Attention-Aware Invertible Hashing Network

Shanshan Li[1], Qiang Cai[1], Zhuangzi Li[1(✉)], Haisheng Li[1], Naiguang Zhang[2], and Jian Cao[1]

[1] School of Computer and Information Engineering, Beijing Technology and Business University, Beijing, China
shanshanli233@126.com, caiq@th.btbu.edu.cn, lizhuangzii@163.com
{lihsh,caojian}@th.btbu.edu.cn

[2] Information Technology Institute, Academy of Broadcasting Science, NRTA, Beijing, China
zhangnaiguang@abs.ac.cn

Abstract. In large-scale image retrieval tasks, hashing methods based on deep convolutional neural networks (CNNs) play an important role due to elaborate semantic feature representation. However, they usually progressively discard information during feature transformation, thus leading to incomplete and unsatisfactory hashing codes for image retrieval. This study tries to design an invertible architecture to maintain image information, meanwhile focus on necessary parts of image features. Consequently, in this paper, we propose a novel attention-aware invertible hashing network (AIHN) for image retrieval. By invertible feature representations, the final hash codes can be completely obtained from input images without any information loss. For highlighting informative regions, we present a novel attention-aware invertible block as the basic module of AIHN, which can promote generalization ability by spatial attention mechanism. Extensive experiments conducted on benchmark datasets demonstrate the effectiveness of our invertible feature representation on hash code generation, and show the promising performance on image retrieval of our methods against the state-of-the-arts.

Keywords: Image retrieval · Deep hashing · Attention mechanism

1 Introduction

With the explosive growth of data in practical applications such as image retrieval, approximate nearest neighbor (ANN) search has become a hot topic in recent years. In the existing ANN technology, hashing method has become one of the most popular and effective technologies because of its fast query speed and low memory cost. Amounts of studies have shown that hashing has improved

S. Li, Q. Cai and Z. Li—These authors contributed equally to this paper and share the first authorship.

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11903, pp. 409–420, 2019.
https://doi.org/10.1007/978-3-030-34113-8_34

the performance on image retrieval tasks [7,23]. However, these methods are defective in feature representation and can not be trained end-to-end.

Recently, convolutional neural networks (CNNs) are gradually applied to image hashing retrieval, and have achieved promising performance. Xia et al. [22] firstly adopt the CNN architecture in the hash algorithm. Later, series of deep hashing methods based on CNN [16,17] are proposed in an end-to-end manner, showing the effectiveness of deep feature representation. The performance of these deep learning hash methods has been greatly improved compared with the traditional hash method in many benchmarks. Moreover, it proves crucial to jointly learn similarity-preserving representations and control quantization error of converting continuous representation into binary codes [3]. However, existing deep feature representation are generated with gradually discarding image information. It may result in discarding of representative feature variability in the process of feature transformation, which can not guarantee obtaining complete image information. In addition, informative regions of image are not highlighted well in existing algorithm, causing poor generalization ability.

To effectively solve the above-mentioned problems, we propose a novel image retrieval framework based on invertible network with spatial attention mechanism. Firstly, a reversible network is proposed, which guarantee the lossless representative features transformed from original image. In such a way, all the information of the image will be forwarded through the network. Then, we adopt spatial attention architecture to tell where to focus, which also improves the representation of interests. Spatial attention effectively learns which information to emphasize or suppress in the process of information transmission. As shown in Fig. 1, our method yield most of state-of-the-art retrieval performance. To summarize, the main contributions of this paper are three-fold:

- We propose an effective invertible network with lossless image information for image retrieval, where the whole framework can be trained end-to-end;
- To excavate informative regions of features, we adopt spatial attention module in our invertible block to learn how to focus on objective information and suppress unnecessary ones.
- Extensive experiments on benchmark datasets show that our architecture is effective and achieves promising performance.

The rest of the paper is organized as follows: in Sect. 2, we introduce some related work about our algorithm. The proposed method is illustrated in Sect. 3, followed by the experimental results in Sect. 4. In Sect. 5, we conclude our work.

