Neural Computing and Applications
https://doi.org/10.1007/s00521-022-08141-0

ORIGINAL ARTICLE

How visual chirality affects the performance of image hashing

Yanzhao Xie[1] · Guangxing Hu[1] · Yu Liu[1,2] · Zhiqiu Lin[3] · Ke Zhou[1] · Yuhong Zhao[4]

Received: 21 March 2022 / Accepted: 29 November 2022

Abstract
Visual chirality reveals the phenomenon that chiral data will present different semantics after flipping. Although image flipping is widely used in image hashing learning as a data augmentation technique, the effect of learning chiral image data on hashing performance has not been fully discussed. To explore this issue, this paper first designs an approach to recognize images with chiral cues, then constructs the chiral datasets including different proportions of images with chiral cues, and finally analyzes and discusses the performance change via testing three representative image hashing methods with different hash code lengths on constructed chiral datasets. In addition, to understand the effect of visual chirality from an internal perspective, we illustrate visual results of activated regions between some original images with chiral cues and their flipped ones. We conduct the above experiments on three public image datasets including VOC2007, MS-COCO, and NUS-WIDE. Experimental results reveal that different proportions of chiral data will greatly affect the performance of image hashing and the best performance appears when the proportion of images with chiral cues accounts for 15% ∼ 25% or 75% ∼ 85%. The code of this work is released at: https://github.com/lzHZWZ/Visual_Chirality_Hashing.

Keywords Visual chirality · Data augmentation · Image hashing · Performance change

1 Introduction

Over the past decades, image hashing [1–4] has been an important research topic in the computer vision community and become a commonly used means in the image retrieval field owing to its compact binary codes and efficient XOR comparison. To meet the practical requirements with generalization ability, researchers used to train hashing models using data augmentation technologies. Note that the existing data augmentation technologies usually transform data by means of geometric transformations [5], color space transformation [6], random erasing [7], generative adversarial networks [8–10], etc., which enhance the perception of semantic information by adjusting related information from original data. Nevertheless, these methods ignore the influence of original semantics on the related information, resulting in that some images cannot express

Guangxing Hu contributed equally to this work.

✉ Yu Liu
liu_yu@hust.edu.cn

✉ Yuhong Zhao
zhaoyuhong@iie.ac.cn

Yanzhao Xie
yzxie@hust.edu.cn

Guangxing Hu
Garson_hu@hust.edu.cn

Zhiqiu Lin
zhiqiul@andrew.cmu.edu

Ke Zhou
zhke@hust.edu.cn

[1] Wuhan National Laboratory for Opto-electronics, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, Hubei, China

[2] School of Computer Science and Technology, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, Hubei, China

[3] Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213-3890, USA

[4] Institute of Information Engineering, Chinese Academy of Sciences, Minzhuang Road 89, Beijing 100093, China

Published online: 25 December 2022

the same semantic information after transformation. As shown in Fig. 1, the activated region of a single-object image (Fig. 1a) has been obviously changed after we flip it to Fig. 1b. Besides, this phenomenon even has been intensified when we flip a multi-object image Fig. 1c to d, which may result in inconsistent hash codes for the same image and severely affect the performance of image hashing. As shown in Fig. 2, we, respectively, draw the t-SNE [11] distribution of hash codes on an original image dataset and its flipped one. As we see, the distribution of Fig. 2a is different from that of Fig. 2b, which illustrates that the semantic information of a dataset has been changed after image flipping. According to our statistics, there exist about 20% images that will be activated with different semantic information after flipping on VOC2007 dataset. Therefore, to reveal the effect generated by this phenomenon for better performance, it is necessary to explore the influence of semantic distribution of data on hashing methods using data augmentation technologies.

The discovery of visual chirality phenomenon is the breakthrough to study this principle. An image with chiral cues means that the semantic information (feature) contained in this image will be changed once we apply image transforms like flipping, rotation or translation operations to this image. However, as far as we know, only Lin et al. [12] observed and revealed this phenomenon that the data distribution has been greatly changed after image mirror flipping, but they did not talk about how to select chiral data or how chiral data affects the model performance. Following these tasks, we first design an effective sifting method to pick pure-chiral data based on [12, 13] and then explore the performance by varying the proportion of pure-chiral data in the public datasets. Based on datasets with predefined chiral data proportion, we evaluate the retrieval performance of hash codes generated by the representative hashing algorithms with data augmentation of flipping. As the first work to explore the visual chirality phenomenon in the field of image hashing, we aim to reveal the relationship between chiral data and performance of image hashing, and enlighten researchers to employ appropriate proportion of images with chiral cues to obtain the superior image hashing performance according to the given task.

In the implementation, the work flow of our evaluation consists of three steps. First, we use ResNet-50 [14] to design an effective sifting method to pick pure-chiral data from public datasets including VOC2007 [15], MS-COCO [16] and NUS-WIDE [17]. Next, we vary the proportion of pure-chiral data to form multiple test sets. Based on this, we finally adopt HashNet [18] (the classical hashing method), DCH [19] (the state-of-the-art hashing method), and BYOL [20]+DCH (the state-of-the-art self-supervised method + the state-of-the-art hashing method) to test the performance on the above generated datasets. Besides, we also illustrate visual results of activated regions between some original images with chiral cues and their flipped ones to better understand the visual chirality from an internal perspective. The experimental results verify our conjecture that visual chirality will greatly affect the performance of image hashing. The best performance appears when the proportion of chiral data accounts for 15% ∼ 25%

