Int. j. inf. tecnol. (December 2023) 15(8):4125–4130
https://doi.org/10.1007/s41870-023-01437-x

ORIGINAL RESEARCH

Vision-based image similarity measurement for image search similarity

Werapat Jintanachaiwat[1] · **Thitirat Siriborvornratanakul**[1]

Received: 14 February 2023 / Accepted: 31 July 2023 / Published online: 19 September 2023

Abstract In various applications across different platforms, image similarity features such as image searching and similar image recommendations are widely used. However, the challenges of semantic gap and querying speed continue to pose significant challenges in image similarity searching. In this study, we propose a novel solution to address these issues using contrastive learning within the TensorFlow Similarity library. Specifically, we trained and tested our proposed method using the Caltech-256 dataset and further evaluated it on the Corel1K dataset. Our work distinguishes itself from previous studies that primarily focus on evaluating accuracy while neglecting the importance of speed evaluation. As such, we propose evaluating both the mean average precision score and query time spending. Our experimental results reveal that our method based on EfficientNet (B7) yields the best average precision scores of 0.93 on the Caltech-256 test dataset and 0.94 on the Corel1K dataset. However, other methods achieve faster query times, although their average precision scores are significantly lower.

Keywords Artificial intelligence · Deep learning · Image similarity · Image search · Image representation

1 Introduction

Image searching is a tool used by users to explore related visual content. Despite the progress made in image searching by image, challenges related to the semantic gap and search speed still need addressing to meet business needs. The semantic gap is a subjective issue, as some image pairs may be visually similar but differ semantically. Previous studies have proposed different methods for image searching by image, ranging from traditional image features [3, 5], machine learning, and deep learning. These studies involve extracting image features and calculating distances to identify the closest matches. However, even with the application of machine learning and neural networks, the semantic gap issue still persists, as highlighted in [8] and [28].

To address the semantic gap and search speed issues and improve image searching by image, we propose a solution that balances accuracy and speed using the Tensorflow similarity library [7]. As a core model, we used the pre-trained EfficientNet (B7) whose last three original layers were further trained with the Caltech-256 dataset. Tensorflow similarity is used in our work as it leverages contrastive learning and offers Fast Approximate Nearest Neighbor that supports scalability for efficient and fast searching in sub-linear time. Our study seeks to find a solution that provides similar images as the input image with high-speed searching. However, our work is restricted to the use of one baseline model at a time. Utilizing feature vectors from an ensemble model like [20] may enhance accuracy at the obviously higher computational cost.

2 Related works

In the field of image retrieval, both text-based [9, 11, 17, 18] and image-based methods have been proposed for image similarity search. However, this section will mainly concentrate on image-based retrieval. [15] proposed the feature-based sparse representation for image similarity using Scale-Invariant Feature Transform (SIFT) features,

✉ Thitirat Siriborvornratanakul
thitirat@as.nida.ac.th

Werapat Jintanachaiwat
werapat.jin@stu.nida.ac.th

[1] Graduate School of Applied Statistics, National Institute of Development Administration (NIDA), Bangkok, Thailand

K-singular value decomposition, and sparse coding. Despite its high precision, dictionary feature extraction has a high computation cost. To improve the accuracy without the need for image databases, [19] proposed multi-feature extraction using the three color space histogram, Color Coherence Vector, and the Sobel edge detection. Furthermore, [13] proposed Wavelets and Principal Component Analysis for feature extraction and dimensional reduction, focusing on an image retrieval system with reduced computational complexity. In [10], the data is divided and randomly distributed among various nodes. The query image is distributed to all nodes, and artificial neural network is performed in each node. The local result from each node is combined in the central node for the final result. Although [10] focuses on query speed in the distributed system, it provides less information on image features for similarity matching.

