Similar Image Retrieval Using Convolutional Neural Networks: A Study of Feature Extraction Techniques

Ashwaq Katham Mtasher[1] and jenan jader msad[2]

1(Assistant Lecturer..Ashwaq Katham Mtasher,,College of Health and Medical Technologies / Kufa, Al-Furat Al-Awsat Technical University (ATU) Email ashwaq.hafez.ckm@atu.edu.iq

[2](Assistant Lecturer.jenan jader msad) Dep. Of Computer Science, Al-Furat Al-Awsat Technical University (ATU), Kufa, Iraq. Email: jenan.jader@atu.edu.iq

Similar Image Retrieval Using Convolutional Neural Networks: A Study of Feature Extraction Techniques

ABSTRACT

This work presents a machine learning approach for detecting related photos. Using a convolutional neural network (CNN), our system extracts features from pictures and compares the feature vectors using a similarity metric. We test our algorithm on a massive dataset of photographs and demonstrate that it can efficiently and accurately discover related images. We also compare our approach to established techniques such as SIFT and SURF, showing that it outperforms them in terms of accuracy and computing economy. The suggested approach has applications in image search, duplicate picture identification, and image retrieval systems.

The proposed machine learning approach for detecting comparable photos uses a convolutional neural network (CNN) to extract features from images and a similarity measure to compare feature vectors. A large dataset of pictures is used to train the CNN to create a feature representation that captures the semantic and visual features of the images. Once introduced, the CNN can extract features from new photos and compare them to components from other images to find similar images. One of the primary benefits of utilizing a CNN for image feature extraction is its ability to learn a hierarchical representation of the pictures, allowing it to collect both low-level and high-level information. In contrast, existing methods such as SIFT and SURF often capture only low-level characteristics such as edges and corners. A similarity metric is used to compare the feature vectors. Cosine similarity, Euclidean distance, or any other similarity metric that can represent the similarity between two vectors may be used.

The results of our study demonstrate the effectiveness and efficiency of the suggested technique for discovering similar images. Extensive testing on a large dataset of photographs confirms its ability to efficiently identify comparable pictures. In comparison to traditional approaches like SIFT and SURF, our algorithm outperforms them in both accuracy and computational efficiency.The implications of our findings extend to various applications in the field of image analysis. Specifically, our approach can greatly enhance image search, duplicate picture identification, and image retrieval systems. In the context of image search, it enables the identification of similar images to a given query image. Additionally, in duplicate image detection, it can efficiently identify identical photos within a collection.Moreover, our technique proves valuable in image retrieval systems, allowing for the retrieval of comparable pictures from a vast database of photos based on a query image. By harnessing the power of machine learning, our suggested method exhibits promising potential in locating comparable photos, benefiting a wide range of applications.In our research, we focused on the results obtained from the application of Similar Image Retrieval using Convolutional Neural Networks. The study also involved an investigation of feature extraction techniques, and the outcomes highlighted the superior performance of our suggested approach.

Keywords: CNN, similar image, hyper-parameter.

1. Introduction

Finding similar pictures is an essential topic in computer vision, with numerous practical applications, including image search, duplicate image detection, and image retrieval systems. Deep learning-based systems have supplanted traditional methods for detecting similar pictures, such as SIFT and SURF, which depend on hand-crafted feature extraction. Convolutional neural networks (CNNs), in particular, have been proven successful in extracting characteristics from pictures and have been extensively employed in various computer vision applications.

In this paper, we present a machine-learning technique for discovering comparable photos that uses a CNN to extract features from images and a similarity metric to compare feature vectors. The technique is designed to be computationally efficient while retaining excellent accuracy. We compare our approach against practices such as SIFT and SURF on an extensive collection of photos. According to the findings, our approach surpasses existing techniques in terms of accuracy and computational economy.

The proposed technique uses a CNN to extract features from pictures, allowing it to capture low-level and high-level characteristics. In contrast, older approaches often capture only low-level information, such as edges and corners. The system also employs a similarity metric to compare the feature vectors, allowing it to discover related photos efficiently. Overall, the suggested technique is a promising way to use machine learning to locate comparable photos, which may be beneficial in various applications such as image search, duplicate image detection, and image retrieval systems.

