Science & Technology Development Journal 2023, 26(4):3048-3059

Open Access Full Text Article

Original Research

Image Retrieval Based on ConvNets and the Hashing Algorithm

Tran Hoang Nam, Nguyen Nhat Khoa, Vo Hoai Viet*

ABSTRACT

Image retrieval is a prominent subject of study in the fields of image processing and computer vision. With its application in various domains, such as logo search, product search or general image search in Google, Bing, *etc*., image retrieval has received significant attention for many years. In this work, we study and investigate a framework that leverages visual transferring features and hashing algorithms for the purpose of finding similar images in a dataset. The key idea of our solution is to find the answer to the following question: ``How can we convert an image into binary code and search it more efficiently in a large-scale dataset?''. To achieve this purpose, we use pretrained CNN models from ImageNet for image representation and then convert them into binary code by using hashing algorithms. These images in the dataset are represented by binary codes, and the Hamming distance is used to find the images in the dataset that are indexed. To demonstrate the robustness of the system, we systematically tested the performance of the system based on speed with raw indexing and hashing indexing on 4 datasets: CIFAR-10, Caltech-101, Oxford-102-Flowers, and MS-COCO 2017. The experimental results show that local sensitive hashing (LSH) algorithms with 2,048 bits in binary code demonstrate the same or greater precision than raw indexing. Furthermore, the findings show that the MobileNet architecture consistently outperforms other architectures across these datasets, effectively balancing speed and precision.

Key words: Image retrieval, Hashing algorithm, ResNet, MobileNet

Dept of Computer Vision, University of Science, VNU-HCMC

Correspondence

Vo Hoai Viet, Dept of Computer Vision, University of Science, VNU-HCMC

Email: vhviet@fit.hcmus.edu.vn

History

- Received: 2023-10-07
- Accepted: 2023-12-26
- Published Online: 2023-12-31

DOI :

https://doi.org/10.32508/stdj.v26i4.4193

VNUHCM PRESS

INTRODUCTION

Image retrieval is an important task in computer vision and computer science. This task focuses on answering the question "***How can similar images be found in a database with high precision?***". The overall flow chart of image retrieval systems can be drawn as **Figure 1**. The system divides image retrieval into two-stage implementations: online for retrieval and offline for indexing or training. Each stage is responsible for various tasks, including querying the images or the representation of images through vector embedding from the database system (from one or many hashing tables) and extracting features from image uploads by any user on the platform.

Three main essential components must be considered for feature extraction, feature representation, and similarity matching. Recent computer vision studies have shown interest in these aspects since deep learning outperforms traditional machine learning methods or global and local descriptors.

Global descriptors: Feature extractions are common in each computer vision task because they are the input of many classification algorithms. The more work performed on feature extractions, the better the result is on classification boundaries between classes, as this reduces the semantic gap between the images and the representation after extracting features. [1] reported a color histogram based on a global descriptor that uses the distribution of color channels through statistical methods within a range of image pixel values. Using the magnitude of image pixels and classifying them into bins, the paper can differentiate the similarity of images because the similarity between two images should have an identical distribution. Using the histogram intersection (the magnitude of the image pixels) and classifying them into bins, the similarity of the images can be differentiated because the two images should have identical distributions. The main disadvantage of using color indexing is the limited use of histograms on large datasets, which leads to the combination of Gabor texture features and Fourier descriptors [2] in three different stages. Specifically, by applying the Euclidean distance when classifying patterns and textures between two images, they prioritized the accuracy over the speed when minimizing the error rate using a Gabor filter. After applying a bank of Gabor filters on images with different magnitudes of frequency and orientation, they are able to analyze the texture or obtain features from the image. The structural features are split into structural and statistical types. The structural methods tend to be most effective when images have highly uniform textures since these methods (morphological operators

Cite this article : Nam T H, Khoa N N, Viet V H. **Image Retrieval Based on ConvNets and the Hashing Algorithm**. *Sci. Tech. Dev. J.* 2023; 26(4):3048-3059.

or adjacency graphs) describe the texture by identifying structural units and their interrelationships, while statistical methods (cooccurrence matrices or Gabor filters) extract quantitative features through figures. Some methods achieve better results when combining these two methods [3,4] to increase the query accuracy.

Local descriptor: In contrast to a global descriptor, especially in some contexts, the background from an image can be noise for misclassification since it does not contain useful information. Local feature descriptors are discovered and categorized into two subdomains: gradient-based descriptors and binary descriptors. Generally, gradient-based descriptors perform two main stages, from finding key points in an image to describing key points to feature vectors. In mid-level image processing, scale-invariant feature transforms (also called SIFT) are used as local descriptors since they are rotation and ratio invariant. The main idea of SIFT is to extract invariant features by [5] searching for stable features across a scale space for handling objects of various sizes; subsequently, in [6], feature vectors were applied after the SIFT extractor before the classification step. The main disadvantage of the SIFT algorithm is that the Gaussian kernel is convolved with many scale spaces, is time-consuming and has a high computational cost; moreover, the Gaussian kernel is improved by speeding up robust features (also called SURF), which are three times faster than SIFT is. Replacing the Gaussian filter with a box filter and feature descriptor via the Haar wavelet results in a faster oriented gradient of local neighborhood key points in SIFT. Not only does single SURF use, but another approach [7] is also integrated as a trademark for similar image retrieval. In practice, SIFT and SURF are trade-off options because SIFT yields better accuracy results but slower inference. In [8,9], they implemented a new approach when creating bag-of-words (BOW) for a large database; despite choosing only SIFT, they decided to add binary descriptions and achieved 79.6% accuracy on a holiday dataset.

Deep Learning visual descriptor: Along with low-level features, the features extracted by these approaches above seem to exploit only a small amount of image information. For instance, SIFT and SURF extract only the local pattern and texture in images, while the local-binary pattern (LBP) extracts only texture information and has an efficient computational cost. With the rise of GPUs to meet hardware requirements, deep learning has gradually replaced the dominance of hand-feature extractors with the breakthrough of convolution neural networks (CNNs) [10] in the image field. The replacement of SIFT [11,12] for extracting the local information with different ratios to binary descriptors was compared with the baseline of VGG16 [13] for classifying the commonalities in the dataset.