For machine learning solutions, [8] proposed the nearest neighbors of images using text-image relevant information. The result shows that some similarity pairs are matched with no semantic meaning but visual matching. Since the popularity of neural networks, many studies have applied neural networks to image retrieval. [1] proposed a content-based medical image retrieval system using similarity or distance among feature vectors extracted by VGG19 and ResNet50. Deep Ranking model [27] outperformed traditional and deep classification models but there is no comparison with other deep similarity models. For image similarity matching, a multi-scale Siamese network (SimNet) [4] containing two Convolutional Neural Networks (CNNs) can capture both semantic and visual of images. Despite the high accuracy of 92.6%, the fully-connected layers limit SimNet to fixed-length image inputs. To solve this problem, [28] proposed the triplet spatial pyramid pooling network (TSPP-Net). However, there is still a semantic gap between a query image and similar images from the model.

In conclusion, traditional approaches of image similarity searching present computational and storage challenges. Despite advancements in accuracy, the semantic gap issue remains a challenge as mentioned by [8, 28]. In this study, we focus on the specific problem of image similarity searching by images, given the ongoing semantic gap issue. Notably, while prior investigations have primarily evaluated performance based on either accuracy or speed, as in Table 1, our work assesses performance in both domains. Furthermore, we employ the Tensorflow Similarity library to overcome the semantic gap and evaluate similarity accuracy and query time.

3 Proposed methods

As shown in Fig. 1, this study employs two datasets, six experimental models, and two evaluation metrics as described in the following subsections.

Table 1 A summary of the differences between our study and previous studies in terms of evaluation metrics

Work	Evaluation method	
	Accuracy	Speed
Kang et al. [15]	✓	
Roy and Mukherjee [19]	✓	
Durmaz, and Bilge [10]		✓
Chechik et al. [8]	✓	
Wang et al. [27]	✓	
Appalaraju and Chaoji [4]	✓	
Yuan et al. [28]	✓	
Gao et al. [11]	✓	
Portaz et al. [18]	✓	
Chen et al. [9]	✓	
Part and Im [17]	✓	
Agrawal et al. [1]	✓	
Harini and Bhaskari [13]	✓	
Ours	✓	✓

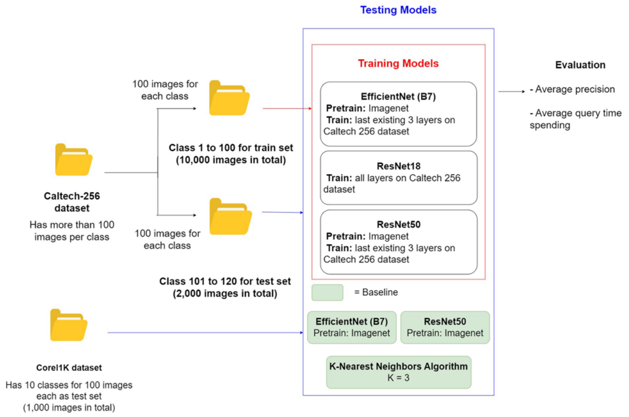

Testing Models
Training Models
Evaluation
- Average precision
- Average query time spending
100 images for each class
Class 1 to 100 for train set (10,000 images in total)
Caltech-256 dataset
Has more than 100 images per class
100 images for each class
Class 101 to 120 for test set (2,000 images in total)
EfficientNet (B7)
Pretrain: Imagenet
Train: last existing 3 layers on Caltech 256 dataset
ResNet18
Train: all layers on Caltech 256 dataset
ResNet50
Pretrain: Imagenet
Train: last existing 3 layers on Caltech 256 dataset
= Baseline
EfficientNet (B7)
Pretrain: Imagenet
ResNet50
Pretrain: Imagenet
K-Nearest Neighbors Algorithm
K = 3
Corel1K dataset
Has 10 classes for 100 images each as test set (1,000 images in total)