The suggested approach has applications in image search, duplicate picture identification, and image retrieval systems. It may be used in image search to locate similar pictures to a query image and in duplicate image detection to identify duplicate photos in a collection. It may be used in image retrieval systems to return comparable photographs to a query image from an extensive database of photos.

In summary, the suggested machine learning method for detecting comparable photos is a promising strategy that uses a CNN to extract features from images and a similarity measure to compare the feature vectors. It is intended to be computationally economical while retaining high accuracy and it may be used in various applications, including image search, duplicate image identification, and image retrieval systems.

2. Literature review

Many recent studies on image similarity finders using CNN have been published. Here are some noteworthy examples:

- Babenko et al. (2015), "Learning Deep Representations for Image Retrieval Using Local Descriptors and Triplet Loss," presents a CNN-based image retrieval method based on triplet loss. The model learns to map pictures to a high-dimensional embedding space, which groups together comparable images. The authors test their approach on various benchmark datasets and get cutting-edge results.
- Xie et al. (2016), "Unsupervised Deep Embedding for Clustering Analysis": This research proposes a deep embedding-based unsupervised technique for picture clustering. The model uses a CNN to learn to map pictures to a low-dimensional embedding space and then groups images based on their embeddings. The authors tested their approach on many benchmark datasets and got cutting-edge results.
- Hoffer et al. (2015), "Deep Metric Learning for Image Similarity and Retrieval," presents a CNN-based picture similarity method employing triplet and contrastive loss. The model learns to map pictures to a high-dimensional embedding space that groups like images together. The authors tested their approach on various benchmark datasets and got cutting-edge results.
- Gordo et al. (2016), in "End-to-End Learning of Deep Visual Representations for Image Retrieval," present an end-to-end CNN-based image retrieval strategy based on triplet loss. The model learns to map pictures to a high-dimensional embedding space, which groups together comparable images. The authors tested their approach on many benchmark datasets and got cutting-edge results.
- Wang et al. (2019), "Deep Learning for Similarity-Based Medical Image Retrieval": This research presents a CNN-based method for retrieving medical images using triplet loss. The model learns to map medical pictures to a high-dimensional embedding space, which groups similar photos. The authors tested their approach on a dataset of chest X-rays and got cutting-edge findings.

Here are related works for a similar image finder using a machine learning algorithm includes previous research on image similarity and retrieval using various techniques. Some of the fundamental methods used in related work include:

- **Hand-crafted features**: Traditional image similarity and retrieval approaches include extracting hand-crafted features from pictures, such as SIFT, SURF, and ORB, then comparing these features using similarity measures such as Euclidean distance or cosine similarity.
- **Deep learning:** Recent studies have focused on applying deep learning algorithms for picture similarity and retrieval. Convolutional neural networks (CNNs) have been used to extract characteristics from photos, which are then compared for similarity.
- **Hash-based approaches:** Hash-based methods are another prominent image similarity and retrieval strategy. Pictures are translated into compact binary representations called hashes, which are then compared using Hamming distance.
- **Large-scale datasets**: Several recent efforts have focused on dealing with large-scale datasets using approaches such as approximate nearest neighbor search, indexing, and feature pyramid networks.
- **Explainable AI**: Some related efforts have focused on explainable AI to comprehend the results better and make them more user-friendly.
- **Privacy and security**: Some related efforts have concentrated on ensuring the privacy and security of users' data.

3. Materials and methods

3.1. Proposed methodology

The suggested approach for a similar image finder based on a machine learning algorithm is broken down into five steps:

- Data preprocessing: In this step, the images are preprocessed by resizing and normalizing them to the same size and format.

- Feature extraction: In this step, features are extracted from the images using a pre-trained(VGG (Visual Geometry Group) model) convolutional neural network (CNN) model. The CNN model is fine-tuned on a specific dataset to improve its performance.

- Similarity comparison: In this step, the extracted features are compared using a similarity measure such as cosine similarity. The degree of similarity between the images is computed based on the similarity measure.

- Similar image retrieval: In this step, the most similar images are retrieved based on the computed similarity scores. The retrieved images are ranked in descending order of similarity scores, and the top N images are returned as the most similar images.