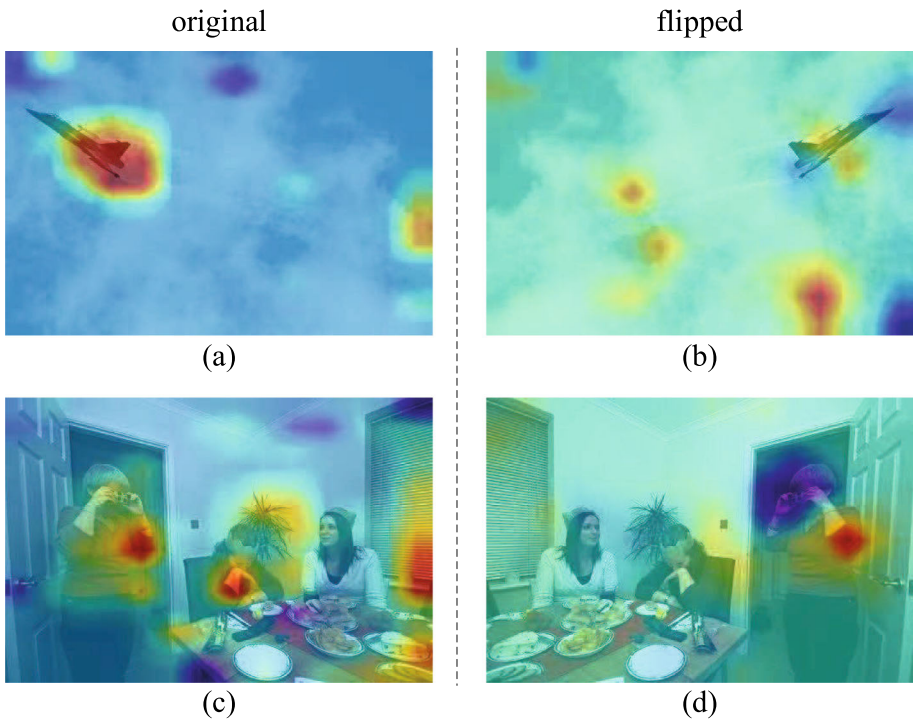


Fig. 1 Activated regions of the Original and Flipped images

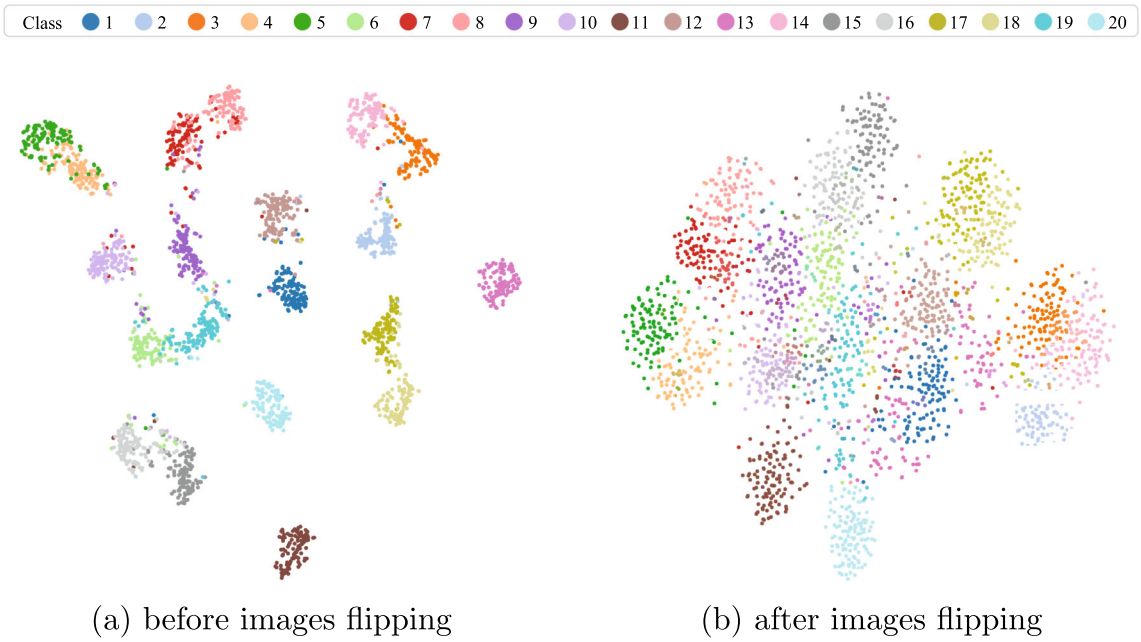


Fig. 2 The t-SNE visualization of hash codes before and after all images flipping on VOC2007

or 75% ∼ 85%. The main contributions of this paper can be summarized below.

(1) We observe the visual chirality phenomenon on image hashing and design an effective scheme to pick chiral datasets.
(2) This is the first study to explore how visual chirality affects the performance of image hashing with different proportions of chiral data.
(3) Extensive experimental results on VOC2007, MS-COCO and NUS-WIDE verify our conjecture and reveal that it will produce better performance for image hashing methods when the proportion of chiral data accounts for 15% ∼ 25% or 75% ∼ 85%.

The rest of this paper is organized as follows. Section 2 talks about related works. We elaborate the work flow of our scheme in Sect. 3 and analyze the experimental results in Sect. 4. At last, we conclude this paper in Sect. 5.

2 Related work

2.1 Visual chirality

If an object cannot be rotated and translated into alignment with its own reflection, we call one is chiral or geometric chirality. Geometric chirality is a binary property of objects. In contrast, visual chirality [12] describes the changes in data distribution due to data augmentation. Most of the work related to visual chirality involves the exploration of symmetric relations, which may be temporal or spatial. Explorations of this temporal relationship include the study of the "Time's arrow" of videos by [21] and [22]. This time asymmetry can be regarded as a kind of temporal chirality to some extent, through which we can understand what makes videos look like they are being played forwards or backwards. The spatial version of chirality is related to other orientation problem, such as detecting "which way is correct." Considering the possible of 3D model that might be oriented incorrectly, Liu et al. [23] use convolutional network to inspect 3D object upright orientation estimation.

The problem of chirality can also be regarded as a variation of the classical task of detecting whether the image is symmetrical or not [24, 25]. These work uses neural networks to determine the symmetry in the image. As such, our work involves the detection and classification of asymmetric chiral objects. For example, how to determine whether the hand using the keyboard is left or right hand [26]. Most of these prior works generally analyzed the nature of geometric chirality, which is not equivalent to the concept of visual chirality that this work talked about. For example, a right hand has geometric chirality but not visual chirality, while due to the prevalence of right-handed people, the right hand using a mouse might be visually chiral. The visual chirality is widespread in most of the datasets, but the existing studies have seldom explored this issue. Motivated by our observation, we explore and analyze how chiral datasets affect the performance of image hashing in detail.

2.2 Image hashing with data augmentation

Image hashing is an approach of mapping the image from a high-dimensional vector to a low-dimensional representation, or equivalently a short code consisting of a sequence of bits. After realizing the data-independent hashing methods like LSH [27] that cannot achieve ideal performance, researchers find that the accuracy of image hashing depends on the perception of key objects in the image. Thus, due to the advantage of deep learning in object perception, deep image hashing has been widely studied and achieved the state-of-the-art performance [19, 28, 29]. To enhance the perception of deep models, some deep hashing methods use data augmentation technologies to transform the image expression, but they require activating the same regions after performing image transforms. Feng et al. [30] employ four types of data augmentation methods: cropping, scaling, flip and color shuffle to introduce variations. Long et al. [19] take image rotation and flipping as the necessary transformation of training input to complete hashing learning. Lin et al. [31] introduce the data augmentation technique of rotation into the loss function, by minimizing the difference between the binary descriptors describing the reference image and the rotated image to maintain the rotation invariance of the local binary descriptors. With the development of unsupervised and self-supervised deep learning methods [32], researchers use data augmentation technologies to carry out contrastive learning without handcrafted labels on the default assumption that the semantics of transformed data are the same as the original data.

However, the visual chirality phenomenon does not support this assumption illustrated in Figure 1. Thus, it is necessary to explore the rationality of hashing learning with visual chirality data using data augmentation technologies.

3 Proposed method

In this section, we elaborate the work flow of our evaluation scheme shown in Fig. 3, which mainly consists of three steps. Firstly, we design an effective method to recognize whether an image in public datasets (i.e., VOC2007, MS-COCO and NUS-WIDE) owns chiral cues via ResNet-50 classification model. Secondly, based on the classification results, we construct multiple chiral datasets with different proportions of images with chiral cues. Finally, we employ three representative deep hashing models (i.e., HashNet, DCH and BYOL+DCH) to evaluate the performance on the above chiral datasets. In the following, we will, respectively, introduce each part of the work flow in detail.

3.1 Chiral data classification

In this part, we recognize whether an image owns chiral cues by comparing the classification result of each original image and its flipped one. Given a dataset D consisting of N images $X = \{x_1, x_2, \cdots, x_N\}$, we flip all N images to generate a corresponding flipped dataset D' consisting of $X' = \{x_1^{'}, x_2^{'}, \cdots, x_N^{'}\}$. For each image $x \in X$, we allocate one binary label $y \in \{1, 0\}$ to this original image, while its corresponding flipped $x' \in X'$ will obtain a label $y' \in \{0, 1\}$. Next, we adopt ResNet-50 pretrained on ImageNet [33] and fine-tune its *fc* (fully connected) layer to train a binary classification model as follows:

$$\begin{aligned} y_p &= F(x; \theta), \\ y_p^{'} &= F(x^{'}; \theta), \end{aligned} \tag{1}$$

where y_p and $y_p^{'}$, respectively, denote the prediction labels of x and x', F denotes ResNet-50, and θ denotes the parameters of this network. This network will be updated using the commonly used cross-entropy loss function below.

$$\begin{aligned} \mathcal{L} = &-\sum (y log(y_p) + (1-y) log(1-y_p)) \\ &-\sum (y^{'} log(y_p^{'}) + (1-y^{'}) log(1-y_p^{'})), \end{aligned} \tag{2}$$

where $\mathcal{L}$ denotes the loss function. We will train this network until its accuracy exceeds 95%, thereby producing a chiral data classification model. Therefore, each sample in D will be regarded as an image with chiral cues only if it obtains a prediction label (1, 0) and its flipped one obtains a predicted label (0, 1).