Fig. 1 The overview workflow of this study

3.1 Datasets

The Caltech-256 dataset [12] is used to train and test the models, with no overlap in classes between the two sets. Caltech-256 is a benchmark dataset consisting of 30,607 real-world colorful images with 257 classes and at least 80 images per class. For this study, the classes with more than 100 images are selected before we randomly choose only 120 classes for further uses. To limit the number of images to 100 per class, we randomly remove images from each class. Then, the total 120 classes are randomly split to 100 classes for training and 20 classes for testing. The models to be trained with Caltech-256 are EfficientNet (B7), ResNet18, and ResNet50, all pre-trained on ImageNet. Additionally, the Corel1K dataset [6, 16, 23–26] is used for testing the models. To prevent bias in data selection, classes and images from both datasets are randomly chosen. Corel1K is a dataset designed for content-based image retrieval, consisting of 10,800 colorful images (in various sizes) grouped to 80 classes. In this study, 10 classes with more than 100 images are randomly selected; each class is reduced to 100 images for testing the models. This dataset is selected because of its semantic grouping of similar images.

3.2 Models

This study aims to identify the optimal model for image similarity by utilizing the Tensorflow Similarity library [7] version 0.16 as the core library to construct the similarity models. The library was developed by Tensorflow specifically for similarity learning and can be integrated with deep learning model architectures that are well-suited for similarity, such as EfficientNet, ResNet18, and ResNet50. Therefore, we chose to experiment with these three architecture models instead of developing a new architecture from scratch.

Six models are evaluated in this study as shown Fig. 1. Three baseline models include EfficientNet (B7) pre-trained on ImageNet, ResNet50 pre-trained on ImageNet, and K-Nearest Neighbors (KNN). The KNN algorithm with a value of K=3 is used here as it provides the highest accuracy for both Caltech-256's test dataset and the Corel1K Dataset. The other three models are EfficientNet (B7) pre-trained on ImageNet and then finetuned on Caltech-256, ResNet18 trained on Caltech-256, and ResNet50 pre-trained on ImageNet and then finetuned on Caltech-256. The performance of these six models is evaluated on both Caltech-256's test dataset and Corel1K dataset using the average precision score and average query time spent.

To facilitate deep feature learning, we utilized the Circle loss [21] as shown in Eq. 1. The Circle loss function is an improvement over the classification loss function and metric loss function, as described in [21]. This function provides flexibility in the penalty strength applied to within-class similarity and between-class similarity, enabling the application of different weights to various similarity scores.

$$\mathcal{L}_{circle} = log[1 + \sum_{i=1}^{K} \sum_{j=1}^{L} exp(\gamma(\alpha_n^j s_n^j - \alpha_p^i s_p^i))]. \tag{1}$$

3.3 Model training

The study utilized a training batch size of 10 images per batch, with a Circle loss gamma of 256. The model was compiled using Adam optimizer with an initial learning rate of 0.0001. The models were trained for 40 epochs with 1000 steps per epoch and 200 validation steps.

- **EfficientNet (B7)**

In this study, we propose to implement the Tensorflow Similarity model using EfficientNet (B7) [22] as the backbone. EfficientNet models were specifically designed for image classification, as described in [7]. The input image is first fed into EfficientNet (B7), which is pre-trained on ImageNet, with a shape of (600, 600, 3). We freeze all layers except for the last three layers, which are trainable. We then apply Global Max Pooling for feature extraction, and the last layer is the Metric Embedding layer, which produces an embedding output of size 512. The final validation loss is 25.34, while the final training loss is 3.27.

- **ResNet18**

We employed ResNet18 [14], a deep neural network for similarity learning [7], as another underlying architecture. The input images, sized (224,224,3), were processed through the ResNet18 model to extract features using Global Max Pooling, and finally fed into the Metric Embedding layer to produce the 512-dimensional output embeddings. As the Tensorflow Similarity library did not provide a pre-trained ResNet18 model with accessible parameters trained on Imagenet, we opted to train ResNet18 from scratch with randomly initialized weights. To ensure reliable evaluation, we trained the model ten times, each with different initial weights. The average final validation loss and training loss across the ten models were found to be 25.42 and 6.85, respectively.