From the trivial approach, a Siamese network has been developed for producing models for both feature extraction and image retrieval [14–16]. They proposed the Siamese network for similarity matching with a limited source of data, which is valuable for few-shot or one-shot learning. However, because of the sharing of parameters between two identical networks, more training time is needed for these networks than for traditional CNNs. In today's context, the diversity and versatility of the data we collect every day make it difficult to select and preprocess the data efficiently. In some specific fields, especially in medical image processing, only numerous instances of healthy body parts of humans exist, and the opposite examples are lacking. The development of generative adversarial networks (GANs) has led to new strategies for image retrieval for unsupervised learning. A group of researchers [17–19] for solving domain classifiers via look-alike retrieval have proven its effectiveness since it yields competitive results with previous related methods in these fields.

In this research, we focus on researching and applying deep learning in visual representations for image retrieval systems. The main contributions of this research are as follows: i) propose a framework for image retrieval based on deep learning features and hashing algorithms; ii) evaluate the proposed method on 4 datasets, CIFAR-10, Caltech-101, Oxford-102-Flowers, and MS-COCO 2017; and iii) test these difference deep learning features and lengths of bits in hashing code to determine the best practices for application.

OUR PROPOSED METHOD

Overview of our framework

Figure 2 shows the overall framework pipeline for a standard image retrieval system. The content-based image retrieval (CBIR) framework acts as a search engine for image data types. In the CBIR framework, a user specifies a query image and obtains the images in the database as the query image. The gray part represents the offline stage of indexing images in the database system. The remaining stage involved retrieving images in the database system. In the production stage, when running in a real project, these stages iterate repetitively until enough information is provided to users.

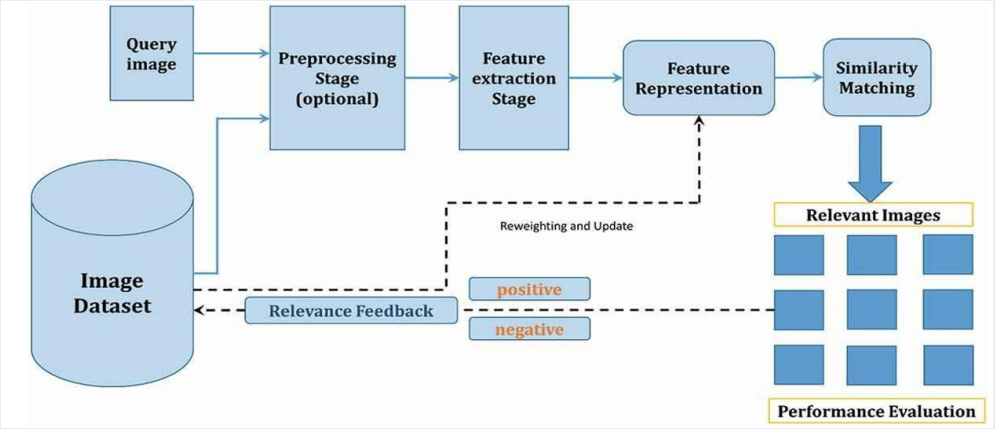

Figure 1: **Overall of image retrieval system framework.**

In the offline stage, we adapt the transfer learning method for knowledge distillation as a feature extraction step to provide additional visual features before the hashing step. We can apply both methods for the pretrained model: apply as a fixed feature extractor or fine-tune the network with a case of retraining data similar to the original dataset or a new small dataset that makes minor changes to the overall parameters. In contrast to the online stage, whenever users upload or provide an image for searching, these images are encoded after forwarding the model checkpoint of the pretrained model, thus creating the representation feature from images.

Feature extraction and image representation.

The main objective of the feature extraction stage is to create a representative vector of semantic meaning of the image. The input images have the same resolution with three color channels of $R^{w\times h\times 3}$, where w×h represents the size of the input image. After propagating through the models for extraction, we obtain a feature vector n in length. Through deep learning layers, we can aggregate from a low level as a pattern or detect the edge or boundary of an image to high-level features as a part or whole object in an image. A wide range of models are available for this purpose; we propose using ResNet-18 and ResNet-50 as baseline models.

Resnets are used as the SOTA models of CNN methods for computer vision tasks because they solve the vanishing gradient in very deep neural networks, as occurs during training when the gradients used to update the network become extremely small or approach zero. Theoretically, the more layers added to a neural network, the more likely the model performance is to increase or remain the same; however, in practice and throughout experimentation, a diminishing return occurs, thus decreasing the accuracy of the model and limiting its learning capacity. Using the shortcut connection with the identity function, we can overcome the function of the hidden state at layer t as h(x)=Relu(f(x)+x) as the input is passed directly to the output, acting as a mapping identity function, avoiding degradation issues in stacking deeper networks, as shown in **Figure 3**.

Moreover, the compact size of MobileNet allows it to be used in mobile applications since it is lightweight but still achieves favorable results. Moreover, deep learning algorithms can generate new features from among a limited number of samples located in the training dataset without additional human intervention. By combining handcrafted features, self-learning can adjust the weight inside each node to allow deep learning to mine its own errors. The layer convolution (known as pointwise convolution) introduced in [20] can be considered feature pooling or channelwise spatial convolution. Instead of multiplying all channels as in traditional convolution, we split the feature map into multiple groups, and each group is fixed in channel size to reduce the number of parameters in the filters. Inspired by group convolution, depthwise convolution is used in MobileNet to split feature maps to each channel size. After depthwise convolution, pointwise convolution is applied to change the dimensions, as shown in **Figure 4**.

Binary hashing algorithm

The motivation for using binary code for mapping feature vectors is its application in information technol-

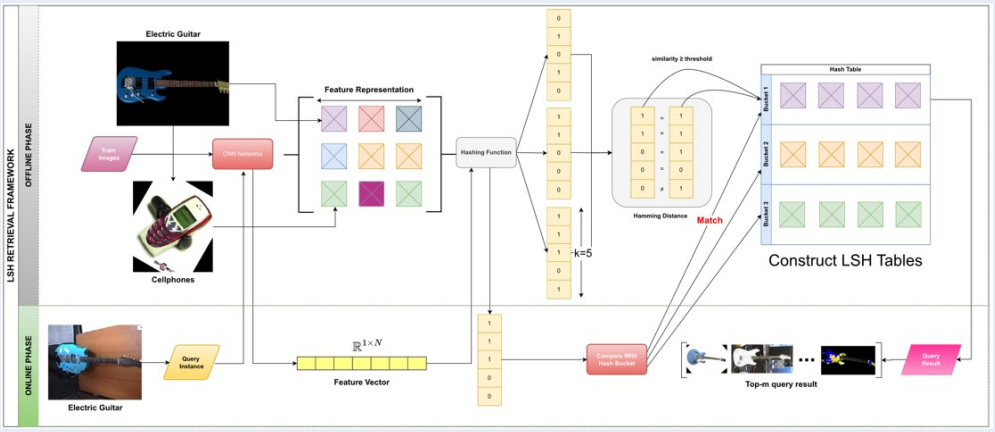

Figure 2: **Our CBIR framework based on hashing indexing.**

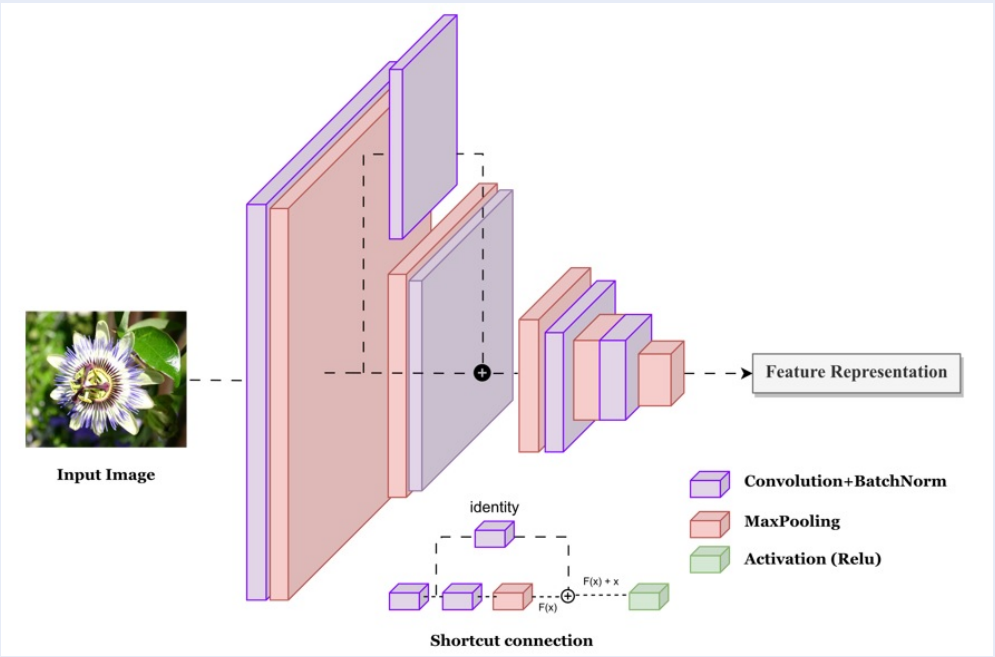

Figure 3: **The residual network (ResNet) architecture for feature extraction and representation**.

ogy and, remarkably, in information security. Normally, when directly applying a feature after propagating a neural network, the flattening vector is usually large, as shown in **Table 1**. The lightweight hashing code takes advantage of handling large datasets as well as training new incoming data. However, in the medical field, imaging is challenging when these algorithms are applied when retrieving images with high resolution to distinguish them from illness and wellness at the cellular level (cells acting as tumors or cancer cells in imbalanced datasets); moreover, it is better to apply these algorithms to retrieve general body parts instead of a greater level of detail.

In the CBIR context, the purpose of the system is to search for related images that have the closest semantic meaning to the query images. There are several drawbacks in the case of variance in affine transformation and rotation, such as the computational cost of handling images of different sizes and resolutions when directly comparing images at the pixel level. In this research, we focused on the progress of creating indexing with different aspects. First, we pro-

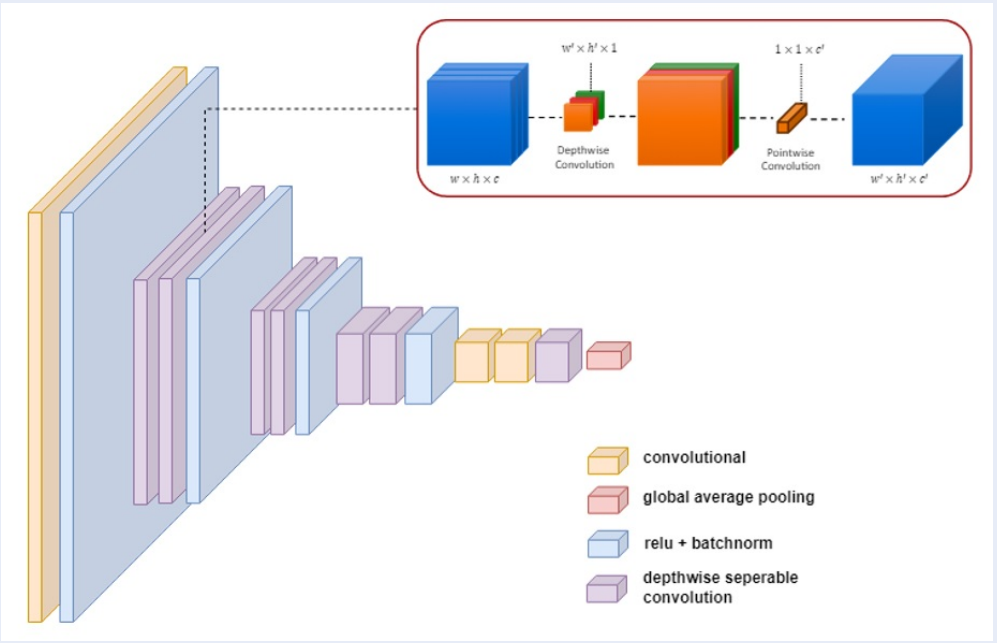

Figure 4: **Illustration of the MobileNet architecture**.

Table 1: **The different vector features of the deep learning models before the softmax layer**

Model	Feature vector (Before softmax)
VGG16/VGG19	4096
Resnet18/Resnet34	512
Resnet50/Resnet101	2048

pose the use of linear indexing, a method in which an array is used for saving data. In the retrieval stage, data are browsed by sequentially iterating over an array and comparing them with a query from the user. Although this search strategy works well with any kind of data if the appropriate metric is used, time is an intractable problem and inefficient for large database systems. In CIFAR-10 dataset, an image with size of 960×7×7 cost 5.5Gb space resource proves it is not a profitable choice when dealing with high-dimensional data. To overcome the weakness of the speed and memory footprint of linear indexing, we propose using hashing binary code, and local sensitive hashing is one of the typical algorithms for this problem.

Local-sensitive hashing (LSH) is an algorithm for addressing the problem of approximate nearest neighbor search in high-dimensional space. It converts data points from high-dimensional space to hashing code in different hash tables whether two data points have the same label as the same table. LSH differs from other hashing methods in minimizing the conflict between data points, while other hashing methods are not available for approximate searches. In the nearest neighbor search problem, given a set of Ps that include n points in d-dimensional space, inherited from [21] when applying the indexing method for approximate nearest neighbors, we applied LSH to indexing feature data from the previous feature extraction step. Suppose that we have two points p,q and that the probability of hashing p,q to the same cluster is directly proportional to the distance p,q in all spaces, whether the distance is below threshold t, as provided in **Figure 5**. Like other studies on the same application of using LSH in indexing [22], we prefer to use ResNet as a state-of-the-art model in deep learning and Mobile-Net to reduce the size of the feature extraction approach. In contrast to [23], we also experiment on different datasets with various bit lengths to check the hashing capability before and after propagating the flattening layer.

EXPERIMENTAL SETTINGS

There are many datasets for empirical study and observation of CBIR systems. In this implementation, we apply our running and testing methods on the CIFAR-10, Caltech-101, Oxford-102-Flower and MS-COCO 2017 datasets, as shown in **Table 2**. In the

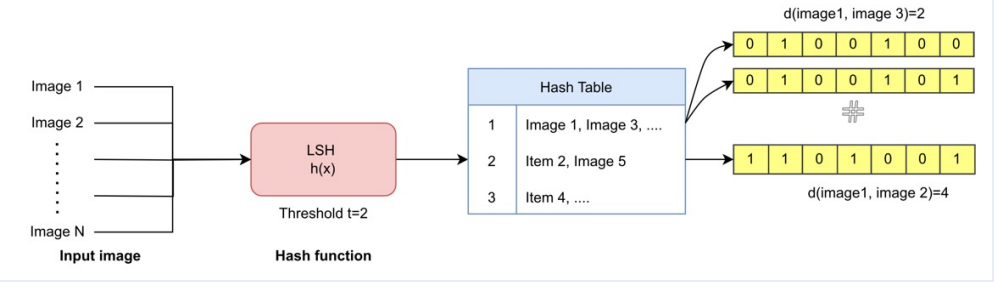

Figure 5: **The pipeline of the LSH algorithm for binary codes and indexing.**

CIFAR-10 dataset, there are 50K images for training and 10K images for validation; however, in this research, only 40K images are used as the dataset for queries, and 10K images are used as query images. Along with Caltech-101, each class contains between 40 and 800 images, for a total of 9K images for the whole dataset. In this case, this study decided to split at a ratio of 8:2 for the dataset and query images, corresponding to 7316 and 1829, respectively, for querying. Oxford-102-Flowers, a dataset created by Oxford, included 102 common flower species in the United Kingdom. The last dataset is MS-COCO 2017, a dataset of common objects in context collected by Microsoft. These objects serve multiple computer vision tasks, such as object detection, image captioning and image segmentation.

As described in **Figure 6**, the samples contained in the four datasets are diverse enough to test our hypothesis. For the Caltech-101 dataset and CIFAR-10 dataset, the objects were distinguished from other classes with a clean background. With images of shape color images in 10 classes, CIFAR-10 tends to have a smaller image resolution than the remaining datasets. The MS-COCO 2017 has a wide range of objectives with complex backgrounds for image retrieval.

In artificial intelligence, particularly models serving computer vision, the empirical results were judged on various evaluation metrics for optimizing parameters and model architectures. Generally, there are three common metrics for many purposes: thresholding, model performance and probability measurement [23,24]. Evaluating classifiers requires careful consideration, and one of the most widely used metrics for classification is accuracy, which is calculated as follows:

$$Accuracy = \frac{Correct\,predictions}{Total\,predictions} \quad (1)$$

Suppose the dataset from real life contains an imbalance of data between classes; accuracy is not always a great measure since when the minority class has plenty of samples still achieves high accuracy. To help address these problems, researchers often use confusion matrices to visualize the performance of classification models. The model was built on matrix $R^{n\times n}$, where n is the number of labels in the dataset. Suppose that there is a binary classification with ground truth y and predicted label ỳ, as shown in **Table 3**.

The confusion matrix provides additional information, detailed as class performance in binary classification. From Table 3, we address a formula called precision and recall, which remedies the weakness of using accuracy when dealing with an imbalanced dataset.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Specifically, the precision and recall provide much more information about the model performance. According to the equation, the precision is calculated as the ratio of the number of correctly predicted positive classes to the number of all items predicted to be positive. Sometimes, precision is more important for false positives than for false negatives (*e.g.*, spam detection), while recall focuses on false negatives (*e.g.*, adult prohibitive content). In an image retrieval system, for querying relevant documents, these metrics become precision-recall at k (the number of retrieved documents); in this case, precision refers to the relevant document based on k return results, while recall refers to the ratio of the number of relevant items retrieved. In the parameter optimization process, models usually perform a tradeoff between precision and recall, finding a relatively optimal value based on the context of model usage with a precision–recall curve. The precision and recall always run in opposite directions. The precision–recall curve was generated

Table 2: Dataset settings for image retrieval evaluation

Dataset	Number of Classes	Image size (Image)	Query Size (Image)
CIFAR-10	10	40000	10000
Caltech-101	101	7316	1829
Oxford-102-Flowers	102	6552	818
MS-COCO 2017	91	118287	5000

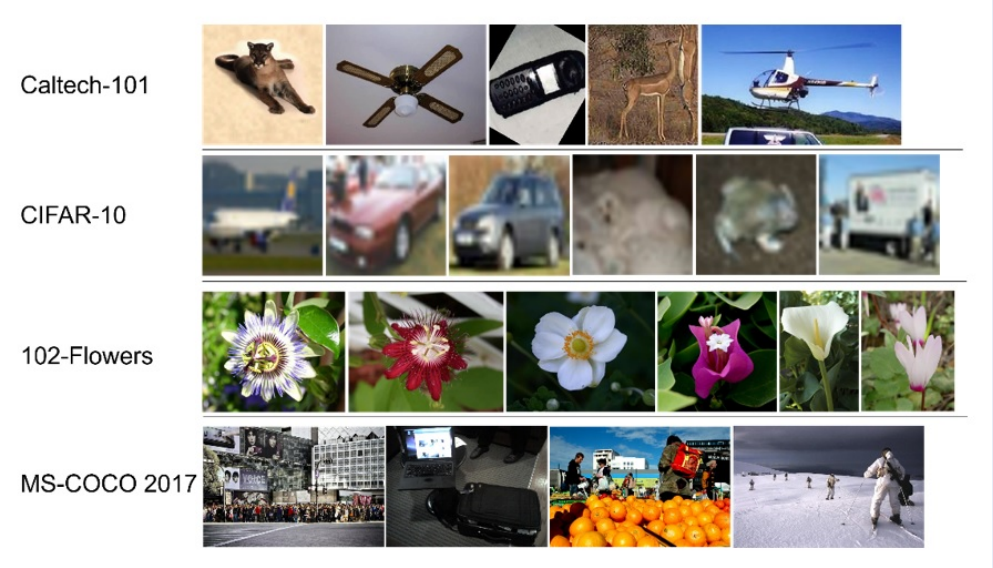

Figure 6: Some sample images from the 4 datasets.

Table 3: Confusion matrix

		Predicted Values	
		Positive	Negative
Actual Value	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

to indicate the differences in the model classification thresholds. The term average precision (AP) represents the retrieved result based on its relevant score. The mAP is a well-known performance metric used to evaluate machine learning models in image retrieval. The average precision is the average precision value at all ranks where relevant documents are found.

$$AP = \frac{\sum_{k=1}^{n} [P(k) \times rel(k)]}{number of retrieved items} \quad (5)$$

EXPERIMENTAL RESULTS AND DISCUSSION

For standard comparison, this research applies two baseline methods, CNN deep feature-based methods and hashing-based methods. For deep learning-based models, we prioritize transfer learning methods with off-the-shelf models for general judgment. Then, we apply a fine-tuning model to the MS-COCO 2017 dataset to ensure that the domain knowledge for each image belongs to a specific label in feature extraction by eliminating the final softmax layer. For the evalu-

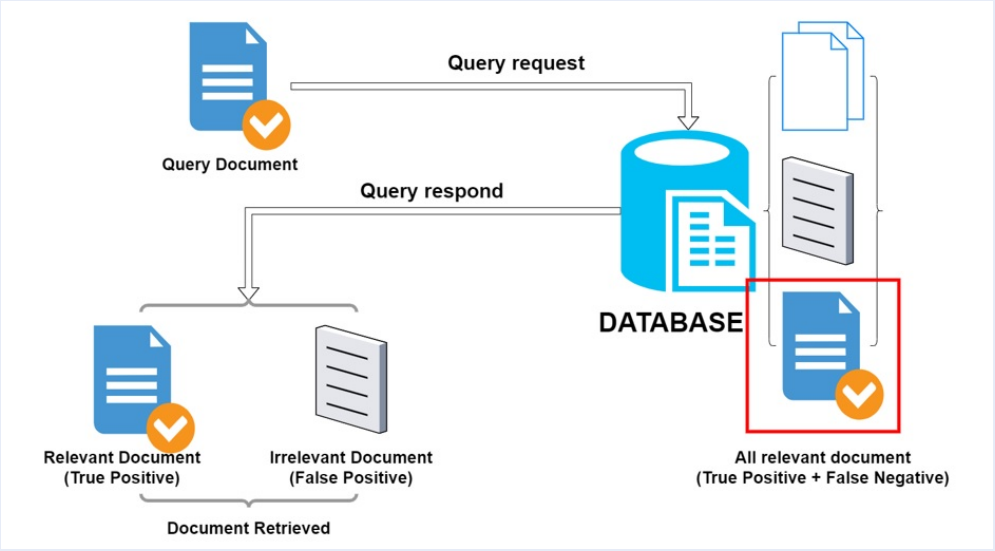

Figure 7: **Relevant document retrieval evaluation metric.**

ating method, we propose taking the top 5 query results and redeem them relevant whether one of them includes the ground truth of the image. Using the Euclidean distance, future vectors are mapped into a flattened array, and a sequential search is applied as brute-force querying. Every new extracted feature vector is appended to the indexing list to determine whether there is new incoming data.

The query result from the CNN based on **Figure 8** shows the difference between the two methods. According to the CNN results, all five query results are considered relevant, while the LSH-based results exhibit slightly lower performance on two irrelevant cases in the Caltech-101 dataset. This phenomenon is identical to that of a hard dataset, which has many objects in one image in the MS-COCO 2017 dataset. Although better accuracy scoring is achieved, the size of indexing files is more of a concern when dealing with large datasets. ResNet50 in the CNN descriptor approach has a 100-fold greater file capacity than does the LSH approach, which has a 128-bit length and is currently more efficient at storing large amounts of media data.

After the top 5 related images are obtained, when more k-top query results are considered, the larger search space of LSH tends to have more irrelevant results. In the context of using a CNN descriptor, the longer the hashing code is, the better the number of relevant results given by the system since long hashing maintains more information about images. In the first approach, we align the CNN feature vector as a pipeline for measuring the baseline approach. The mAP results indicate that at a bit length of 2048, LSH yields the same result as does raw indexing when deep learning features are used with less space for storage. At the object level, when retrieving images from the Caltech-101, CIFAR-10 and 102-Flowers datasets, ResNet18 achieved mAP scores of 74.3%, 53.1% and 58.8%, respectively, while at the context level, the highest mAP score was achieved at 6% in the MS-COCO 2017 dataset. The main reason for choosing ResNet and MobileNet is model compression; as the purpose of decreasing the size of the model architecture, we can assume an approximate result when increasing the number of bits in the LSH approach.

Second, in a hashing-based approach, various feature vector sizes are mapped into 128 bits of length in LSH. Although ResNet50 is larger in terms of dimensions, it seems to have lower results than the other methods and proposes the hypothesis that when there are too many features, the discrepancies between images within the same class are farther apart, resulting in a decrease in performance. The LSH method is suitable for determining the trade off between storage capacity and accuracy in CBIR systems. In the case of large databases such as MS-COCO 2017, indexing via LSH notably differs from raw indexing via features from deep learning models.

As shown in **Table 5**, as the bit length increases, the results gradually achieve the same performance as that of the CNN feature vector-based approach. In the context of using ResNet50, the result surpasses that of

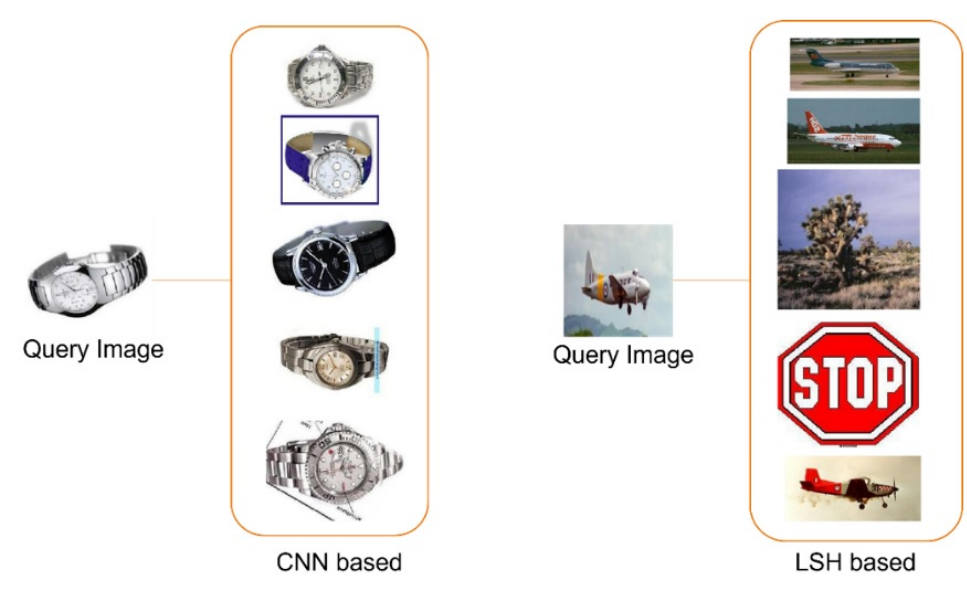

Figure 8: Example of the Top-5 retrieval results of individual approaches on the Caltech-101 dataset.

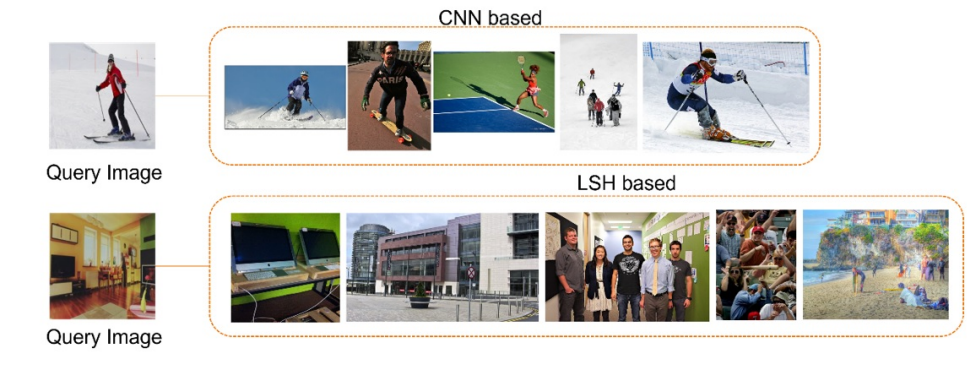

Figure 9: **An example of a Top-5 retrieval result from the MS-COCO 2017 dataset.**

Table 4: mAP of the feature-based CNN descriptor with k = 5

	Resnet18	Resnet50	MobilenetV3_small	MobilenetV3_large
Caltech-101	74.3	24.3	34.2	23.3
CIFAR-10	53.1	8.1	16.7	7.5
102-Flowers	59.8	4.3	14.1	3.3
MS-COCO 2017	76.7	27.4	38.5	31.9

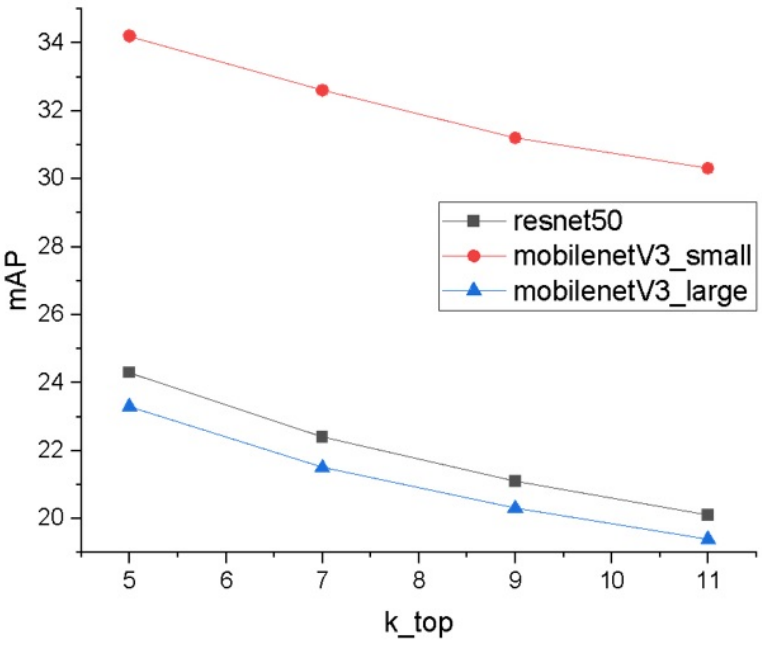

Figure 10: **mAP on the Caltech-101 dataset on the k-top query result.**

Table 5: **mAP on different datasets using LSH with 128 bits**

	Resnet18	Resnet50	MobilenetV3_sm	MobilenetV3_large
Caltech-101	52	7.5	4.9	3.5
CIFAR-10	29.7	5.8	6.2	5.5
102-Flowers	29.6	0.9	1.1	1
MS-COCO 2017	3.4	0.16	0.08	0.11

the baseline method while having a smaller indexing file size (17.7 mb compared to 57.1 mb). The LSHs are not highly sensitive to noise in the data. A similarity search is performed by mapping the same (or nearby) hash buckets with high probability, even if there is noise in the data. The noteworthy result of ResNet50 when achieving 25 as opposed to 24.3 in feature vector approaches remains our proof on different datasets and other model options. 102-Flowers have moderate results, as some classes in the dataset having identical patterns for hashing lead to class distances in the search space being quite close to each other.

In the retrieval stage, we choose a distinctive layer for feature vector representation. Choosing between the layer before the fully connected layers or the last layer would achieve better semantic-based indexing than feature-based indexing. In our experiment, we implement 3 different layers with a wide range of bit lengths from 16 bits to 2048 bits. It seems that flattening yields a better description of feature presentation than does the representation of semantic meaning in construct buckets.

CONCLUSION

This paper has researched and developed an image retrieval system based on key ideas that combine deep visual features and hashing algorithms. The main contribution of our research is the proposed framework for image representation with binary code from deep visual features, which can search for semantic similarity among large-scale image datasets. This paper researched and applied two state-of-the-art deep learning architectures, ResNet and MobileNet, for the feature extraction stage. These visual features are fed into the LSH algorithm to transform into binary code

Table 6: mAP for various bit lengths of the LSH

	Bits Length	Caltech-101	CIFAR-10	102-Flowers	MS-COCO2017
MobileNetV3-Large	256	4.8	6.8	0.8	27
	512	7.9	7.3	1.1	27.8
	1024	8.9	8.2	1.2	28.3
	2048	12.9	9.2	1.5	29.9
MobileNet	256	7.7	6.8	1.4	27.3
	512	8.8	7.3	1.5	28
	1024	13	8.2	2.7	28.1
	2048	17.7	9.2	3.7	28.5
ResNet50	256	9.6	6.4	0.96	31
	512	16	6.6	1.7	34.8
	1024	21.9	7.6	2.3	37.6
	2048	25	8.2	3.4	38.7

Table 7: Different bit lengths in the two last layers of the model architecture

Model	Caltech-101		
MobileNetV3-Large	Bits Length	Flatten	FC-CLF
	256	13	3.9
	512	17.7	4.3
	1024	22.2	7
	2048	23.8	8.6

for speed and scalability in searching. A comprehensive experiment is tested on four datasets (CIFAR-10, Caltech-101, Oxford-102-Flowers, and MS-COCO 2017), and multiple lengths of bits are used in binary code. The results show that the binary code of 2048 bits and the ResNet architecture have the best performance on our pipeline. Moreover, the MobileNet proves that the architecture is stable and robust when balancing speed and mAP. However, the limitations of this system include the use of multiple stages for binary codes for each image. For further research, we will investigate a deep architecture that can generate binary code directly by training using labeled image data from one stage in future work. We also consider combining deep visual code with GPUs or distributed computing to further improve the speed and scalability of the retrieval system.

ABBREVIATIONS

LSH: Local Sensitive Hashing
SIFT: Scale-invariant Feature Transform
SURF: Speed Up Robust Feature
BOW: Bag of Words
LBP: Local Binary Pattern
ResNet: Residual Network
CNN: Convolution Neural Network
ConvNets: Convolution Neural Networks
GAN: Generative Adversarial Network
VGG: Visual Geometry Group
SOTA: State of the art
CIFAR-10: Canadian Institute for Advanced Research
MS-COCO 2017: Microsoft Common Object in Context 2017
Caltech-101: California Institute of Technology 101
GPUs: Graphics Processing Units
CBIR: Content - Based Image Retrieval
TP: True Positive
FP: False Positive
FN: False Negative

TN: True Negative
AP: Average Precision
mAp: Mean Average Precision

ACKNOWLEDGMENTS

This work was supported by Computer Vision Dept, Faculty of Information Technology, University of Science, VNU-HCMC, ImgLabs, and Zetgoo through Intern Research Program 2023.

AUTHOR CONTRIBUTIONS

Tran Hoang Nam has contributed in coding algorithms, conducting experiments, and reporting results. Nguyen Nhat Khoa has contributed in coding algorithms, analyzing the experimental results, and composing the manuscript. Vo Hoai Viet has made significant contributions in making the literature review, algorithms, conducting experiments, interpreting the data, analyzing the experimental results, and composing the manuscript to be submitted. All authors read and approved the final manuscript.

COMPETING INTERESTS

The authors declare no conflict of interest.

REFERENCES

1. Swain MJ, Ballard DH. Indexing via color histograms. In: Proceedings of the third international conference on computer vision. Vol. 1990; 1990:390-3;Available from: https://doi.org/10.1109/ICCV.1990.139558.
2. Shrivastava N, Tyagi V. An efficient technique for retrieval of color images in large databases. Comput Electr Eng. 2015;46:314-27;Available from: https://doi.org/10.1016/j.compeleceng.2014.11.009.
3. Song X, Li Y, Chen W. A textural feature-based image retrieval algorithm. In: Fourth International Conference on Natural Computation. Vol. 2008; 2008;Available from: https://doi.org/10.1109/ICNC.2008.153.
4. Bani N, Fekri Ershad S. Content-based image retrieval based on combination of texture and color information extracted in spatial and frequency domains. The electronic library, Vols. ahead-of-print, August 2019;.
5. Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. November 01 2004;60(2):91-110;Available from: https://doi.org/10.1023/B:VISI.0000029664.99615.94.
6. Bakar SA, Hitam MS, Yussof WNJHW. Content-Based Image Retrieval using SIFT for binary and grayscale images. In: IEEE International Conference on Signal and Image Processing Applications. Vol. 2013; 2013;Available from: https://doi.org/10.1109/ICSIPA.2013.6707982.
7. Tripathi RK, Agrawal SC, "A shape and texture features fusion to retrieve similar Trademark Image Material," IOP Conference Series. Mater Sci Eng. April 2021;1116:012026;Available from: https://doi.org/10.1088/1757-899X/1116/1/012026.
8. Zheng L, Wang S, Tian Q. Coupled binary embedding for large-scale image retrieval. IEEE Trans Image Process. 2014;23(8):3368-80;PMID: 24951697. Available from: https://doi.org/10.1109/TIP.2014.2330763.
9. Zheng L, Yang Y, Tian Q. SIFT meets CNN: A decade survey of instance retrieval. IEEE Trans Pattern Anal Mach Intell. 2018;40(5):1224-44;PMID: 29610107. Available from: https://doi.org/10.1109/TPAMI.2017.2709749.
10. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;.
11. Gong Y, Wang L, Guo R, Lazebnik S. Multiscale orderless pooling of deep convolutional activation features; 2014;Available from: https://doi.org/10.1007/978-3-319-10584-0_26.
12. Jégou H, Douze M, Schmid C, Pérez P. Aggregating local descriptors into a compact image representation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2010; 2010;Available from: https://doi.org/10.1109/CVPR.2010.5540039.
13. Lin K, Lu J, Chen C-S, Zhou J. Learning compact binary descriptors with unsupervised deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 2016; 2016;PMID: 27319213. Available from: https://doi.org/10.1109/CVPR.2016.133.
14. Liu Y, Ding L, Chen C, Liu Y. Similarity-based unsupervised deep transfer learning for remote sensing image retrieval. IEEE Trans Geosci Remote Sensing. 2020;58(11):7872-89;Available from: https://doi.org/10.1109/TGRS.2020.2984703.
15. Rahman A, Winarko E, Mustofa K. Product image retrieval using category-aware Siamese convolutional neural network feature. J King Saud Univ Comput Inf Sci. 2022;34(6):2680-7;Available from: https://doi.org/10.1016/j.jksuci.2022.03.005.
16. Gordo A, Almazán J, Revaud J, Larlus D. Deep image retrieval: learning global representations for image search. In: European Conference on Computer Vision; 2016;Available from: https://doi.org/10.1007/978-3-319-46466-4_15.
17. Zhang H, Sun Y, Liu L, Wang X, Li L, Liu W. ClothingOut: a category-supervised GAN model for clothing segmentation and retrieval. Neural Comput Appl. 2020;32(9):4519-30;Available from: https://doi.org/10.1007/s00521-018-3691-y.
18. Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks; 2018;Available from: https://doi.org/10.1109/CVPR.2017.632.
19. Pahariya G. Bi-modal content based image retrieval using multiclass cycle-GAN. Digit Image Comput Tech Appl (DICTA). 2018;2018:1-7;Available from: https://doi.org/10.1109/DICTA.2018.8615838.
20. Brown CD, Davis HT. Receiver operating characteristics curves and related decision measures: A tutorial. Chemom Intell Lab Syst. 2006;80(1):24-38;Available from: https://doi.org/10.1016/j.chemolab.2005.05.004.
21. Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing. In: Very Large Data Bases Conference; 1999;.
22. Varga D, Szirányi T. Fast content-based image retrieval using convolutional neural network and hash function. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC). Vol. 2016; 2016;Available from: https://doi.org/10.1109/SMC.2016.7844637.
23. Powers DMW. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Arxiv, vol. abs/2010.16061. 2011;.
24. Luo Y, Li W, Ma X, Zhang K. Image retrieval algorithm based on locality-sensitive hash using convolutional neural network and attention mechanism. Information. 2022;13(10);Available from: https://doi.org/10.3390/info13100446.