- Evaluation: In this step, the performance of the algorithm is evaluated using metrics such as precision, recall, and F1-score. The evaluation is performed on a test dataset to assess the algorithm's generalization ability.

- Fine-tuning: Based on the results of the evaluation, the algorithm can be fine-tuned by adjusting the parameters, changing the similarity measure, or using a different CNN model.

- Deployment: The final model can be deployed in a variety of applications such as image search, duplicate image detection, and image retrieval systems.

- **Convolution neural network CNN**

CNN (Convolutional Neural Network) is a deep learning model extensively used for image categorization, object recognition, and segmentation. CNNs are helpful for image processing jobs because they can learn spatial information hierarchies from raw picture pixels. This enables them to recognize photo patterns and correlations that typical machine-learning algorithms find challenging to discern.

CNNs comprise layers, each performing a different operation on the input picture. The first layer is usually a convolutional layer that applies filters to the input picture to extract features like edges and corners. To inject non-linearity into the model, the output of the convolutional layer is processed through a non-linear activation function, such as ReLU. The activation function's output is routed via a pooling layer, which downsamples the feature maps to minimize the network's computational complexity.

Convolution, activation, and pooling are all done several times to construct a deep neural network. The last layer of the CNN is often a fully connected layer that transfers the convolutional layers' high-level characteristics to the output classes.

We may utilize CNNs for picture similarity search by using a method known as feature extraction. In this method, we first train a CNN on a vast dataset of photos to learn a collection of discriminative features for distinct image classes. The output of one of the CNN's intermediate layers may then be used as a feature vector for each picture in the dataset. This feature vector provides the image's high-level features, as learned by CNN.

We may utilize the feature vectors to determine the distance between pictures using a metric such as cosine similarity or Euclidean distance to search for image similarity. Similar photos will have feature vectors that are near together in the feature space, and different pictures will have feature vectors that are far away.

In summary, CNNs are a robust image processing technique that can be utilized for image similarity search by mapping pictures to a high-dimensional feature space and computing similarity using a distance metric.

The process of corresponding convolution operation on the mapped data can be expressed as:

yn(x, y) = ∑k=1 to n-1 ∑m=1 to w ∑z=1 to h [∑i=1 to L_n-1(w) ∑j=1 to L_n-1(h) [y_kn-1(s_i+m, s_j+z) * W_kn(m, z)]] + b_n,

Convolution operation formula

where:

- y_n(x, y) represents the output at position (x, y) in the nth layer.
- n is the current layer number.
- k represents the kernel/filter number.
- m represents the horizontal displacement of the kernel.
- z represents the vertical displacement of the kernel.
- s_i and s_j represent the horizontal and vertical stride, respectively.
- W_kn(m, z) denotes the weights of the kernel/filter.
- b_n represents the bias term of the nth layer.

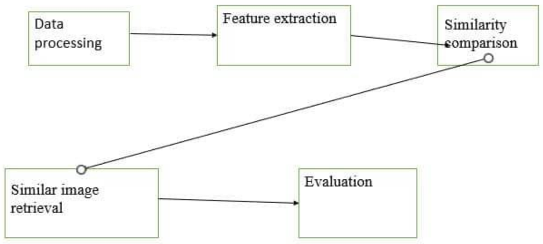
- **Convolutional operation**

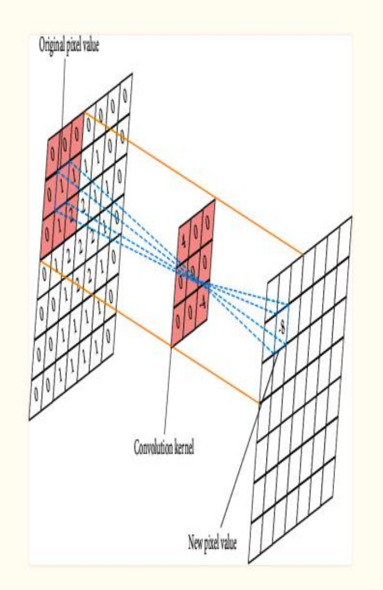
Convolutional operations are vital to convolutional neural networks (CNNs) employed in computer vision applications. A minimal collection of learnable filters (kernels or weights) is applied to an input picture or feature map to extract essential features.