- **ResNet50**

ResNet50 [14], a well-known deep neural network for similarity learning [7], was chosen as another underlying architecture. We utilized the pre-trained ResNet50 model on ImageNet, with the input images sized (512,512,3). All layers except the last three were frozen, and the remaining layers were made trainable. Global Max Pooling was applied for feature extraction, and the Metric Embedding layer produced the final embeddings with a dimension of 512. A final validation loss is 21.96 whereas a final training loss is 3.76.

4 Experimental results and discussion

The software utilized in this study included Python 3.7, Numpy 1.21.6, Matplotlib 3.5.2, Tensorflow 2.6.4, Tensorflow Similarity 0.16.3, and tabulate 0.8.9. For hardware resources, we used Model 79, Intel(R) Xeon(R) CPU @ 2.20GHz with one CPU core and 12 GB memory. The GPU used was the Tesla P100-PCIE with 16 GB memory.

The study employed the Tensorflow Similarity library as the primary tool for similarity learning. Three models, EfficientNet (B7), ResNet18, and ResNet50, were utilized as the backbone for the study because these models have been shown to perform well in similarity learning tasks [7]. Two datasets, namely Caltech-256's test set and Corel1K, were used for evaluating the models. The Caltech-256's test set comprised 2,000 images, while the Corel1K dataset contained 1,000 images. The indexing component of the Tensorflow Similarity was applied to all test images.

For each test dataset, ten query images were randomly selected to find the top 20 similar images based on the lowest Cosine Similarity distance. Evaluation metrics included the average precision score and average query time spent. For ResNet18, we trained it from scratch ten times with different random weights. The standard deviation (SD) was calculated based on the average precision score and average query time spent for each test dataset. Evaluation metrics included the mean average precision (Eq. 2) and query time spent (Eq. 3), along with the corresponding SD.

$$Average\ precision = \sum \left(\frac{TP}{TP + FP} \right) / N, \tag{2}$$

$$Average\ time\ spending = \frac{\sum Query\ time\ spending}{N}. \tag{3}$$

4.1 Results

EfficientNet (B7) pretrained on ImageNet, with the last 3 layers finetuned on the Caltech-256 dataset, achieved an average precision score of 0.88, which is higher than the average precision score of ResNet50 pretrained on ImageNet, with the last 3 layers finetuned on the Caltech-256 dataset, which achieved a score of 0.69. For ResNet18 trained on the Caltech-256 dataset, the 10 trained-from-scratch models results in 10 different validation loss values.

Overall, EfficientNet (B7) pretrained on ImageNet and fine-tuned on Caltech-256 dataset provides the best results in terms of average precision scores of 0.93 on the Caltech-256's test dataset and 0.94 on the Corel1K dataset. Meanwhile, ResNet18 trained from scratch on Caltech-256 dataset provides the lowest query time spending of 6.76±1.90 s for the Caltech-256's test dataset and 3.85±0.63 s for the Corel1K dataset. Both models can compete with their baseline models as shown in Tables 2, 3, and Fig. 2. According to [2] that applied the traditional method on the Corel1K dataset, it got an average precision score of 0.90 which is lower than ours.

KNN with K = 3 provides the fastest query speed of 0.67 s for both test datasets. However, it suffers from low average precision scores of 0.16 on the Caltech-256's test dataset and 0.00 on the Corel1K dataset, compared to deep learning-based models. Despite its best result in precision, EfficientNet (B7) pretrained on ImageNet and finetuned on Caltech-256 dataset is still not able to improve the query time spending compared to its baseline of Efficient (B7) pretrained on ImageNet.

