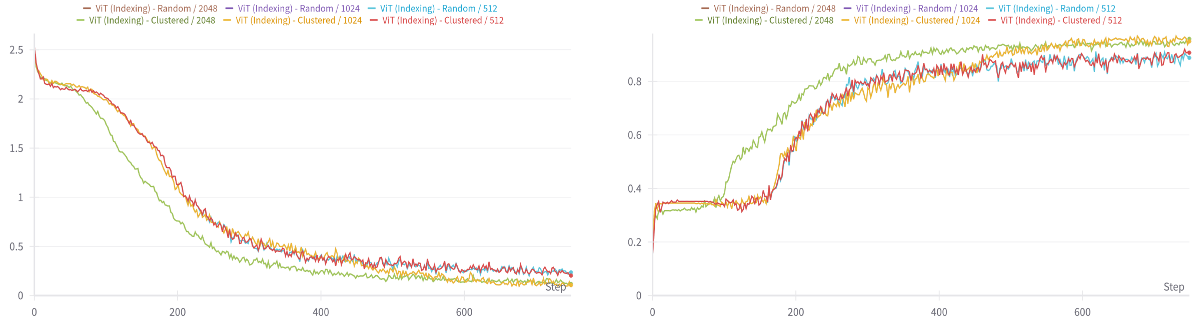
As for the dropout rate of each transformer model variation's training, the value that led to the best performances on both the indexing and retrieval accuracy was found to be 0.1: this is probably because the same dropout rate applies to both the indexing and retrieval training phases, thus a too high dropout rate (above 0.2) seemed to lead to worse indexing accuracy performances (model would not overfit the indexing data enough), while a dropout rate of 0.0 led to worse performances on the retrieval task (worse generalization).

For the remaining training parameters, test results showed that, for every considered dataset and transformer model variation, the best parameters for the "training/performance trade-off" were a transformer embeddings size of 128, a transformer's "fully connected layer" embedding size of 256, a number of "Multi-Head Attention" blocks (depth) of 6, with 4 attention heads each, a learning rate of 0.0005 and a batch size of 32.

3.2.2 Transformers Training Results

The following graphs (Figures 1, 2, 3) show the training and validation loss over epochs for the various ViT models and training approaches presented in Section 3.2.1 (for both the indexing and retrieval training phases) along with the training and validation accuracy (for the retrieval training phase only).

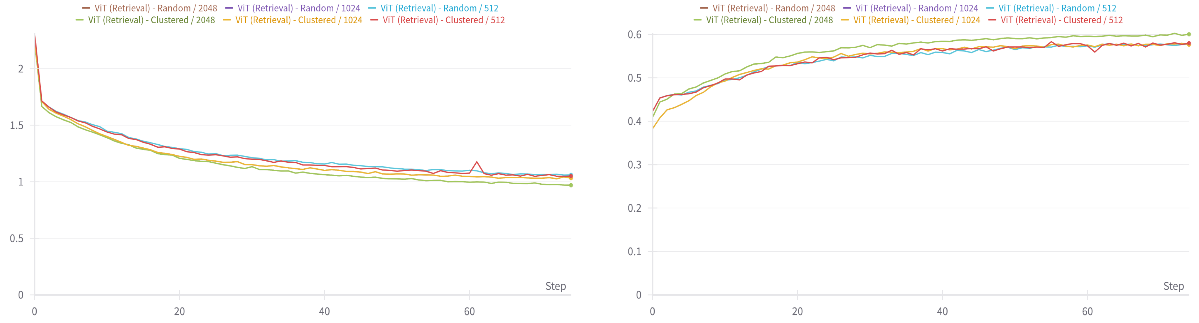
The considered Vision Transformer models' training results refer to all possible combinations of tested database sizes (512, 1024, 2048 total images) and tested indexing variations (either random image IDs or clustered image IDs), with the rest of the parameters, hyperparameters and configurations for both the datasets and the ViT models initialized as described in Section 3.2.1.



(a) Indexing Train Loss

(b) Indexing Train Accuracy

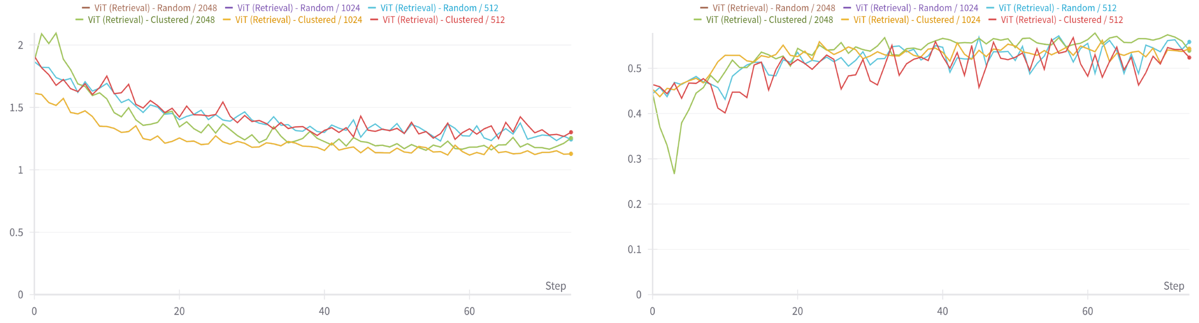
Figure 1: Indexing Phase Training Results



(a) Retrieval Train Loss

(b) Retrieval Training Accuracy

Figure 2: Retrieval Phase Train Results



(a) Retrieval Validation Loss

(b) Retrieval Validation Accuracy

Figure 3: Retrieval Phase Validation Results

4 Results

The following table (Table 1) shows the performance results of the Vision Transformer models, using a "Differentiable Search Index" approach and trained using the approaches mentioned in Section 3.2, compared with the BoVW baselines, using an "index-then-retrieve" approach (see section 3.1). The evaluated ViT models have simple structures and network configurations (described in Section 3.2.1), and were trained on a limited dataset: these constraints may have influenced the evaluation results, but the hardware limitations of our training setting made it impossible to test complex models.

The "Indexing Accuracy" (IA) metric was computed by providing all the images of our indexed database as input to our ViT model, and then checking if the original image ID is found in the list of the retrieved K results (with K=1 and K=5). The final accuracy value is the proportion of correct tests over all tests (thus over the number of images in the database). No "indexing accuracy" results are available for the baseline model because its "index-then-retrieve" nature does not provide indexing features.

The "Mean Average Precision" (MAP) metric was computed by considering the number of results outputted by each model out of K=10 total outputs with respect to the total relevant image IDs of a query image (Average Precision, AP), with the final value given by the mean of the AP calculated on N=10 different image queries (MAP@10).

The "Recall at K" (Recall@K) metric was computed in a similar way by considering the number of relevant outputs out of K=100 total outputs for an image query (Recall@100): in order to reduce the variance of this metric's results, the final value was calculated by considering the mean Recall@100 over 10 different image queries.

Table 1: Models Performance Evaluation

Model	Variations		Results			
Type	**DB Size**	**Database IDs**	**IA@1**	**IA@5**	**MAP@10**	**Recall@100**
BoVW[1]	512	/	/	/	**0.58**	**0.875**
	1024	/	/	/	0.5	0.825
	2048	/	/	/	0.42	0.78
ViT DSI[2] (Fine-Tuned)	512	Clustered	0.415	0.566	**0.32**	**0.58**
		Random	0.402	0.472	0.21	0.365
	1024	Clustered	0.378	0.404	0.28	0.38
		Random	0.266	0.3	0.18	0.265
	2048	Clustered	0.305	0.333	0.17	0.22
		Random	0.233	0.275	0.07	0.135
ViT DSI[3] (Zero-Shot)	512	Clustered	**0.833**	**0.962**	0.18	0.275
		Random	0.794	0.871	0.1	0.23
	1024	Clustered	0.763	0.828	0.12	0.21
		Random	0.725	0.766	0.07	0.175
	2048	Clustered	0.678	0.774	0.1	0.195
		Random	0.533	0.719	0.06	0.16

The results show how none of the tested DSI-based ViT architectures can beat our "index-then-retrieve"-based BoVW baseline: it is worth noting, however, that computing results for the former is considerably faster than it is for the latter in case of large databases (constant O(1) time instead of linear O(N) time, respectively).

The baseline model also seems to have retrieval performances that only slightly worsen as the size of the database grows: this is because, unlike the DSI-based method, its retrieval performances do not depend on the length of the image IDs, while the Sequence-to-Sequence nature of the ViT model leads to a token-by-token output generation which, with longer image IDs, therefore larger numbers of digits to generate, adds up to a larger error probability already given by the larger number of images in the database.

[1] No machine learning used for the BoVW model

[2] Fine-tuned onto <image,relevant_id> tuples constituting 75% of the total database for 75 epochs

[3] Not fine-tuned onto <image,relevant_id> tuples, only trained for the indexing task on <image,id> tuples of the IDB.

5 Conclusion and Future Works

The conducted study was centered around a Vision Transformer-based DSI framework, faster than traditional "index then retrieve" pipelines, used as a "Neural Inverted Index" for image retrieval tasks. Despite the limited computing resources used for the training of the DSI ViT models, the study showed promising results for this novel image retrieval approach: in the setting of dense clustered image IDs used for the indexing of a small image database, a relatively small Vision Transformer model was capable of achieving good results for both the indexing and the retrieval tasks, in both the "zero-shot" and "fine-tuned" retrieval settings. A zero-shot setting seems to lead to better indexing results in a . A "dense random IDs" setting was also tested and seemed to have achieved promising results.

While none of the tested solutions led to better results than the considered BoVW baseline for the evaluated metrics, it is worth noting that, unlike the "index-then-retrieve" approach (proper of the baseline model) the employed "DSI" approach (proper of the proposed solution) allows to achieve image retrieval in constant time (by means of computing a single output of the Vision Transformer model) rather than O(N) time (with an "index-then-retrieve" approach, we need to compute similarity scores of the given query with respect to all of the items in our database).
Furthermore, the proposed "DSI" approach for image retrieval tasks could be used in real-life applications by simply storing the weights of a Transformer model rather than the whole indexed database (assuming a good indexing accuracy), since the information of each item of the database gets encoded inside the Transformer model itself, hence obtaining a ViT used as a "Neural Inverted Index".

Future expansions of the proposed solution might focus on tackling the problem of clustered image IDs still being required to obtain reliable results on the retrieval task, which would make the applications of a DSI-based Neural Inverted Index less practical (as discussed in Section 3.2.1). A "sparse semantic IDs" solution, also tested in the original DSI paper work by Tay et al. [1], would be a better fit for a real-life scenario, but the solution still leads to the need for longer image ID sequences, thus making the Sequence-to-Sequence ID generation of the ViT model more prone to errors (more tokens to generate translate to a higher chance of errors). Ultimately, the best solution should lead to high indexing and retrieval performances without being dependant on the indexing methods, thus without needing to rely on clustered or semantic image IDs, therefore simply adopting the more general "dense random IDs" indexing approach (discussed in Section 3.2.1).

Because of the widespread use of the Transformer architecture, being pervasive in solving tasks related to different areas, from Natural Language Processing (NLP), to image processing, to audio and video, the proposed DSI solution based on Vision Transformer could also be integrated with other kinds of image retrieval pipelines, starting from the use of natural language queries to retrieve images using a Neural Inverted Index approach.

References

[1] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index, 2022.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.