3.2 Chiral datasets construction

After the classification of images, we aim to establish chiral datasets with different proportions of images with chiral cues in this section. First, for a dataset D consisting of N images, we could pick all the K ($K \leq N$) images with chiral cues from these N images based on the above classification network to form a set $D_K = \{x_1, x_2 \ldots x_K\}$, and there are another set $D_{N-K} = \{x_{K+1}, x_{K+2} \ldots x_N\}$ left. Next, we will randomly select images with different proportions from the above two sets to form chiral datasets. For example, if we would like to construct a chiral dataset $C_{10\%}$ that only contains 10% images with chiral cues, we will first randomly choose $\frac{1}{10}K$ images from D_K and then randomly choose $\frac{9}{10}K$ images from D_{N-K}. In this way, we can

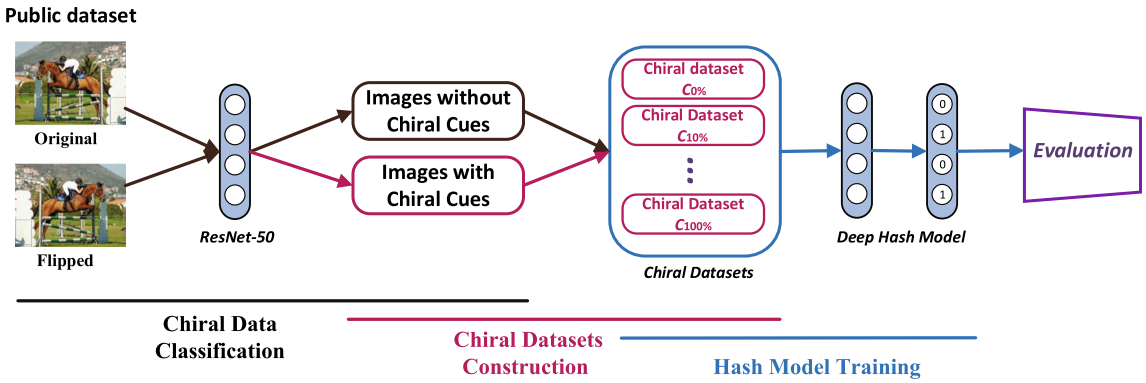


Fig. 3 The work flow of our evaluation

construct multiple chiral datasets with different proportions of images with chiral cues, such as $C_{0\%}, C_{10\%}, \cdots, C_{100\%}$.

As an exploratory work, we try to explain that how the datasets containing different proportion of chiral data affect the performance of image hashing.

After the construction of chiral datasets, each chiral dataset is divided into training set, validation set and test set, which will be input to the hashing models to train and test the performance of image hashing.

The division of chiral datasets will be introduced in Sect. 4. In the following, we will talk about the hashing models used for performance evaluation.

3.3 Hashing model training

Given a training set with M points $\{x_i\}_{i=1}^{M}$, each sample is represented by a d-dimensional feature vector $x_i \in \mathbb{R}^d$. The goal of deep hashing is to learn a nonlinear hashing function $f : x \mapsto h \in \{-1, 1\}^k$ from original input space $\mathbb{R}^d$ to Hamming space $\{-1, 1\}^k$ using neural networks. As the key component of image retrieval systems, the hashing function encodes each training sample x into k-bit compact binary hash code. In this part, we detail three representative deep hashing methods, i.e., HashNet, DCH, and BYOL+DCH.

HashNet HashNet [18] addresses the deep hashing learning from imbalanced similarity data by a continuation method. We adopt ResNet-101 as the backbone for learning deep image representations. The optimization problem of HashNet is defined as below:

$$\min_{\Theta} \sum_{s_{ij} \in S} \omega_{ij} \left(\log\left(1 + \exp\left(\alpha \langle h_i, h_j \rangle\right)\right) - \alpha s_{ij} \langle h_i, h_j \rangle \right), \tag{3}$$

where Θ denotes parameters of networks, ω_{ij} represents the weight of each training pair (x_i, x_j, s_{ij}), α is hyperparameter to control the bandwidth of *adaptive* Sigmoid function, $S = \{s_{ij}\}$ is the set of pairwise similarity, where each similarity label in S can only be $s_{ij} = 1$ (similar) or $s_{ij} = 0$ (dissimilar), (h_i, h_j) denote the generated hash codes corresponding to training pairs (x_i, x_j), and $\langle h_i, h_j \rangle$ denotes the inner product similarity between h_i and h_j. Note that HashNet uses Tanh as an activation function to approximate the sign function,

$$\lim_{\beta \to \infty} \text{Tanh}\ (\beta z) = \ \text{sgn}\ (z), \tag{4}$$

where $\beta > 0$ is a scaling parameter.

The training process starts with Tanh $(\beta_t z)$ as the activation function, where $\beta_0 = 1$ and we set $t \in \{0, 1, \cdots, 9\}$ by default. HashNet will converge after each stage t with an increasing β_t, and the parameters of the converged model will be taken as the initial parameters for training in the next stage. Finally, as $\beta \to \infty$, the optimization eventually goes back to the original sign activation functions. As a supervised deep hashing method, HashNet first accepts pairwise input images $\{(x_i, x_j, s_{ij})\}$ mentioned in Eq. (3) and then processes them through an end-to-end pipeline of deep representation learning to obtain hash codes.

DCH Deep Cauchy hashing [19] (abbreviated as DCH) improves the Hamming space retrieval by designing a pairwise cross-entropy loss based on Cauchy distribution, which penalizes similar image pairs with Hamming distance larger than the given Hamming radius threshold. DCH employed a Bayesian learning framework, which quantifies the similarity of pairwise images to perform deep hashing on the query image. Given the training images with pairwise similarity labels as $\{(x_i, x_j, s_{ij}) : s_{ij} \in S\}$ which we introduced in Eq. (3), the goal is to maximize the likelihood function:

$$\begin{aligned} \log P(H \mid S) &\propto \log P(S \mid H) P(H) \\ &= \sum_{s_{ij} \in S} \omega_{ij} \log P(s_{ij} \mid h_i, h_j) + \sum_{i=1}^{N} \log P(h_i), \end{aligned} \tag{5}$$

where $P(S \mid H) = \prod_{s_{ij} \in S} [P(s_{ij} \mid h_i, h_j)]^{\omega_{ij}}$ is the weighted

likelihood function, and ω_{ij} is the weight for each training pair, it can be formulated as:

$$\omega_{ij} = \begin{cases} |\mathcal{S}| / |\mathcal{S}_s|, & s_{ij} = 1, \\ |\mathcal{S}| / |\mathcal{S}_d|, & s_{ij} = 0, \end{cases} \tag{6}$$

where $\mathcal{S}_s = \{s_{ij} \in S : s_{ij} = 1\}$ and $\mathcal{S}_d = \{s_{ij} \in S : s_{ij} = 0\}$, respectively, denote similar and dissimilar pairs in one batch. $P(s_{ij} \mid h_i, h_j)$ can be naturally defined by Bernoulli distribution:

$$P(s_{ij|h)i,h_j}) = \sigma(d(h_i, h_j))^{s_{ij}}(1 - \sigma(d(h_i, h_j)))^{1-s_{ij}}, \tag{7}$$

where $d(h_i, h_j)$ is the Hamming distance between hash codes h_i and h_j, and σ is probability function. Most deep learning methods utilize Sigmoid function as probability function. DCH employs a novel probability function based on Cauchy distribution to concentrate relevant images to be within a small Hamming ball and makes efficient Hamming space retrieval possible:

$$\sigma(d(h_i, h_j)) = \frac{\gamma}{\gamma + d(h_i, h_j)}, \tag{8}$$

where γ is the scale parameter of the symmetric Cauchy distribution. Since both DCH and HashNet are supervised methods, we use the same backbone and training process.