Table 2 Models evaluation on Caltech-256's test dataset. The bold text is the best (the highest) average precision score

No	Model	Avg precision	Avg time
1	EfficientNet (B7) pretrained on ImageNet (baseline)	0.88	7.12 s
2	ResNet50 pretrained on Imagenet (baseline)	0.80	10.26 s
3	EfficientNet (B7) pretrained on ImageNet, and trained last existing 3 layers from Caltech-256 (ours)	**0.93**	7.08 s
4	ResNet18 trained from Caltech-256 (ours)	0.33 ± 0.01	6.76 ± 1.90 s
5	ResNet50 pretrained on ImageNet, and trained last existing 3 layers from Caltech-256 (ours)	0.69	10.92 s
6	KNN with K = 3	0.16	0.67 s

Table 3 Models evaluation on Corel1K dataset. The bold text is the best (the highest) average precision score and the best (the lowest) average time spending

No	Model	Avg precision	Avg time
1	EfficientNet (B7) pretrained on ImageNet (baseline)	0.92	5.36 s
2	ResNet50 pretrained on Imagenet (baseline)	0.86	7.42 s
3	EfficientNet (B7) pretrained on ImageNet, and trained last existing 3 layers from Caltech-256 (ours)	**0.94**	5.49 s
4	ResNet18 trained from Caltech-256 (ours)	0.46 ± 0.01	**3.85±0.63 s**
5	ResNet50 pretrained on ImageNet, and trained last existing 3 layers from Caltech-256 (ours)	0.77	6.92 s
6	KNN with K = 3	0.00	0.67 s

4.2 Discussion

EfficientNet (B7) pretrained on ImageNet and trained on the last existing 3 layers from Caltech-256 demonstrates the best average precision score in this study. However, it requires a larger input image size of 600×600 pixels, which is larger than the input size required for other models. This model is particularly effective for images with unique patterns, shapes, and colors, as demonstrated by the accurate results obtained for images of dinosaurs, bonsai, and balloons in Fig. 3. Nonetheless, the model is prone to returning similar images of different classes with similar patterns, shapes, or colors as the query image, as shown in Fig. 4. Average query time spending of the deep learning models remains an area of potential improvement. ResNet18 trained on Caltech-256 and tested on Corel1K exhibits the fastest query time among the deep learning models, albeit with the lowest average precision score. In contrast, KNN provides the fastest query time among all models, but with the lowest average precision score compared to the deep learning models. Despite this, the deep learning approach still yields superior accuracy compared to the traditional machine learning method, as evidenced by the results in this study.

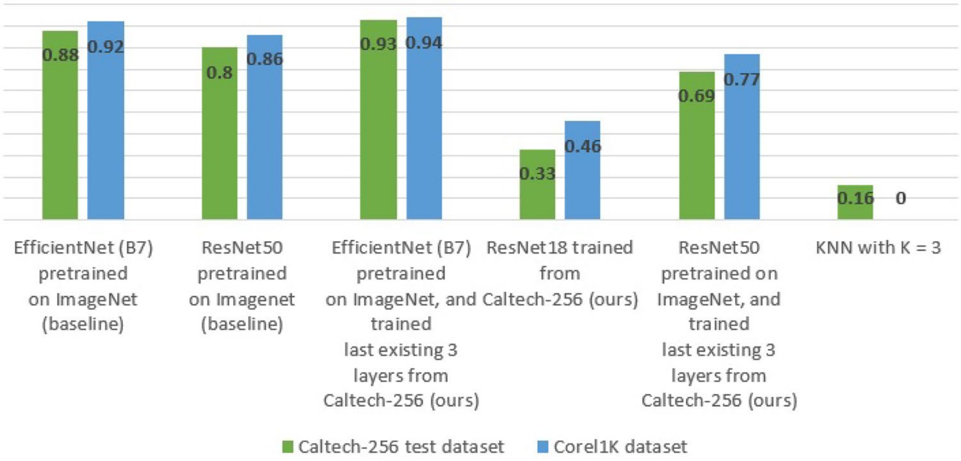
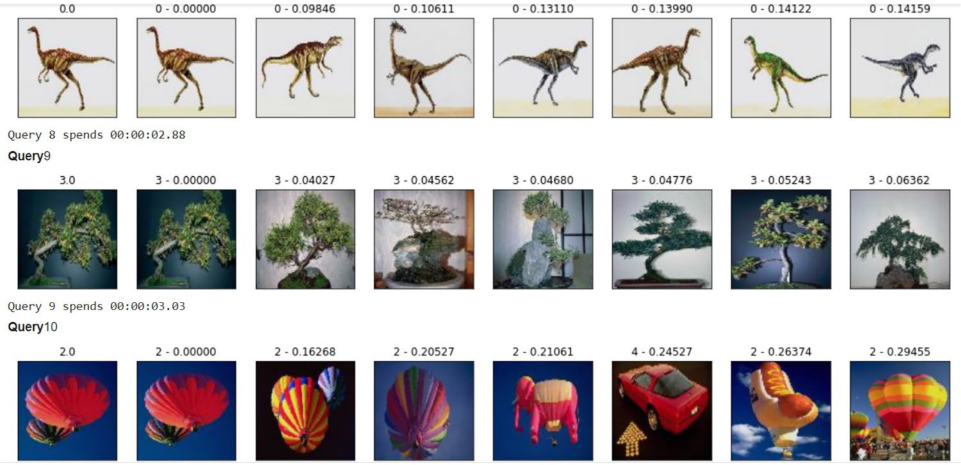