2 Related Work

2.1 Hashing Methods

Existing hashing methods [1,25] can be roughly divided into two categories, namely unsupervised hashing and supervised hashing. Unsupervised hashing exploit unlabeled data to learn a set of functions, which encode data to binary

codes [5,21]. Locality-Sensitive Hashing (LSH) [5] is the most representative unsupervised hashing algorithm, achieving promising performance compared with previous approaches. LSH guarantees similar data points preserve similar binary codes after the same hash mapping, vice versa. Supervised hashing [18,20] further exploit label information during learning to generate compact hash code. Supervised Hashing with Kernels (KSH) utilizes the pair-wised labels to generate effective hash functions, which guarantees minimizing the Hamming distances for similar pair-wise data and meanwhile maximizing the dissimilar ones.

In recent years, CNN have shown significant success in computer vision [13–15,19,26–31,34–37]. In the domain of hashing retrieval, [22] was the first deep neural network, achieving promising performance compared with conventional approaches. Deep Hashing Network (DHN) [33] not only preserves pairwise similarity but also controls the quantization error. For improving DHN, HashNet balances training data consisting of positive pairs and negative pairs, and reduces quantization error by continuation technology, thus gaining the most advanced performance on several benchmark datasets. But the high-dimensional features obtained in these methods are accompanied with gradual loss of image information, and we can not ensure whether the discarded information variability is significant.

2.2 Attention Mechanism

The attention mechanism can be viewed as a strategy to bias the allocation of available processing resources towards the most informative components of an input [10]. Attention module has been widely applied in the Natural Language Processing (NLP) field like machine translation, sentence generation etc. And these performance is surprisingly remarkable. Meanwhile, in the image vision field, attention mechanism also demonstrates powerful capabilities. For example, Hu et al. [9] utilize attention to propose an object relation module, which models the relationship among a set of objects and improves object recognition. In this work [24], a self-attention module is introduced in order to better generate images. A channel-wise attention was proposed for image super-resolution task [32]. In our work, the attention-aware invertible hashing network aims at utilizing spatial attention to enhance informative features from the spatial domain, which can accurately tell which information to emphasize or suppress.

3 Our Method

3.1 Overview

The architecture of AIHN is shown in Fig. 1. The pair-wise images are firstly fed into an invertible downsample layer to increase the number of output channels, while decreasing the spatial resolution. Then, the output is split into two sublayers (x_1, y_1) of equal channel dimension. Next, sublayers (x_1, y_1) are put into the invertible block. It is worth noting that spatial attention and invertible

downsampling module are introduced in the invertible block. Spatial attention module is to notice the most informative components of an input, and invertible downsampling module is adopted to reduce the number of computations while maintaining good performance. More details about these two will be introduced in Sects. 3.2 and 3.3 below. After totally 100 similar blocks, invertible high-dimensional features are obtained through followed concatenation operation. The invertible features are send to average pooling and linear layer after a ReLU non-linearity. The results are quantized by Sgn function to get pair-wise binary hash codes. The pairwise similarity loss is adopted for similarity-preserving learning in the Hamming space, and quantization loss is to control both the binarization error and the hash code quality. The invertible downsampling, spatial attention module, as well as invertible block will be introduced in next sections in detail.

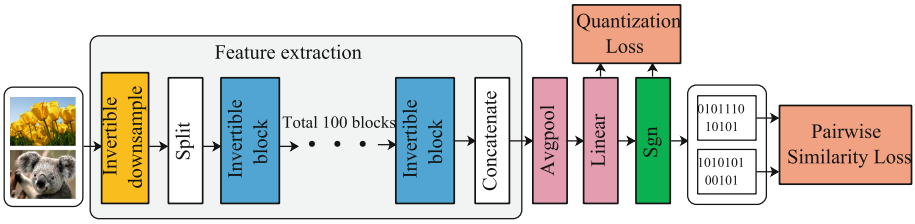

Fig. 1. The framework of the proposed invertible spatial-attention hashing network.

3.2 Invertible Downsampling

In order to facilitate calculation and avoid the use of irreversible module at the same time, we introduce invertible downsampling module to our architecture instead of Maxpooling used in [6]. It not only reduce the spatial resolution of the input for the sake of simplicity but also potentially increase the number of channel for lossless information. As shown in Fig. 2, downsampled by a factor of θ 4, the output's channel is 4 times the original, and the size of each feature map is reduced by 4 times. And also invertible downsampling preserves roughly the spatial ordering, thus avoiding mixing different neighborhoods via the next convolution. Invertible downsampling operation can be written as below:

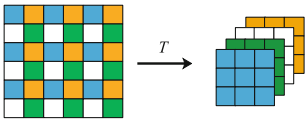

Fig. 2. The illustration of invertible downsampling.

$$T(\theta, Fe(c, w, h)) = Fe(\theta \times c, w/(\theta/2), h/(\theta/2)) \tag{1}$$

where θ represents scaling factor which determines the downsampled size directly, T is the function of downsampling operation, and $Fe(c, w, h)$ denotes feature maps with channel c, width w, and height h.

For reducing computational costs, invertible downsampling is designed tightly for our architecture. It will correspond to an invertible downsampling operator respectively at the begin of our network and depth $d = 6, 22, 94$.

3.3 Invertible Block

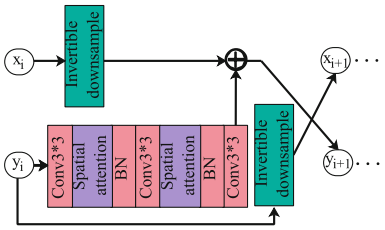

Fig. 3. The structure of invertible block.

The invertible block is an important component for our invertible hashing network. It not only determines the reversibility of information flow, but also generates attentioned features with lossless information. Spatial attention module and invertible downsampling module introduced in Sects. 3.2 and 3.4 are adopted in the invertible block. In particular, spatial attention module mining the objective information and invertible downsampling module allows us to reduce the number of computations while maintaining good performance. The details of the invertible block are illustrated as Fig. 4.

Detailedly, sublayers (x_i, y_i) obtained through splitting operation are doing different two operations. x_i is feed to a invertible downsampling layer with scaling factor θ 4 directly, so we can get $T(4, x_i)$. y_i is sent to a bottleneck block F, mainly consisting of a succession of 3 convolutional operators. The second convolutional layer has four times fewer channels than the other two, while their corresponding filter sizes are respectively 1×1, 3×3, 1×1. The first and the second are preceded by spatial attention module, Batch normalization (BN) and ReLU non-linearity. What needs to be emphasized is that the last convolution layer are followed by batch normalization and ReLU non-linearity only. Obtained $F(y_i)$ plus $T(4, x_i)$, then Y_{i+1} is got. Meanwhile, y_i is also feed to an invertible downsampling layer for convenient calculation, and x_{i+1} is equal to output $T(4, y_i)$. In summary, the detailed operation is described as below:

$$x_{i+1} = T(4, y_i) \tag{2}$$

$$y_{i+1} = F(y_i) + T(4, x_i) \tag{3}$$

and reverse propagation can be computed by the following:

$$y_i = T^{-1}(4, x_{i+1}) \tag{4}$$

$$x_i = T^{-1}(4, (y_{i+1} - F(y_i))) \tag{5}$$

where T^{-1} represents reverse calculation of T function.

3.4 Spatial Attention

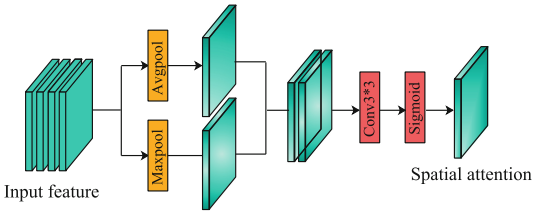

Fig. 4. Diagram of spatial attention module.

The spatial attention module aims to highlight the expressions of key objects for image retrieval. Firstly, it learns a set of weight maps from the feature maps, and provides a larger weight for the informative region in each feature map, while providing a smaller weight for the background region. Then, the learned weight maps is multiplied by the feature map, so feature maps focusing on key objects and suppressing background regions is obtained. More specifically, the spatial attention tell which information to emphasize or suppress in the process of feature transmission. As shown in Fig. 3, feature maps are send to a Max-pooling and average pooling operation respectively, both which demonstrate effective in highlighting informative regions. Then concatenating the both outputs to generate a concentrated feature descriptor. Next, we apply a convolution layer followed by sigmoid operation on the attentioned feature descriptor to get a spatial attention map $SA(Fe) \in R^{H \times W}$, which tell information flow which part to emphasize or suppress. In short, the detailed operation is described as below:

$$SA(Fe) = \sigma(g^{7\times7}(Cat(Ap(Fe), Mp(Fe)))) \tag{6}$$

where σ presents the sigmoid operation, $g^{7\times7}$ denotes a convolution operation with kernel size of 7×7, cat is concatenation operation along the channel axis, $Ap(Fe)$ and $Mp(Fe)$ respectively represent average pooling and Max-pooling operation, and Fe is a brief expression of feature map.

3.5 Loss Function

In our paper, we focus on the supervised setting utilizing label information. We can easily obtain a set of image pairs, where each pair (a_i, a_j) consists of an image a_i and $a_j (j \neq i)$. Using both category information, we can get the similarity s_{ij} of image pair (a_i, a_j). Following [3,33], the similarity information is constructed directly by image labels:if two images a_i and a_j share at least one label, they are similar and $s_{ij} = 1$; otherwise, they are dissimilar and $s_{ij} = 0$.

Intuitively, the desired hash codes should be able to preserve the relative similarities in the image pairs. Corresponding optimization goal is to make the Hamming distance between two similar points as small as possible, and simultaneously make the Hamming distance between two dissimilar points as large as possible. In this way, we can define a pairwise loss that has also been successfully applied in prior research [16], which is defined over the output binary codes (b_i, b_j) corresponding to the training image pair (a_i, a_j):

$$l_1 = min(-(s_{ij}\beta_{ij} - log(1 + e^{\beta_{ij}}))) \tag{7}$$

where $\beta_{ij} = \frac{1}{2} b_i^T b_j$, s_{ij} presents the similarity of image pair (a_i, a_j). To pursue representative hash codes, we learn our Invertible Hashing Network by minimizing the pairwise loss. This can drive our network to process strong capability of distinguishing the images. Since the hash codes are discrete, we additionally adopt the following quantization loss for each image a_i:

$$l_2 = ||b_i - u_i||_2^2 \tag{8}$$

where $u_i \in \mathbb{R}^{c \times 1}$, $b_i \in (-1, 1)^c$, and c represent the hash code length. Based on the two type of loss, we can train our Invertible Hashing Network through the following function:

$$l = l_1 + l_2 = -\sum_{s_{ij} \in S} (s_{ij}\beta_{ij} - log(1 + e^{\beta_{ij}})) + \lambda \sum_{i=1}^{n} ||b_i - u_i||_2^2 \tag{9}$$

where λ is the hyper-parameter; $||.||_2$ denotes the l_2 norm.

4 Experiments

4.1 Dataset and Evaluation

We evaluate the effect of the proposed AIHN with several state-of-the-art hashing methods on two benchmark datasets.

– CIFAR-10 is a single-label dataset with 60000 images divided into 10 categories (6000 images per class). We follow [2,33] to randomly select 100 images per class for training, 500 images per class for testing, and the rest 54000 images used as database.

– NUS-WIDE81 is a multi-label dataset, which contains 269648 images consisting of 81 categories. We follow similar experimental protocols in [3,33], and randomly sample 5000 images as test images, 10000 images for training, and the remaining images used as database. To evaluate our method, the mean average precision (MAP) is used to measure the accuracy of our proposed method and other baselines (Fig. 5).

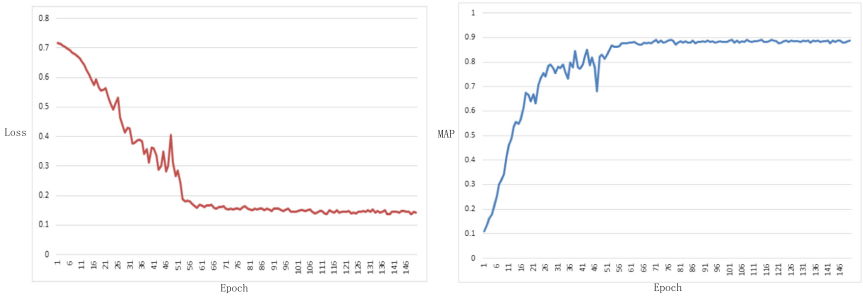

Fig. 5. The curve convergence of our network on CIFAR-10 with 16 bits code.

4.2 Implementation Detail

Network Detail. The proposed network is trained specifically for image hashing retrieval. The input image size of our network is $3 \times 224 \times 224$. After the invertible downsampling with $\theta = 4$, the size of features becomes $12 \times 112 \times 112$. Then, the spliting operation guarantees two sublayers with equal channel. Next, both sublayers pass through totally 100 similar invertible blocks. It will correspond to an invertible downsampling operator respectively at the block = 6, 22, 94. The spatial resolution of these layers is reduced by a factor 4 while increasing the number of channels respectively to 48, 192, 768 and 3072. Furthermore, it means that the corresponding spatial resolutions are respectively 56×56, 28×28, 14×14, 7×7. Last, the obtained representation is spatially averaged and projected onto one-dimensional vector after a ReLU nonlinearity. Binary hashing code can be obtained through Sgn computation conducted on the one-dimensional vector.

Training Detail. We randomly crop a set of 224×224 patches for training. The training batch size is set to 64 in each back-propagation. This network is trained via an end-to-end manner. Pairwise similarity loss and quantilization loss are concurrently adopted in CIFAR-10 and NUS-WIDE81, where the data augmentation with random horizontal flip is adopted. The SGD is adopted for optimizing our network, and the initial learning rate is set to 0.05. For each 50 epochs, the learning rate will decrease by the scale of 0.1. The hyper-parameter

λ in our network is chosen by a validation set, which is 10 for CIFAR-10 and 100 for NUS-WIDE81. At test time, we rescale the image size to 256×256 and perform a center crop of size 224×224. The curve convergence of our network on CIFAR-10 with 16 bits code are shown below. Experiments are performed on two NVIDIA Titan XP GPUs for training and testing.

4.3 Compare with State-of-the-Arts

We use MAP evaluation metrics to compare retrieval performance of AIHN with classical or state-of-the-art methods: supervised shallow methods ITQ-CCA [8], BRE [11], KSH [18], SDH [20] and supervised deep methods CNNH [22], DNNH [12], DHN [33], HashNet [3]. For fair comparison, all methods use identical training and test sets. We adopt MAP@5000 for evaluation in NUS-WIDES. For shallow hashing methods, we use as image features the 4096-dimensional $DeCAF_7$ feature [4]. For deep hashing methods, we use raw images as the input. We adopt the AlexNet architecture for all deep hashing methods.

Table 1. The best MAPs for each category are shown in boldface. Here, the MAP value is calculated based on the top 5000 returned neighbors for NUS-WIDE dataset.

Method	CIFAR-10 (MAP)				NUS-WIDES (MAP)			
	16-bits	32-bits	48-bits	64-bits	16-bits	32-bits	48-bits	64-bits
ITQ-CCA [8]	0.4258	0.4652	0.4774	0.4932	0.5706	0.4397	0.0825	0.0051
BRE [11]	0.4216	0.4519	0.4002	0.3438	0.5502	0.5422	0.4128	0.2202
KSH [18]	0.4368	0.4585	0.4012	0.3819	0.5185	0.5659	0.4102	0.0608
SDH [20]	0.5620	0.6428	0.6069	0.5012	0.6681	0.6824	0.5979	0.4679
CNNH [22]	0.5512	0.5468	0.5454	0.5364	0.5843	0.5989	0.5734	0.5729
DNNH [12]	0.5703	0.5985	0.6421	0.6118	0.6191	0.6216	0.5902	0.5626
DHN [33]	0.6929	0.6445	0.5835	0.5883	0.6901	0.7021	0.6685	0.5664
HashNet [3]	0.7476	0.7776	0.6399	0.6259	0.6944	0.7147	0.6736	0.6190
AIHN	**0.7897**	**0.7967**	**0.8054**	**0.8076**	**0.7434**	**0.7555**	**0.7599**	**0.7590**

Experimental results are as shown in Table 1. It can be seen that our method AIHN achieves the best performance among all the methods. Specifically, compared to the best shallow hashing method using deep features as input, ITQ-CCA, we achieve absolute boosts of 33.45%, 48% in average MAP for different bits on CIFAR-10 and NUS-WIDE dataset respectively. Compared to the state-of-the-art deep hashing method, HashNet, we achieve absolute boosts of 10.21%, 7.9% in average MAP for different bits on the two datasets, respectively. An interesting phenomenon is that the performance boost of AIHN over HashNet is significantly different across the two datasets. Specifically, the performance boost on NUS-WIDES is much larger than that on CIFAR-10 generally. But with the code length increasing, MAP has an exciting increase in CIFAR10 dataset.

4.4 Ablation Experiment

For investigating the effectiveness of proposed two different components, we research two AIHN variants: (1) AIHN-AI is a AIHN variant without using spatial attention module, and replace invertible network with Alexnet which may cause gradually information lost. (2) AIHN-A is a AIHN variant using invertible network for feature extraction. But in each invertible block, there is no spatial attention module adopted.

AIHN-A outperforms AIHN-AI by very large margins of 10.58%, 8.69%, 10.71% and 9% in MAP with corresponding 16, 32, 48, 64 code lengths on CIFAR-10. The invertible Network guarantees that the final hash codes can be completely obtained from input images without any information loss. AIHN outperforms AIHN-A by 0.75%, 1.48%, 0.03%, 1.77% in MAP with different 16, 32, 48, 64 code lengths on CIFAR-10 respectively. These results validate that the spatial attention module can enhance efficiency and improve MAP results. That is because the spatial attention module can better capture the objective information. As shown in Table 2, our proposed AIHN achieves the highest result in terms of the MAP evaluation metrics in CIFAR-10 dataset. Further analysis, we can find that Invertible Network which guarantees the lossless generated features contributes to our network largely. This can be explained as the following: when learning image features, progressively discarding variability about the input image may cause effective information to be discarded.

Table 2. Results of ablation study on CIFAR-10

Method	CIFAR-10(MAP)			
	16-bits	32-bits	48-bits	64-bits
AIHN-AI	0.6764	0.6950	0.6980	0.6999
AIHN-A	0.7822	0.7819	0.8051	0.7899
AIHN	**0.7897**	**0.7967**	**0.8054**	**0.8076**

5 Conclusion

In this paper, we propose a novel attention-aware invertible hashing network for image retrieval. By invertible feature representations, the final hash codes can be completely obtained from input images without any information loss, so as to produce accurate hash codes with complete image information. For highlighting informative regions, we present a novel attention-aware invertible block as the basic module of AIHN, which can promote generalization ability by spatial attention mechanism. Extensive experiments conducted on benchmark datasets have demonstrated the state-of-the-art performance of our method.

Acknowledgement. This work was supported by National Key R&D Program of China (2018YFB0803700) and National Natural Science Foundation of China (61602517, 61877002).

References

1. Cao, Y., Long, M., Wang, J., Liu, S.: Collective deep quantization for efficient cross-modal retrieval. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
2. Cao, Y., Long, M., Wang, J., Zhu, H., Wen, Q.: Deep quantization network for efficient image retrieval. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
3. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: deep learning to hash by continuation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5608–5617 (2017)
4. Donahue, J., et al.: Decaf: a deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning, pp. 647–655 (2014)
5. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: VLDB, vol. 99, pp. 518–529 (1999)
6. Gomez, A.N., Ren, M., Urtasun, R., Grosse, R.B.: The reversible residual network: backpropagation without storing activations. In: Advances in Neural Information Processing Systems, pp. 2214–2224 (2017)
7. Gong, Y., Kumar, S., Rowley, H.A., Lazebnik, S.: Learning binary codes for high-dimensional data using bilinear projections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 484–491 (2013)
8. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2013)
9. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3588–3597 (2018)
10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
11. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: Advances in Neural Information Processing Systems, pp. 1042–1050 (2009)
12. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3270–3278 (2015)
13. Li, C., Liu, Q., Liu, J., Lu, H.: Ordinal distance metric learning for image ranking. IEEE Trans. Neural Netw. Learn. Syst. **26**(7), 1551–1559 (2014)
14. Li, C., Wang, X., Dong, W., Yan, J., Liu, Q., Zha, H.: Joint active learning with feature selection via cur matrix decomposition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(6), 1382–1396 (2018)
15. Li, C., Wei, F., Dong, W., Wang, X., Liu, Q., Zhang, X.: Dynamic structure embedded online multiple-output regression for streaming data. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 323–336 (2018)
16. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. arXiv preprint arXiv:1511.03855 (2015)
17. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2064–2072 (2016)
18. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2074–2081. IEEE (2012)

19. Liu, Y., Zhu, X., Zhao, X., Cao, Y.: Adversarial learning for constrained image splicing detection and localization based on atrous convolution. IEEE Trans. Inf. Forensics Secur. (2019)
20. Shen, F., Shen, C., Liu, W., Tao Shen, H.: Supervised discrete hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 37–45 (2015)
21. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems, pp. 1753–1760 (2009)
22. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
23. Yu, F., Kumar, S., Gong, Y., Chang, S.F.: Circulant binary embedding. In: International Conference on Machine Learning, pp. 946–954 (2014)
24. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318 (2018)
25. Zhang, P., Zhang, W., Li, W.J., Guo, M.: Supervised hashing with latent factor models. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 173–182. ACM (2014)
26. Zhang, X.Y.: Simultaneous optimization for robust correlation estimation in partially observed social network. Neurocomputing **205**, 455–462 (2016)
27. Zhang, X.Y., Shi, H., Li, C., Zheng, K., Zhu, X., Duan, L.: Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision. arXiv preprint arXiv:1902.07370 (2019)
28. Zhang, X.Y., Shi, H., Zhu, X., Li, P.: Active semi-supervised learning based onself-expressive correlation with generative adversarial networks. Neurocomputing **345**, 103–113 (2019)
29. Zhang, X.Y., Wang, S., Yun, X.: Bidirectional active learning: a two-way exploration into unlabeled and labeled data set. IEEE Trans. Neural Netw. Learn. Syst. **26**(12), 3034–3044 (2015)
30. Zhang, X.Y., Wang, S., Zhu, X., Yun, X., Wu, G., Wang, Y.: Update vs. upgrade: modeling with indeterminate multi-class active learning. Neurocomputing **162**, 163–170 (2015)
31. Zhang, X.: Interactive patent classification based on multi-classifier fusion and active learning. Neurocomputing **127**, 200–205 (2014)
32. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 294–310. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_18
33. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
34. Zhu, X., Li, Z., Zhang, X.Y., Li, C., Liu, Y., Xue, Z.: Residual invertible spatio-temporal network for video super-resolution. In: AAAI Conference on Artificial Intelligence (2019)
35. Zhu, X., Li, Z., Zhang, X., Li, H., Xue, Z., Wang, L.: Generative adversarial image super-resolution through deep dense skip connections. In: Computer Graphics Forum, vol. 37, pp. 289–300. Wiley Online Library (2018)
36. Zhu, X., Liu, J., Wang, J., Li, C., Lu, H.: Sparse representation for robust abnormality detection in crowded scenes. Pattern Recogn. **47**(5), 1791–1799 (2014)
37. Zhu, X., Zhang, X., Zhang, X.Y., Xue, Z., Wang, L.: A novel framework for semantic segmentation with generative adversarial network. J. Vis. Commun. Image Represent. **58**, 532–543 (2019)