BYOL+DCH Bootstrap Your Own Latent (BYOL) is a state-of-the-art self-supervised learning method that relies on two neural networks to learn the representative features for downstream tasks. Since this method does not need to learn the negative samples, it acquires better generalization ability and high practicality. As a method of contrastive learning, BYOL uses the data augmentation technology in the input of two networks. Based on this, we learn the features y_θ as an pretext task by BYOL and learn the hashing function as a downstream task by DCH. We use this hybrid method to explore the influence of chiral features on the independent learning of hashing function.

In the implementation, we first set two neural networks, i.e., **online** network and **target** network. Online network and target network have the same architecture, but their parameters are different. The online network is defined with a set of weight parameters θ while the target network is determined by weights ξ and provides the regression targets to train the online network. The online network is updated with each training batch, while the parameters of target network are an exponential moving average of the online parameters. More precisely, given a target decay rate $\tau \in [0, 1]$, each training phase will perform the following update,

$$\xi \leftarrow \tau\xi + (1 - \tau)\theta. \tag{9}$$

At each training step, BYOL performs a stochastic optimization step to minimize $\mathcal{L}^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \tilde{\mathcal{L}}_{\theta,\xi}$ with respect to θ only. BYOL's dynamics are summarized as follows:

$$\begin{aligned} &\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}^{\text{BYOL}}_{\theta,\xi}, \eta), \\ &\xi \leftarrow \tau\xi + (1 - \tau)\theta, \end{aligned} \tag{10}$$

where we use LARS [34] as the optimizer and η is the learning rate. At the end of the whole training phase, only the online network encoder has to be retained. Based on the network and parameters learned in BYOL and pseudolabels generated by pre-trained model, *i.e.*, ResNet-101, the hashing function of DCH is incorporated to form a new hashing model. We will learn this model by updating the *fc* layer, *fc* hash layer and loss function. When the loss value is stable, we stop the training process to generate the hashing model.

4 Experimental studies

4.1 Datasets

VOC2007 [15] consists of 9,963 multi-label images and 20 object classes, which are divided into training, validation and testing sets. On average, each image is annotated with 1.5 labels. We incorporate the training set, validation set and testing set to complete chiral data classification.

MS-COCO [16] contains 118,287 training images and 40,504 validation images, where each image is averagely labeled with about 2.9 object labels from the 80 semantic class categories. We randomly select 30,000 images from its validation set and complete chiral data classification on the above selected images.

NUS-WIDE [17] consists of 269,648 multi-label images with 161,789 training images and 107,859 test images, where each image is annotated with multiple labels based on 81 concepts. We randomly select 30,000 images from its training set and complete chiral data classification on the above selected images.

Note that we, respectively, obtain 1,934, 775, and 279 images with chiral cues from VOC2007, MS-COCO, and NUS-WIDE. Take VOC 2007 for example, we will construct 11 chiral datasets including different proportions of images with chiral cues according to the principle mentioned in Sect. 3.2, where the proportions are selected from 0%, 10%, 15%, 20%, 25%, 50%, 75%, 80%, 85%, 90%, 100% and the images with chiral cues come from those 1934 images. In this way, we can, respectively, construct 11 chiral datasets from MS-COCO and NUS-WIDE. Each chiral dataset will be split into training set, validation set and test set with a ratio of 7:1:2, where we use the training set and validation set to train each hashing model and evaluate its performance on the test set.

4.2 Implementation details

All the experiments are implemented by PyTorch [35] on the CentOS (64-bit) platform with 4 Tesla V100 GPUs. We, respectively, employ ResNet-50 to train the chiral data classification network and ResNet-101 as the backbone to train hashing models. The input consists of raw images, each of which is resized into 448 × 448. For the HashNet, DCH and BYOL+DCH model, we, respectively, set α as 0.1 to control the bandwidth of *adaptive* Sigmoid function (see Eq. 3), the scale parameter γ as 1.0 (see Eq. 8) and the target decay rate τ as 0.99 (see Eq. 9). During the training process of HashNet and DCH models, the network will be updated using mini-batch stochastic gradient descent (SGD) algorithm with a momentum of 0.9, a batch size of 48, and an initialized learning rate of 0.001 which decays by a factor of 10 every 100 iterations. While for the BYOL+DCH training, the only difference is that all the parameters will be updated by LARS optimizer with an initialized learning rate of 3e-2. Note that we use the standard metric to measure the quality of hash codes within Hamming radius 2, i.e., mean average precision (MAP@H $\leq$ 2) [19]. All the hyperparameters settings are listed in Table 1.

Table 1 The hyperparameters settings

model	hyperparameter	initial value
HashNet	α	0.1
	β	1.0
DCH	γ	1.0
BYOL+DCH	τ	0.99
	η	3e-2

4.3 Experimental results

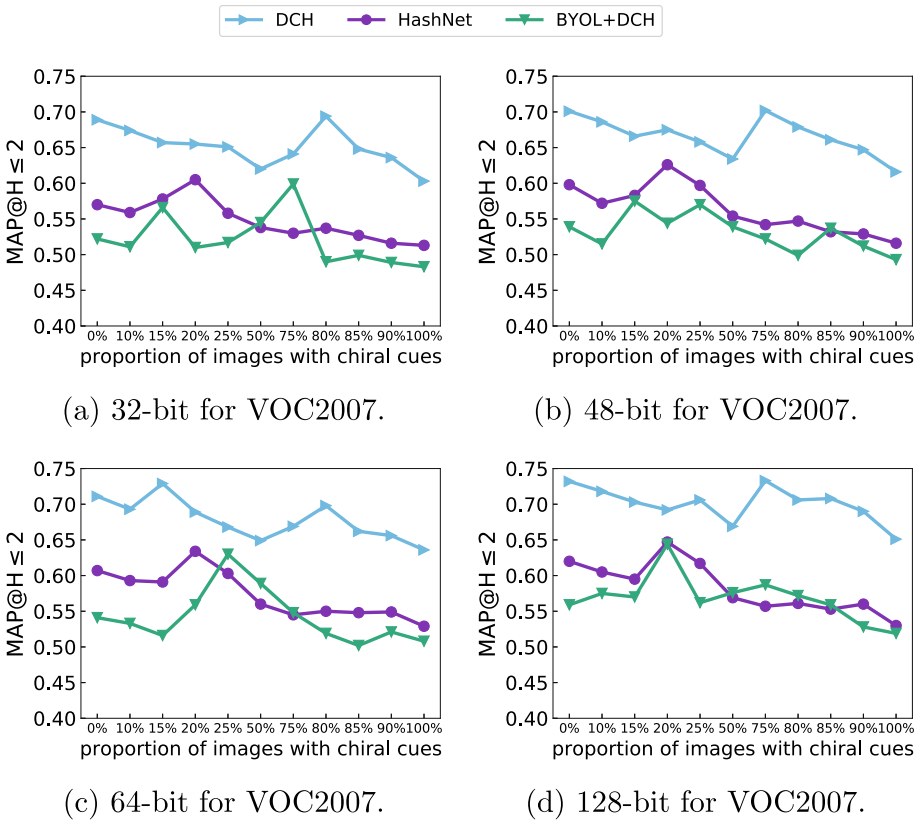
In this section, we first list the results on VOC2007, MS-COCO and NUS-WIDE, then discuss how chiral datasets affect the performance of image hashing, and finally illustrate visual results of activated regions between the original images and flipped ones.

4.3.1 Results on VOC2007

We report the results of MAP@H $\leq$ 2 via three hashing methods on 11 chiral datasets gathered from VOC2007, where the hash code lengths are 32-bit, 48-bit, 64-bit, and 128-bit, respectively.

Figure 4a shows the performance over 32-bit. The performance of HashNet starts from 0.57. Then, it will decrease until the proportion exceeds 10% and rapidly rise to arrive at the peak, i.e., 0.605, when the proportion accounts for 20%. After that, its performance will by and large decline to reach the lowest value 0.513 at 100%. As for the state-of-the-art hashing method, i.e., DCH, it surely obtains higher performance on all the 11 chiral datasets than HashNet. In terms of its performance trend, the MAP@H $\leq$ 2 of DCH starts from 0.689. It steadily drops from 0 to 50% proportion, and then it begins to rapidly rise to obtain the optimal value 0.694 when using 80% images with chiral cues. After the proportion exceeds 80%, its performance will go down to the lowest value 0.603 at 100%. Different from HashNet and DCH, BYOL+DCH is a self-supervised hashing method, so it produces relatively lower overall performance than the previous two ones. Its MAP@H $\leq$ 2 starts from 0.522. It begins to slightly drop from 0 to 10% chiral data and then rapidly rises to the first peak at 15% proportion. After that, the performance of BYOL+DCH first decreases and then continuously increases once the proportion exceeds 20%, and it will obtain the highest value 0.599 at 75%. After this point, its performance will by and large decline to reach the lowest value 0.483 at 100%. From the above results, we can see that HashNet, DCH, and BYOL+DCH will achieve the optimal performance at 20%, 80%, and 75% proportions, respectively. We further analyze the performance trend of these three evaluation methods over 48-bit hash code lengths.

Figure 4b shows the performance over 48-bit. At the beginning, the MAP@H $\leq$ 2 of HashNet is 0.598. It decreases from 0 to 10% proportion. Then, it will increase continuously to reach 0.626 until the proportion accounts for 20%. After this point, it will no longer achieve a higher performance. The lowest value is 0.516 at 100%. As for DCH, its MAP@H $\leq$ 2 starts from 0.701. It will by and large decline from 0 to 50% proportion, and then greatly increase to obtain its highest value 0.702 at 75% proportion. After that, the performance of DCH will continuously go down to the lowest value 0.616 at 100%. Similar to the trend over 32-bit, the overall performance of BYOL+DCH is lower than that of HashNet and DCH. The performance trend of BYOL+DCH has been shaking on these 11 chiral datasets with different proportions of images with chiral cues, but it will, respectively, reach three peaks at 15%, 25% and 85% proportions. Especially, BYOL+DCH yields its optimal performance 0.575 and the lowest one 0.493 when using 15% and 100% images with chiral cues, respectively. From the above results on VOC2007, we find that these three hashing methods achieve the optimal performance over 32-bit and 48-bit when the proportion of images with chiral cues accounts for 15% $\sim$ 25% or 75% $\sim$ 85%. Furthermore, we explore the performance trend over 64-bit and 128-bit.



(a) 32-bit for VOC2007. (b) 48-bit for VOC2007.

(c) 64-bit for VOC2007. (d) 128-bit for VOC2007.

Fig. 4 MAP results on chiral datasets derived from VOC2007 under varying hash code length

As shown in Fig. 4c, the performance trend over 64-bit presents a similar trend to that over 48-bit. HashNet starts with the performance 0.607 and achieves optimal performance 0.634 when the proportion of images with chiral cues accounts for 20%. DCH obtains better performance 0.729 at 15% (peak point). BYOL+DCH produces its highest MAP@H ≤ 2 value 0.63 when using 25% images with chiral cues. As for the trend over 128-bit in Fig. 4d, HashNet, DCH, and BYOL+DCH will achieve the optimal performance 0.647, 0.733, and 0.644 at 20%, 75%, and 20% proportions, respectively. On the whole, we find that the point for peak performance over 64-bit and 128-bit lies in the range 15% $\sim$ 25% or 75% $\sim$ 85%, which is similar to the trend over 32-bit and 48-bit.

Based on all the experimental results on VOC2007, with the proportion increasing, the performance via these three hashing methods will fluctuate. On the whole, they show a downward trend because the lowest value mostly appears at 100%. Interestingly, they obtain the optimal values when this proportion accounts for 15% $\sim$ 25% or 75% $\sim$ 85% over all hash code lengths. Compared with the performance on the datasets without images with chiral cues, the best performance of learning chiral datasets on all hash code lengths is averagely beyond 0.0293, 0.0063 and 0.0718 for three methods, respectively. Therefore, we wonder whether there exists a certain proportion point or a certain proportion range that will bring the optimal performance no matter what hashing methods we used for evaluation. To this end, we further explore the performance trend on MS-COCO.

4.3.2 Results on MS-COCO

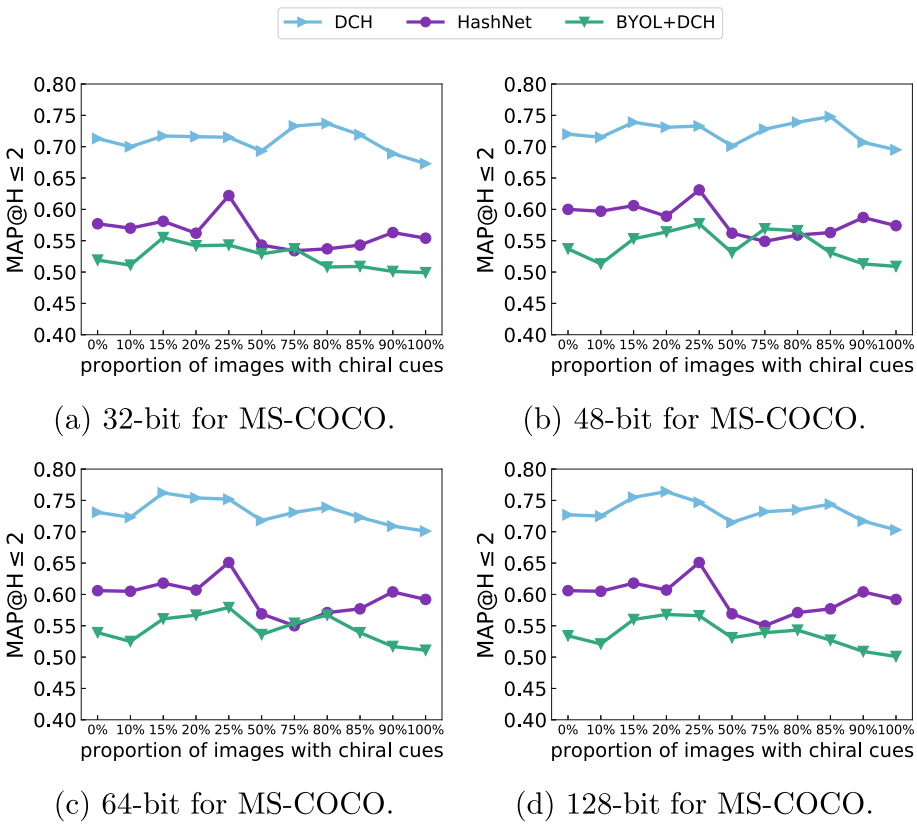
In this part, we report the results on MS-COCO using the same settings of VOC2007.