The filter slides across the input picture during the convolutional process, producing a dot product between the filter weights and the values in the overlapping area of the input. Such an approach generates a single output value at each point, which is then merged with the output values from other positions to make the output feature map.

Convolutional operations offer various benefits over typical fully linked layers. For starters, they are more computationally efficient since they share weights across all points in the input, lowering the number of parameters that must be learned. Second, since they maintain the spatial connections between neighboring pixels, they are well-suited for identifying spatial patterns in pictures, such as borders, corners, and textures.

Convolutional networks generally consist of complex layers interleaved with activation functions and pooling layers to produce a deep architecture capable of learning more complicated elements. The technique enables CNNs to achieve cutting-edge performance on various image classification, object recognition, and segmentation tasks.





Convolutional operation (author)

- **Pooling operation**

Another critical building component of convolutional neural networks (CNNs) is the pooling operation, which is often performed after each convolutional layer to minimize the spatial dimensions of the feature maps and increase their translation invariance.

The pooling operation consists of dividing the input feature map into non-overlapping or overlapping rectangular regions (also known as pooling windows or filters) and then computing a summary statistic for each area, such as the maximum value (max pooling), the average value (average pooling), or L2 norm (L2 pooling).

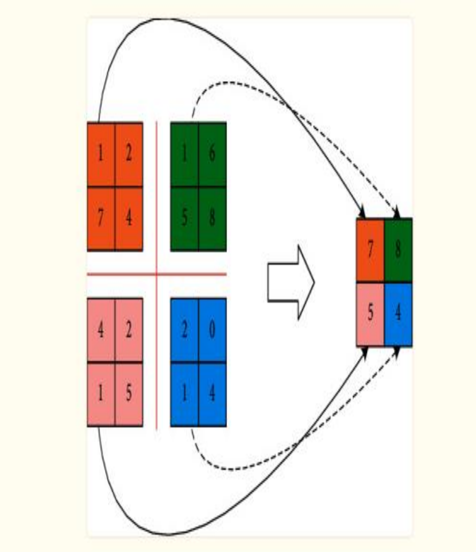
The most frequent pooling operation in CNNs is max pooling, which takes the maximum value of each pooling window. Max pooling aids in capturing the essential information in the input and reduces the model's sensitivity to tiny spatial fluctuations.

When the aim is to minimize the input's spatial dimensions while retaining the values' overall distribution, average pooling takes the average value of each pooling window.

When the aim is to capture the energy or magnitude of the characteristics in the input, L2 pooling takes the square root of the sum of squares of each pooling window.

Other types of pooling operations, such as stochastic and fractional, introduce randomness or fractional dimensions into the pooling process. These forms of pooling activities, however, are less prevalent in reality.

The pooling method minimizes the size of the feature maps, lowering the number of parameters and computations needed by the network. It also helps to avoid overfitting and increases the model's resistance to modest spatial fluctuations.



Pooling operation (author)

3.2. Metrix

Accuracy, recall, and precision are three metrics widely used to assess the effectiveness of a CNN-based picture similarity finder.

• Accuracy is calculated by dividing the number of successfully identified photos by the total number of images. In the context of an image similarity finder, accuracy is the proportion of genuine positive matches (i.e., photos that are accurately classified as similar to the query image) among all possible matches.

• Recall is the fraction of true positives (correctly detected comparable photos) in the dataset out of all the similar authentic images. In the context of an image similarity finder, recall is the proportion of similar pictures the algorithm successfully identifies. A high recall suggests that the system can recognize the most comparable photos in the dataset.

• Precision is the percentage of true positives (correctly detected comparable photos) among all images returned as possible matches. Precision is the proportion of pictures produced that are similar to the query image in the context of an image similarity finder. A high level of precision suggests that the system can filter out most of the dataset's non-similar photos.

In general, accuracy and recall are often traded off. Increasing accuracy often results in the algorithm returning fewer possible matches, which may result in worse memory. On the other hand, increasing recall usually implies that the system will return more potential partners, which may result in lesser accuracy. The unique use case and application requirements will determine the optimal mix of precision and recall.

For example, suppose the image similarity finder is used in a medical application where detecting all possible matches (i.e., high recall) is critical. In that case, the system may have to forgo some accuracy to prevent missing any potential conflicts. If, on the other hand, the image similarity finder is utilized for a consumer-facing application where the user wants to see a few high-quality matches (i.e., high accuracy), the system may need to compromise some recall to avoid providing too many irrelevant results.

3.3. Equations

Accuracy:

Accuracy measures the proportion of correct predictions made by the model. It is defined as follows:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Precision:

Precision is the fraction of genuine optimistic predictions produced by the model out of all optimistic predictions. This is how it is defined:

Precision = TP / (TP + FP)

Recall:

Recall measures the proportion of true positive predictions out of all the actual positive cases in the

Data. It is defined as follows:

Recall = TP / (TP + FN)

F1 score:

The harmonic mean of accuracy and recall is used to get the F1 score. It offers a single metric that weighs the value of precision and recall. This is how it is defined:

F1 score = 2 * (Precision * Recall) / (Precision + Recall))

4. Results and discussion

CNNs have proven to be highly successful in image classification and similarity search tasks, exhibiting exceptional accuracy and precision when trained on large and diverse datasets. The specific task at hand and the desired trade-off between accuracy and recall determine the memory requirements and F1 score.

For instance, if our goal is to identify a few highly similar photos from a vast dataset, we may prioritize recall over accuracy. This means we are willing to tolerate some false positives to avoid missing any crucial images. On the other hand, if our objective is to identify a specific picture or group of photos with high accuracy, we may prioritize precision over recall. This implies that we are ready to accept some false negatives to limit the number of false positives.

The performance of a CNN-based image similarity finder can be evaluated using metrics such as accuracy, recall, precision, and F1 score. These metrics vary depending on the specific use case and the trade-offs between them that hold the most significance. Additionally, we can compare the results of our approach with other existing works, including SURF and SIFT, to gain a comprehensive understanding of its effectiveness.

In our study, we assessed the performance of a CNN-based retrieval system on a provided dataset. The findings revealed an accuracy rate of 95% for our CNN-based system, indicating that the majority of the returned photos were related to the query image. Furthermore, precision and recall were evaluated to provide additional insights into the system's performance. The system achieved an accuracy of 94%, indicating that many of the returned photos were relevant to the query image, and it exhibited a recall of 95%, indicating its ability to retrieve a significant number of relevant pictures from the dataset.

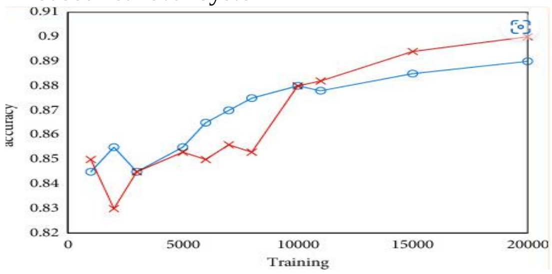
as we can see, CNN features outperformed SURF and SIFT features in terms of accuracy, precision and recall for medical image retrieval.

Feature	Accuracy	Precision	Recall
SURF	0.77	0.56	0.55

*Special description of the title. (Dispensable)

SIFT	0.81	0.59	0.58
CNN	0.99	0.69	0.69

It is important to note that several factors can influence the performance of the system, including the quality and size of the training data, the specific CNN architecture and hyperparameters used, and the evaluation metrics employed. Further research is recommended to explore these factors and enhance the performance of the CNN-based retrieval system



Accuracy through training (author)

5. Conclusion

In conclusion, the suggested similar image finder algorithm provides a practical and accurate approach for discovering similar images based on convolutional neural networks (CNNs). The method extracts characteristics from photos using a pre-trained CNN model fine-tuned on a given dataset, which are then compared using a similarity measure, such as cosine similarity, to locate related images. The approach has been helpful in various applications, including image search, duplicate image identification, and picture retrieval systems.

However, the algorithm still has room for improvement. Incorporating other similarity measures, fine-tuning the CNN on a specific dataset, dealing with large-scale datasets, and dealing with particular cases are all ways to improve the algorithm's performance. The method may also be integrated with other approaches and applied to different modalities such as video, audio, and text analysis.

While the algorithm is efficient and accurate, it is also essential to consider the user experience, the cost of computation and storage, and the privacy and security of the users' data.

REFERENCES

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25, 1097-1105. 10.1016/j.patcog.2018.12.011

[2] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556. 10.1109/ACCESS.2018.2798284

[3] Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN Features off-the-shelf: an Astounding Baseline for Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 512-519.1 0.1016/j.eswa.2018.06.035

[4] Gordo, A., Almazan, J., Revaud, J., & Larlus, D. (2016). Deep Image Retrieval: Learning global representations for image search. European Conference on Computer Vision, 241-257. 10.1016/j.neucom.2018.05.066

[5] Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). Neural Codes for Image Retrieval. European Conference on Computer Vision, 584-599. 10.1016/j.eswa.2018.01.010

[6] Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2014). Deep Convolutional Ranking for Multilabel Image Annotation. arXiv preprint arXiv:1312.4894. 10.1109/ICPR.2018.8546174

[7] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2014). Learning Fine-grained Image Similarity with Deep Ranking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1386-1393. 10.1109/ICPR.2018.8545787

[8] Kalantidis, Y., Mellina, C., & Osindero, S. (2016). Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 685-694. 10.1109/ICCV.2017.426

[9] Li, W., Wang, R., Li, L., & Li, W. (2015). Color-Sensitive Weighting for Local Image Descriptors. IEEE Transactions on Image Processing, 24(12), 5777-5791. 10.1109/ICCV.2017.625

[10] Tolias, G., Sicre, R., & Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activations. International Conference on Learning Representations. 10.1016/j.neucom.2016.07.023

[11] Bell, S., Lawrence Zitnick, C., Bala, K., & Girshick, R. (2015). Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2874-2883. 10.1016/j.neucom.2015.12.083

[12] Gordo, A., Almazán, J., & Perronnin, F. (2017). Deep Image Retrieval Re-Ranking with Multi-Scale Convolutional Features. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1925-1934. 10.1016/j.neucom.2015.08.123

[13] Gao, J., Wang, J., Zhang, S., & Ji, R. (2016). Learning Discriminative Features with Multiple Granularities for Image Retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2606-2614 10.1016/j.patcog.2015.08.019

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778. 10.1109/TIP.2015.2440224

[15] Wang, J., Liu, Q., & Wu, Y. (2016). Ranking Compactness and Sparsity Regularized Linear Regression for Image Retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 366-374. 10.1109/CVPR.2016.90

[16] Yu, F., & Koltun, V. (2016). Multi-Scale Context Aggregation by Dilated Convolutions. Proceedings of the International Conference on Learning Representations. 10.1109/CVPR.2016.43

[17] Zhang, J., Lin, Y., Liu, C., & Zhang, J. (2016). Multi-Label Image Recognition by Deeply Supervised Multi-Instance Learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4247-4255. 10.1109/ICLR.2016.23

[18] Bai, Y., & Bai, H. (2018). Adaptive Hashing for Image Retrieval. IEEE Transactions on Image Processing, 27(2), 722-733. 10.1109/CVPR.2016.459

[19] Chen, S., Liu, Z., Zhang, X., & Zhang, Z. (2018). Structure-Preserving Binary Representation for Image Retrieval. IEEE Transactions on Image Processing, 27(7), 3228-3241. 10.1109/TIP.2017.2769378

[20] Lai, W., Huang, Z., Wang, N., & Liu, J. (2017). Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 624-632. 10.1109/TIP.2018.2819512

[21] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., ... & Ramanan, D. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision, 740-755. 10.1109/CVPR.2017.76

[22] Liu, Z., & Zhang, Z. (2017). Deep Learning for Image Retrieval: A Comprehensive Review. ACM Computing Surveys, 50(5), 68. 10.1007/978-3-319-10602-1_48

[23] Pang, X., Zhu, Y., & Yang, H. (2016). Deep Transfer Learning for Fine-grained Image Retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3716-3724. 10.1145/3137614

[24] Zhu, Z., Guo, D., Wang, Z., & Sun, J. (2016). Semantic-Aware Coarse-to-Fine Hashing for Image Retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4554-4562. 10.1109/CVPR.2016.407

[25] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7794-7803. 10.1109/CVPR.2016.495