Fig. 2 Comparing the average precision score between models on both Caltech-256 test dataset and Corel1K dataset



```
Query 8 spends 00:00:02.88
Query9
```

```
Query 9 spends 00:00:03.03
Query10
```

Fig. 3 An example of the accurate results of the similarity search algorithm applied in this study. The numbers displayed above each image indicate the distance between the respective image and the query image

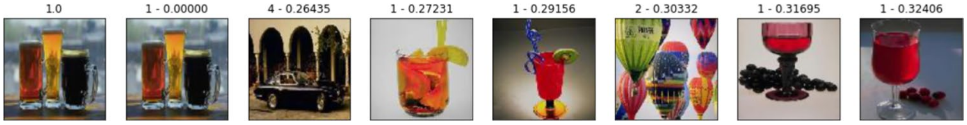


Fig. 4 An example of the inaccurate results of the similarity search algorithm applied in this study. The numbers displayed above each image indicate the distance between the respective image and the query image

5 Conclusion and future works

This study utilized the Tensorflow Similarity library as the core library for identifying the optimal solution for finding similar images of a query image. The Tensorflow Similarity library is plugged in to three models that are efficient for similar searching: EfficientNet, ResNet50, and ResNet18. We conducted experiments on these three backbone models, using the Tensorflow Similarity for similarity calculation. The results indicate that EfficientNet (B7), pre-trained on ImageNet, and finetuned the last three existing layers from Caltech-256, achieved the highest precision scores on both the Caltech-256 test dataset and Corel1K dataset, scoring 0.93 and 0.94, respectively. Among other deep learning models, ResNet18, trained from scratch with Caltech-256, provided the fastest average query time of 3.85±0.63 s. The KNN algorithm exhibited the fastest average query time of 0.67 s but yielded the lowest average precision score.

For future work, there is a need to enhance accuracy of similar image searching while optimize query time spending. This can be achieved by exploring other backbone architectures for identifying similar images, which should accept input sizes smaller than 600×600 pixels to decrease query time. It is also recommended to evaluate the proposed method on the CIFAR-100 dataset because there are 600 images of each class with fine label. So, we can evaluate semantic gap issue using this dataset.

Author contributions All authors contributed equally.

Data availability The Caltech-256 dataset [12] is available in https://authors.library.caltech.edu/7694/ and is derived from https://paperswithcode.com/dataset/caltech-256. The Corel1K dataset [6, 16, 23–26] is derived from https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval.

Declarations

Conflict of interest No conflicts of interest to declare.

References

1. Agrawal S, Chowdhary A, Agarwala S, Mayya V, Kamath S (2022) Content-based medical image retrieval system for lung diseases using deep CNNs. Int J Inf Technol 14(7):3619–3627
2. Ahmed KT, Irtaza A, Iqbal MA (2017) Fusion of local and global features for effective image extraction. Appl Intell 47(2):526–543
3. Ahmad K, Sahu M, Shrivastava M, Rizvi MA, Jain V (2020) An efficient image retrieval tool: query based image management system. Int J Inf Technol 12(1):103–111
4. Appalaraju S, Chaoji V (2017) Image similarity using deep CNN and curriculum learning.arXiv:1709.08761
5. Baliga BS, Medepalli R, Muralikrishna SN (2021) Securing textual and image data on cloud using searchable encryption. Int J Inf Technol 13(3):1111–1117
6. Bian W, Tao D (2010) Biased Discriminant Euclidean Embedding for Content based Image Retrieval. IEEE Trans Image Process 19(2):545–554
7. Bursztein E, Long J, Lin S, Vallis O, Chollet F (2021) TensorFlow Similarity: A Usable, High-Performance Metric Learning Library. Fixme
8. Chechik G, Sharma V, Shalit U, Bengio S (2010) Large Scale Online Learning of Image Similarity Through Ranking. J Mach Learn Res 11(3):1109–1135
9. Chen Y, Gong S, Bazzani L (2020) Image search with text feedback by visiolinguistic attention learning. CVPR:2998–3008
10. Durmaz O, Bilge HS (2019) Fast image similarity search by distributed locality sensitive hashing. Pattern Recognit Lett 128:361–369
11. Gao Y, Wang M, Luan H, Shen J, Yan S, Tao D (2011) Tag-based social image search with visual-text joint hypergraph learning. ACM MM:1517–1520
12. Griffin G, Holub A, Perona P (2022) Caltech 256. CaltechDATA, https://doi.org/10.22002/D1.20087
13. Harini DND, Bhaskari DL (2012) Image retrieval system based on feature extraction and relevance feedback. CUBE:69–73
14. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. CVPR:770–778
15. Kang L, Hsu C, Chen H, Lu C, Lin C,Pei S (2011) Feature-based sparse representation for image similarity assessment. IEEE Trans Multimed. 13(5):1019–1030
16. Li J, Allinson N, Tao D, Li X (2006) Multitraining Support Vector Machine for Image Retrieval. IEEE Trans Image Process 15(11):3597–3601
17. Park G, Im W (2016) Image-text multi-modal representation learning by adversarial backpropagation. arXiv:1612.08354
18. Portaz M, Randrianarivo H, Nivaggioli A, Maudet E, Servan C, Peyronnet S (2019) Image search using multilingual texts: a cross-modal learning approach between image and text. arXiv:1903.11299
19. Roy K, Mukherjee J (2013) Image similarity measure using color histogram, color coherence vector, and sobel method. IJSR 2(1):538–543
20. Sachar S, Kumar A (2022) Deep ensemble learning for automatic medicinal leaf identification. Int J Inf Technol 14(6):3089–3097
21. Sun Y, Cheng C, Zhang Y, Zhang C, Zheng L, Wang Z, Wei Y (2020) Circle loss: a unified perspective of pair similarity optimization. CVPR:6397–6406
22. Tan M, Le QV (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML:6105–6114
23. Tao D, Tang X, Li X, Rui Y (2006) Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans Multimed 8(4):716–727
24. Tao D, Tang X, Li X, Wu X (2006) Asymmetric Bagging and Random Subspace for Support Vector Machines-based Relevance Feedback in Image Retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099
25. Tao D, Li X, Maybank SJ (2007) Negative Samples Analysis in Relevance Feedback. IEEE Trans Knowl Data Eng 19(4):568–580
26. Wang JZ, Li J, Wiederhold G (2001) SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. IEEE Trans Pattern Anal Mach Intell 23(9):947–963
27. Wang J, Song Y, Leung T, Rosenberg C, Wang J, Philbin J, Chen B, Wu Y (2014) Learning fine-grained image similarity with deep ranking. CVPR:1386–1393
28. Yuan X, Liu Q, Long J, Hu L, Wang Y (2019) Deep image similarity measurement based on the improved triplet network with spatial pyramid pooling. Information 10(4):1–17