Figure 5a depicts the performance of three evaluation methods over 32-bit. The MAP@H ≤ 2 of HashNet starts from 0.577 and is stable in the range of 0% to 20%, and then achieves the peak value 0.622 where the proportion of images with chiral cues account for 25%. When the proportion exceeds 25%, its performance will rapidly decline and then arrive at a relatively stable value. As for DCH, its performance starts from 0.713 and has a slight decrease at the range of 0% to 10% and then keeps a relatively high level from 15% $\sim$ 25%. After reaches the lowest performance 0.673 at 50%, its performance will increase until the proportion exceeds 85%. After this point, its performance will continuously go down to 0.673 at 100%. The performance of BYOL+DCH reaches a relatively high level at the proportion range from 15% $\sim$ 75%, and there is no significant change in other proportions.

As shown in Fig. 5b, HashNet reaches its peak value 0.646 when the proportion is 25%, and its overall performance of 11 chiral datasets is better than that of 32-bit. For DCH, it has a superior performance at 75% ~ 85% and reaches the peak performance 0.748 at 85%. The performance fluctuation of DCH is weak in this case. The MAP@H ≤ 2 of BYOL+DCH starts from 0.537, which first decreases from 0% to 10% and then will increase to reach its peak value 0.577 at 25%. After that, it drops to a lower level at 50%. In the range of 75% ~ 80%, it will keep a high value, and then it will continuously drop after 80% proportion and reaches the lowest value 0.695 at 100%. From the above results on MS-COCO, we find that the optimal values over 32-bit and 48-bit appear again in the range of 15% ~ 25% or 75% ~ 85%. We further analyze the performance trend over 64-bit and 128-bit hash code lengths.

Figure 5c shows the performance trend over 64-bit. HashNet achieves the peak performance 0.631 at 25%. Then the performance slumps in the range of 25% to 75% and reaches the lowest value 0.549 at 75%. It will keep a relatively stable value in other proportions. The performance of DCH is better at 15% ~ 25%. After this point, it will no longer achieve a higher value. BYOL+DCH, respectively, achieves two peak values at 25% and 80%, but its overall performance is not as good as HashNet and DCH. Figure 5d shows the performance trend over 128-bit. The performance of HashNet arrives at the peak value 0.674 with 25% chiral data, and then goes down to the lowest value 0.55 at 75%. DCH maintains a better performance at the range from 15% to 25% and reaches the highest value 0.764 at 20%. The performance of BYOL+DCH keeps a high level at the range from 15 to 25%, where the best performance 0.568 appears at 20%. In other proportions, its performance is relatively stable.

Compared with the performance at 0%, the best performance of learning chiral datasets on all hash code lengths has been averagely enhanced by 0.0433, 0.03 and 0.0375 for three methods, respectively. In addition, the probability that the lowest value appears at 100% is 2/3. According to the experimental results on MS-COCO, we find that the performance trend is similar to that on VOC2007. All hashing methods obtain the optimal performance when this chiral data proportion accounts for 15% ~ 25% or 75% ~ 85% as expected. Therefore, we assume that chirality data will promote hashing performance at this proportion range.



(a) 32-bit for MS-COCO.

(b) 48-bit for MS-COCO.

(c) 64-bit for MS-COCO.

(d) 128-bit for MS-COCO.

Fig. 5 MAP results on chiral datasets derived from MS-COCO under varying hash code length

4.3.3 Results on NUS-WIDE

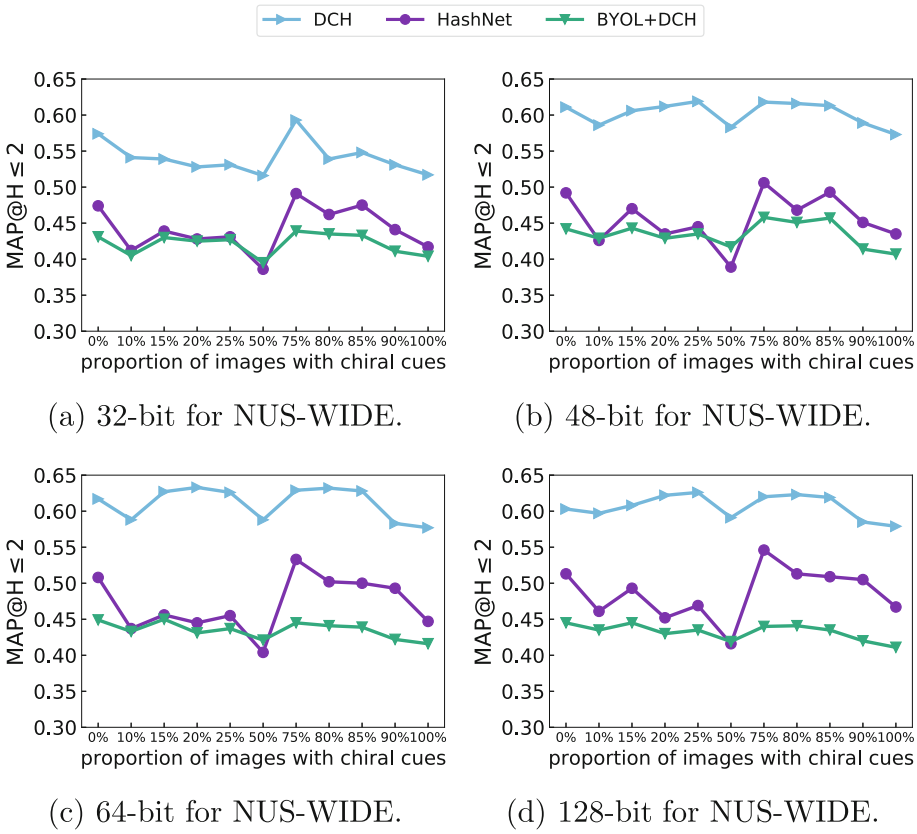
We employ the same settings of VOC2007 and MS-COCO to test the MAP@H ≤ 2 results on NUS-WIDE to confirm our assumption.

Figure 6a describes the performance change over 32-bit. The performance of HashNet starts with 0.474 and slumps from 0 to 10%, and then keeps relatively stable in the range of 10% $\sim$ 25%. Next, it drops to the lowest point 0.386 at 50% and rapidly rises to the peak value 0.491 at 75% proportion. After that, it will gradually go down to a relatively low value. As for DCH, its performance starts from 0.574 and has a slight decrease at the range of 0% to 50%, and then it rapidly achieves the optimal value 0.593 at 75% proportion. After this point, the performance of DCH will also go down to a relatively low value. The performance trend of BYOL+DCH is similar to HashNet and DCH, and it also obtains its highest performance 0.439 at 75% proportion.

As shown in Fig. 6b, the performance of HashNet starts from 0.492 and drops to the lowest MAP@H ≤ 2 value 0.389 at 50%. After that, the performance of HashNet rises to reach its peak value 0.506 at 75%, and then it will gradually drop until 100% proportion. For DCH, on the whole, it owns a high level on all the 11 chiral datasets. Especially, it will yield two peak values at 25% and 75% proportions, respectively, where the best performance 0.619 appears at 25%. The performance of BYOL+DCH will not be greatly changed from 0% from 50% proportion, but it will produce the peak performance at the range of 75% $\sim$ 85%, where the best performance is 0.458 at 75%. These results are consistent with our assumption.

Figure 6c shows the performance trend over 64-bit. The performance of HashNet starts with 0.508 and keeps relatively low at other proportions. It will obtain the optimal value 0.533 at 75% and the lowest value 0.404 at 50%. The performance trend of DCH over 64-bit is similar to that over 32-bit that it also yields two peak values at 25% and 75% proportions, respectively, where the best performance is 0.633 at 20%. As for BYOL+DCH, its performance is basically stable except for two slightly high values at 15% and 75%, where the best performance 0.45 appears at 15%. Furthermore, Figure 6d shows the performance trend over 128-bit. On the whole, the performance trend of these three hashing methods over 128-bit is similar to that over 64-bit.

Similar to the results on MS-COCO, the probability that the lowest value appears at 100% on NUS-WIDE is 2/3. The average performance gaps between the starting point value and peak value are, respectively, 0.0223, 0.0165, and 0.063 for three methods. According to the experimental results on NUS-WIDE, we believe our assumption has been confirmed.



(a) 32-bit for NUS-WIDE. (b) 48-bit for NUS-WIDE.

(c) 64-bit for NUS-WIDE. (d) 128-bit for NUS-WIDE.

Fig. 6 MAP results on chiral datasets derived from NUS-WIDE under varying hash code length

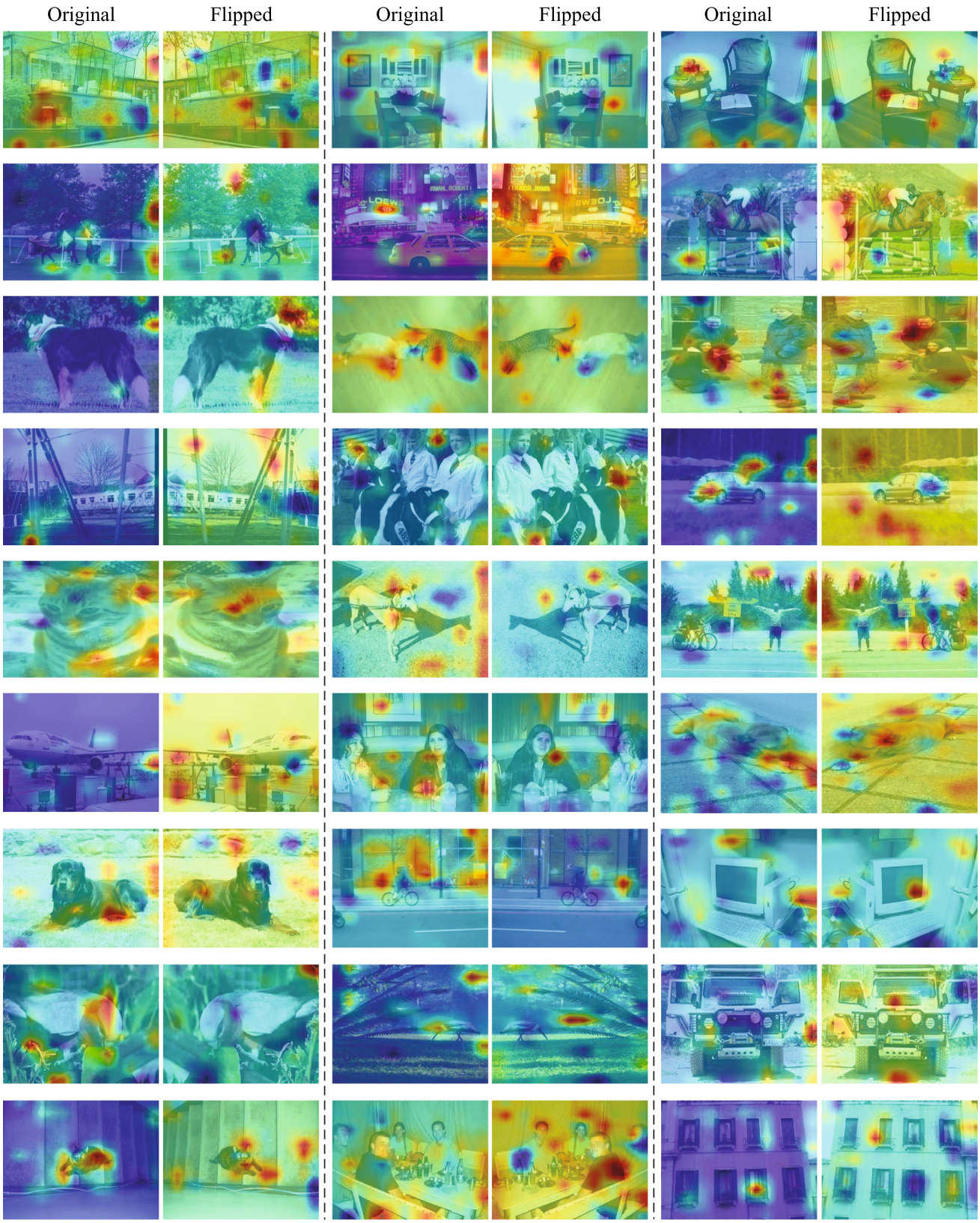


Fig. 7 Visual activated regions of images with chiral cues

4.3.4 Discussion

This is the first study to discuss and analyze the effect of performance caused by visual chirality on image hashing. The previous work [12] only revealed the existence of visual chirality, but it has never explored how this visual chirality affects the performance of a model. From the above experiments on VOC2007, MS-COCO and NUS-WIDE, we find three interesting phenomena.

(1) Different proportions of images with chiral cues will greatly influence the performance of image hashing. As we see, no matter what hashing methods we adopted, the performance will fluctuate with the change of proportion of chiral data. The reason is that different proportions of chiral data will lead to different data distribution, thus affecting the hashing model learning.
(2) Compared with the dataset containing 100% proportion of images with chiral cues, learning the images without chiral cues can bring better performance for hashing model. These overflowed values are 0.0768, 0.0823, and 0.0395 on VOC2007, 0.024, 0.0298, and 0.0273 on MS-COCO, and 0.053, 0.0398, and 0.0323 on NUS-WIDE. However, when we incorporate both of these images for learning, it may bring better performance than that at 0% or worse performance than that at 100%. To our surprise, the worst results may occur when two kinds of images are equal in quantity.
(3) By varying hash code lengths, we find all hashing methods will obtain their optimal performance when the proportion of images with chiral cues accounts for $15\% \sim 25\%$ or $75\% \sim 85\%$. Specifically, this phenomenon continues to happen even if we evaluate the performance on different public datasets. We believe this is not a coincidence. Instead, it really reveals what proportion of chiral data will contribute more to the hashing model learning to yield a higher-performance model. Especially, we have identified a proportion range of chiral data to help train better model. We hope other colleagues in this community can greatly benefit from this exploratory work.

In addition, we still have some items to explore, such as more metrics of evaluation and other advanced data augmentation technologies. We hope that more researchers can invest in this research and provide more data augmentation support for hashing model learning with image or video retrieval.

4.3.5 Visual results

To further reveal the effect on hashing produced by visual chiral, we give examples of images with chiral cues by illustrating the activated regions of each original image and its flipped one, where the activation is determined by a given hashing model. As shown in Fig. 7, we randomly choose 27 images with chiral cues on VOC2007 and highlight the activated regions of each original and flipped image, where the hashing model is HashNet. Obviously, their activated regions have been changed more or less after image flipping, resulting in that the original semantic information contained in this image is no longer (completely) preserved. This inconsistency seriously interferes with the judgment of hashing model for the same features whenever this model uses data augmentation technologies. Figure 7 exemplifies the motivation and necessity of our study.

5 Conclusion and future work

In this paper, we explore how visual chirality affects the performance of image hashing. We design an approach to recognize images with chiral cues and discuss the performance change via testing image hashing methods on constructed datasets with different proportions of chiral data. In addition, we illustrate visual results of activated regions between some original images with chiral cues and their flipped ones. Experimental results reveal that different proportions of chiral data will greatly affect the performance of image hashing and the best performance appears when the proportion of images with chiral cues accounts for $15\% \sim 25\%$ or $75\% \sim 85\%$. However, there exists little research on this interesting chiral topic, but we can only reveal the inherent insights from extensive experimental studies. Instead, we believe there perhaps exists more theoretical explanation about this observation that remains to be further explored. In addition to image hashing tasks, we believe chiral data will also affect the performance of other computer vision tasks. We give our observation and experimental analysis in this paper, and we hope other peers in this community can also pay more attention to this topic that is easy to be ignored. We will continue to push forward this work. As more and more researchers join this team, more intrinsic laws about this chiral cues topic will be greatly revealed in the future.

Acknowledgements Thanks for the support of the National Natural Science Foundation of China No. 61902135 and the National Natural Science Foundation of China Grant No. 62232007. This work was also achieved in Key Laboratory of Information Storage System and Ministry of Education of China.

Data availability The datasets generated during and/or analyzed during the current study are available in the open-source github repository: https://github.com/lzHZWZ/Visual_Chirality_Hashing.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Wang Y, Song J, Zhou K, Liu Y (2021) Unsupervised deep hashing with node representation for image retrieval. Pattern Recognit 112:107785. https://doi.org/10.1016/j.patcog.2020.107785
2. Zhou K, Liu Y, Song J, Yan L, Zou F, Shen F (2015) Deep self-taught hashing for image retrieval. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM, pp 1215–1218. ACM, Brisbane. https://doi.org/10.1145/2733373.2806320
3. Liu Y, Wang Y, Song J, Guo C, Zhou K, Xiao Z (2020) Deep self-taught graph embedding hashing with pseudo labels for image retrieval. In: IEEE International Conference on Multimedia and Expo, ICME, pp 1–6. IEEE, London. https://doi.org/10.1109/ICME46284.2020.9102819
4. Liu Y, Song J, Zhou K, Yan L, Liu L, Zou F, Shao L (2019) Deep self-taught hashing for image retrieval. IEEE Trans Cybern 49(6):2229–2241. https://doi.org/10.1109/ICME46284.2020.9102819
5. Hoorick BV, Vondrick C (2021) Dissecting image crops. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, pp 9721–9730. IEEE, Montreal. https://doi.org/10.1109/ICCV48922.2021.00960
6. Jurio A, Pagola M, Galar M, Lopez-Molina C, Paternain D (2010) A comparison study of different color spaces in clustering based image segmentation. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications - 13th International Conference, IPMU. Communications in Computer and Information Science, vol. 81, pp 532–541. Springer, Dortmund. https://doi.org/10.1007/978-3-642-14058-7_55
7. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2020) Random erasing data augmentation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI, pp 13001–13008. AAAI Press, New York. https://ojs.aaai.org/index.php/AAAI/article/view/7000
8. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial networks. CoRR arXiv:abs/1406.2661
9. Frid-Adar M, Diamant I, Klang E, Amitai M, Goldberger J, Greenspan H (2018) Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing 321:321–331. https://doi.org/10.1016/j.neucom.2018.09.013
10. Lim SK, Loo Y, Tran N, Cheung N, Roig G, Elovici Y (2018) DOPING: generative data augmentation for unsupervised anomaly detection with GAN. In: IEEE International Conference on Data Mining, ICDM, pp. 1122–1127. IEEE Computer Society, Singapore. https://doi.org/10.1109/ICDM.2018.00146
11. van der Maaten L, Hinton G (2008) Visualizing highdimensional data using t-sne. J Mach Learn Res (JMLR) 9(Nov):2579–2605
12. Lin Z, Sun J, Davis A, Snavely N (2020) Visual chirality. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, pp 12292–12300. Computer Vision Foundation / IEEE, Seattle. https://doi.org/10.1109/CVPR42600.2020.01231
13. Zheng Y, Zhang Y, Xu X, Wang J, Yao H (2021) Visual chirality meets freehand sketches. In: 2021 IEEE International Conference on Image Processing, ICIP, pp 1544–1548. IEEE, Anchorage. https://doi.org/10.1109/ICIP42928.2021.9506772
14. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. CoRR arXiv:abs/1512.03385
15. Everingham M, Gool LV, Williams CKI, Winn JM, Zisserman A (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338. https://doi.org/10.1007/s11263-009-0275-4
16. Tsung-Yi Lin SJB Michael Maire et al. (2014) Microsoft COCO: common objects in context. In: Computer Vision - ECCV 2014 - 13th European Conference. Lecture Notes in Computer Science, vol. 8693, pp 740–755. Springer, Zurich. https://doi.org/10.1007/978-3-319-10602-1_48
17. Chua T, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) NUS-WIDE: a real-world web image database from National University of Singapore. In: Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR. ACM, Santorini Island. https://doi.org/10.1145/1646396.1646452
18. Cao Z, Long M, Wang J, Yu PS (2017) Hashnet: deep learning to hash by continuation. In: IEEE International Conference on Computer Vision, ICCV, pp 5609–5618. IEEE Computer Society, Venice. https://doi.org/10.1109/ICCV.2017.598
19. Cao Y, Long M, Liu B, Wang J (2018) Deep cauchy hashing for hamming space retrieval. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1229–1237. Computer Vision Foundation / IEEE Computer Society, Salt Lake City. https://doi.org/10.1109/CVPR.2018.00134
20. Jean-Bastien Grill FA Florian Strub et al. (2020) Bootstrap your own latent –a new approach to self-supervised learning. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual. https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html
21. Pickup LC, Pan Z, Wei D, Shih Y, Zhang C, Zisserman A, Schölkopf B, Freeman WT (2014) Seeing the arrow of time. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 2043–2050. IEEE Computer Society, Columbus. https://doi.org/10.1109/CVPR.2014.262
22. Wei D, Lim JJ, Zisserman A, Freeman WT (2018) Learning and using the arrow of time. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 8052–8060. Computer Vision Foundation / IEEE Computer Society, Salt Lake City. https://doi.org/10.1109/CVPR.2018.00840
23. Liu Z, Zhang J, Liu L (2016) Upright orientation of 3d shapes with convolutional networks. Graph Model 85:22–29. https://doi.org/10.1016/j.gmod.2016.03.001
24. Krippendorf S, Syvaeri M (2021) Detecting symmetries with neural networks. Mach Learn Sci Technol 2(1):15010. https://doi.org/10.1088/2632-2153/abbd2d
25. Barenboim G, Hirn J, Sanz V (2021) Symmetry meets AI. CoRR arXiv:abs/2103.06115
26. Zhang Z, Zhang F, Chen H, Liu J, Wang H, Dai G (2014) Left and right hand distinction for multi-touch tabletop interactions. In: 19th International Conference on Intelligent User Interfaces,

IUI, pp 47–56. ACM, Haifa. https://doi.org/10.1145/2557500.2557525
27. Indyk P, Motwani R (1998) Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, pp 604–613. ACM, Dallas. https://doi.org/10.1145/276698.276876
28. Shen Y, Qin J, Chen J, Yu M, Liu L, Zhu F, Shen F, