Aerial Image Indexing and Retrieval Using Facebook AI Similiraty Search

Bianca-Alexandra Ghiorghiu, George-Vlad Enache

***Index Terms*—Image Retrieval, Deep Learning, Triplet Loss, Feature Embeddings, Similarty Search**

I. Introduction

In recent years, the rapid increase in the number of digital images and the necessity for efficient retrieval systems in large-scale databases have become increasingly important. This paper portrays the development of an advanced image retrieval system, leveraging the combination between deep learning architectures and high-dimensional similarity search techniques. At the core of this system lies the integration of a convolutional neural network (CNN), specifically ResNet-50, trained using a triplet loss function, and the use of Facebook AI Similarity Search (FAISS) for efficient indexing and retrieval in a high-dimensional embedding space.

This paper provides a comprehensive description of the system architecture, detailing the integration of deep learning models with efficient indexing techniques for real-world image retrieval applications. The practical efficacy of the system is validated through experiments and visualizations, highlighting its potential for application in digital image libraries, recommendation systems, and other domains where efficient and precise image retrieval solutions are necessary.

II. Theoretical Background

The vast field of deep learning has significantly advanced the capabilities of feature extraction and representation learning, particularly in the domain of image processing. CNNs have emerged as the representative models for such tasks, because of their proficiency in hierarchically extracting features and learning discriminative representations. Particularly, the ResNet architecture, a residual learning framework, is renowned for its efficacy in solving the vanishing gradient problem, thus enabling the training of deeper networks. This architecture forms the backbone of our feature extraction module.

A. ResNet

Residual Networks (ResNet) represent an important improvement in deep neural network architecture, primarily designed to combat the vanishing gradient problem encountered in deep networks. The core innovation of ResNet lies in its residual blocks, which incorporate skip connections to bypass one or more layers. These connections allow the network to learn residual functions with reference to the layer inputs, facilitating easier training of deeper networks by alleviating the vanishing gradient issue.

The ResNet architecture introduces a novel approach to constructing deep neural networks through the use of residual blocks. Each residual block contains a shortcut that skips one or more layers. Traditional neural networks map an input x to an output $H(x)$ directly, whereas ResNet blocks are designed to learn a residual function $F(x) = H(x) - x$. This approach simplifies the learning process, as learning the residual is often easier than learning the unreferenced function.

In practice, the residual block consists of two or three convolutional layers, each followed by batch normalization and ReLU activation. The input x is added back to the output of these layers before the final activation function. This skip connection effectively forms an identity mapping, ensuring that the deeper layers can at least perform as well as the shallower ones, thus preventing the degradation problem. In deeper ResNet variants, such as ResNet-50 and beyond, a bottleneck design is employed. This design uses a 1x1 convolutional layer to reduce the dimensionality, a 3x3 convolutional layer for processing, and another 1x1 convolutional layer to restore the dimensions, optimizing computational resources and model complexity.

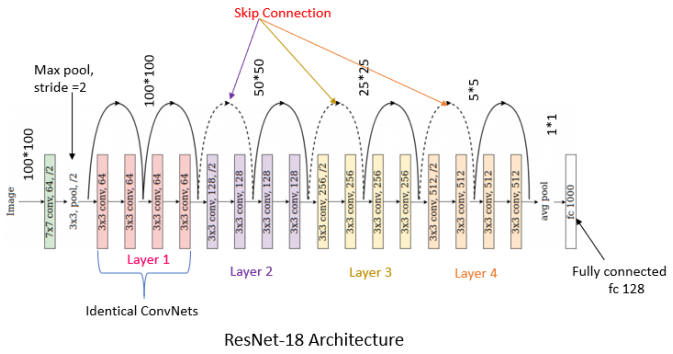

Fig. 1. ResNet Architecture - Consisting of 8 residual blocks, each containing 2 convolutional layers and skip connections [1]

The deeper the network, the more capable it is of learning complex patterns, but this also increases the computational load. ResNet's design allows for significantly deeper networks without the training difficulties typically associated with such depth, thanks to the mitigating effect of the skip connections on the vanishing gradient problem. This has enabled the training of networks with depths previously thought impractical, pushing the boundaries of what is achievable in deep learning applications.

B. Triplet loss

Triplet loss, an objective function specifically tailored for learning a metric space, is employed to train the ResNet-50 model. This loss function operates on triplets of images - an anchor, a positive sample (similar to the anchor), and a negative sample (dissimilar to the anchor).

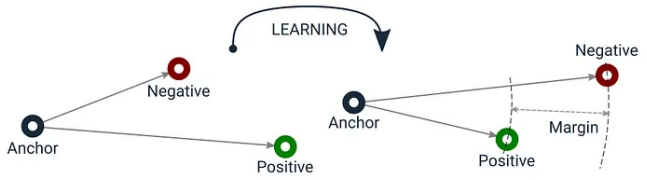

Fig. 2. Triplet Loss Example. The algorithm learns to minimize the distance between the anchor data sample and the positive sample [2]

The objective is to learn embeddings where the distance between the anchor and the positive sample is minimized, and the distance between the anchor and the negative sample is maximized, within a predefined margin.

Mathematically, it is formulated as maximizing the margin between the sum of the distances from the anchor to the positive and from the anchor to the negative:

$$\mathcal{L}_{\text{triplet}} = \max\left(d(\text{A}, \text{P}) - d(\text{A}, \text{N}) + \text{margin}, 0\right) \qquad (1)$$

where

A : the anchor image
P : the positive image
N : the negative image
$margin$: minimum desired separation between the distances (or similarities) of the anchor-positive pair and the anchor-negative pair.

This approach is particularly effective in image retrieval, where the goal is to learn fine-grained similarities between different instances.

C. Similarity Search

Image similarity search involves the retrieval of images from a database that are visually similar to a query image. This process typically employs deep learning models, such as convolutional neural networks (CNNs), to extract high-dimensional feature vectors from images. These vectors capture the visual characteristics of the images, enabling a quantitative comparison. The similarity between images is often determined using distance metrics like Euclidean or cosine similarity. Efficient retrieval in large-scale databases poses a significant challenge due to the high dimensionality and volume of the data.

For the task of similarity search in the embedding space, this paper leverages Facebook AI Similarity Search (FAISS) [3], an optimized library for efficient similarity search and clustering of dense vectors.

FAISS is a library specifically designed to address the efficiency challenges in large-scale similarity search. It operates primarily on the principle of quantization, which compresses high-dimensional vectors into compact codes, significantly reducing memory usage and speeding up the search process. FAISS employs two levels of quantization: coarse quantization for partitioning the search space and fine quantization for more precise representation within each partition. Additionally, it supports different indexing strategies, like IVF (Inverted File) and PQ (Product Quantization), to balance between search accuracy and speed.

The library is optimized for GPUs, further enhancing its performance for large-scale applications. By efficiently handling billions of vectors, FAISS facilitates rapid and scalable retrieval in tasks like image search, recommendation systems, and clustering.

The general usage of FAISS in an image retrieval application is described in Figure 3:

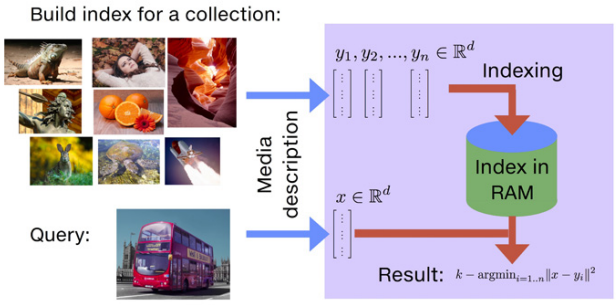

Fig. 3. FAISS based application general flow [4]

The use of FAISS facilitates the rapid retrieval of similar images by efficiently indexing and searching in the high-dimensional space generated by the ResNet-50 embeddings. This indexing is crucial for scalability and high retrieval performance in large image databases.

III. Dataset

The UC Merced dataset is a dataset created by the University of California, Merced. It is designed for land use and land cover classification tasks. The dataset contains 2,100 high-resolution aerial images, with a resolution of 0.3 meters per pixel. These images are divided into 21 distinct classes of land use, such as agricultural, airplane, baseball field, beach, buildings, etc. [5].

Each of the 21 classes contains 100 images of size 256x256x3, providing a balanced dataset for classification tasks. Since its launch, the UC Merced dataset has been widely used in the remote sensing and computer vision research communities for tasks such as image classification, segmentation, and object detection [5]. A few examples from the dataset can be seen in the image below.

Fig. 4. UC Merced dataset sample images from each class

The dataset has been divided into a 70%-15%-15% split, corresponding to the training, validation and test subsets.

IV. Implementation

In order to piece everything togheter and obtain a functional similarity search, the following steps have been implemented:

- **Triplet dataset loading:** The dataset has been re-arranged in triplets, in order to be able to use the triplet loss function. The triplets are formed around each image one-by-one, that becomes the **anchor**. Then, an image from the same class is selected, to serve as the **positive image**. Lastly, a random image from any other random class is selected to serve as the **negative image**;
- **Triplet loss definition:** As described in the previous section, the triplet loss is necessary to train the neural network to learn meaningful feature representations that distinguish between different classes or categories while also making sure that similar images (i.e., images from the same class) are mapped closer together in the embedding space.
- **Neural Network Model Creation:** The network used for feature extraction is ResNet. The selected size is ResNet-50, as it offers a good balance between training time and performance.
  In order to acces the features desired to be used for indexing and retrieval later on, the last fully-connected layer has been removed, offering us acces to an array of 2048 features to be used.
- **Indexing and retrieval:** FAISS has been used for indexing and retrieval of the images selected by the similarity search. From this point onwards, the functionality of the application is described in Fig. 3. The algoritm indexes the features of the training images, generated by the neural network. Afterwards, it is presented with images outside of the training sub-set, which undergo feature extraction using the same network and FAISS retrieves the images with the most similar features.

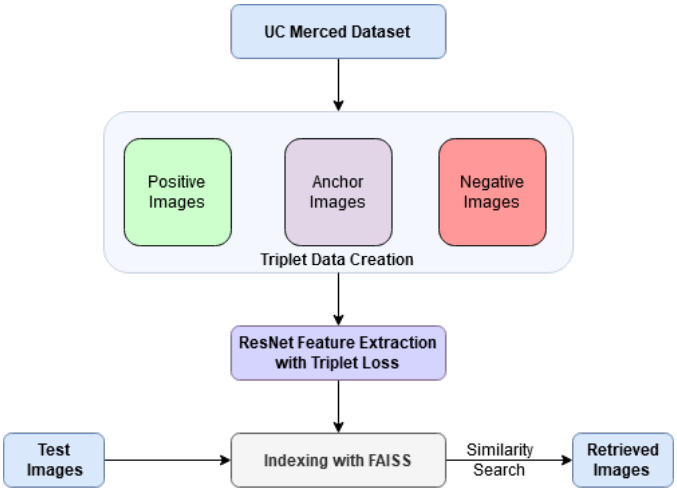

Fig. 5. Application Architecture, encapsuling all the general steps performed

V. Results

While there are no clear metrics for evaluating a similarity search application, as it is an application of subjective nature, the images retieved by the algorithm present a decent degree of visual similarity to the test images used.

Firstly, in Figures 6-7, we can observe the images retrieved for test images obtained from the dataset itself:

Fig. 6. Retrieved Images - Various Random Test Images

Fig. 7. Retrieved Images - Various Random Test Images

Moreover, a small set of custom made images, captured above Bucharest using the satellite view of Google Maps have been put to the test, in order to verify how would the model behave when encountering entirely fresh data. The retrieval appears to be less accurate, comparing to the images from the dataset:

Fig. 8. Retrieved Images - Custom Bucharest Images

We can take a look at the retrieved image classes for the pictures in the test sub-set. To do so, we can observe the histograms depicting the total sum of all of the image classes retrieved for each class folder in the test set.

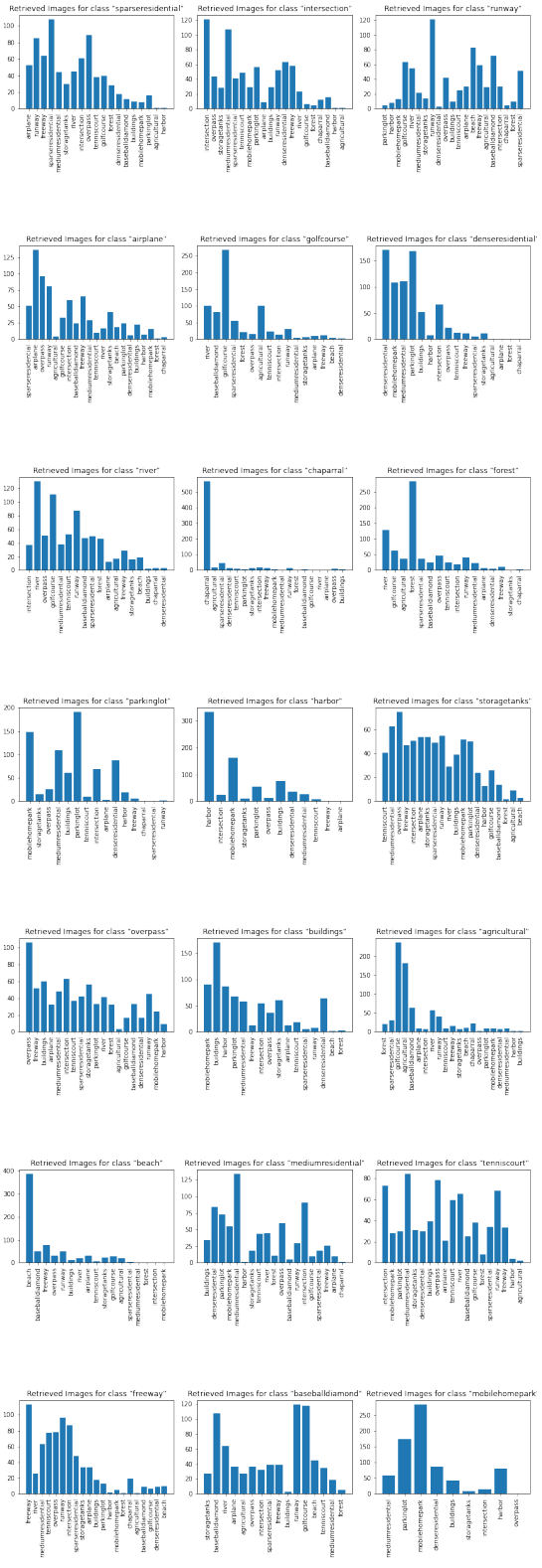

Fig. 9. Total retrieved image classes for each of the classes in the testing set

In order to verify that the classes have been correctly retrieved, we have checked that the distance between the test image and the subsequent 50 retrievals is stricly increasing, making the first retrieved image the "mathematically closest" image to the picture provided, as it can be observed in Figure 10:

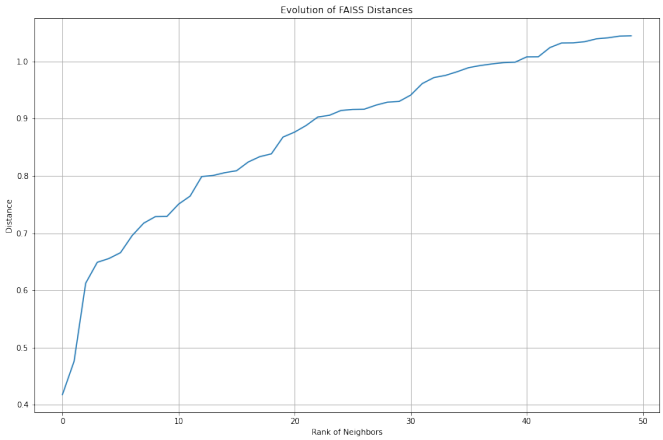

Fig. 10. Distances between a test image and 50 retrieved images

Throughout this evaluation, it has been noticed that some classes are easier to succesfully retrieve than other. Mainly, classes depicting nature landscapes have a far less variation in details, compared to urban images. Therefore, while test images belonging to classes like "Forest", "Beach" or "Agricultural" are tipically returning images belonging to the same class, test pictures from urban landscapes sometimes get mixed up with images that are extremely different. An example of a misretieved image can be seen in Figure 11:

Fig. 11. Misretrieval example

This can be the product of a few possible reasons:

- **Similar Features:** Urban Landscapes might share similar visual features (like concrete structures, cars, etc.), leading the model to confuse between them.
- **Inadequate Training Data:** The dataset features some images with overlapping features to samples belonging to different classes. If the training data does not sufficiently represent the diversity within each category, the model may not learn to differentiate them effectively.
  Ineffective Triplets: Due to the nature of the data, the negative class has to be selected at random, as it is difficult to control the level of dissimilarity of the selected image.The selection of anchor, positive, and negative samples during training is crucial. If negative samples are not sufficiently distinct from the anchor, or if positive samples are not similar enough, the model may not learn the correct boundaries.
- **Domain Shift:** Differences between the training data and real-world data (like lighting, angles, country, etc.) can affect the model's ability to generalize.
- **Algorithmic Bias:** While images have been splitted in a perfectly balanced manner, with each class having an equal number of images, the ratio between urban and natural landscapes is not even. The model might develop biases based on the predominant features of the training dataset, leading to skewed results in certain scenarios.

VI. Conclusions

In this paper, we presented an efficient image retrieval system combining deep learning with similarity search algorithms. The UC Merced dataset was preprocessed and utilized for training a ResNet model with a triplet loss function, focusing on learning discriminative features for image retrieval tasks.

The integration of the trained model with the FAISS library for efficient similarity search was a key aspect of our approach. This combination proved effective in achieving rapid and accurate image retrieval, demonstrating the practical applicability of our method in scenarios requiring both speed and precision. Our results indicate a significant improvement in image retrieval tasks, showcasing the effective combination of deep learning models and advanced search algorithms.

References

[1] Content Moderation using ResNet. https://github.com/RocketChat/content-moderation/blob/master/server/notebooks/PyTorch/moderation_v1(resnet18).ipynb, 2020.

[2] Triplet Loss — Advanced Intro. https://towardsdatascience.com/triplet-loss-advanced-intro-49a07b7d8905, 2022.

[3] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

[4] Jegou Hervé, Douze Matthijs, and Johnson Jeff. FAISS: A library for efficient similarity search. https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/, 2017.

[5] UC Merced Land Use Dataset. http://weegee.vision.ucmerced.edu/datasets/landuse.html.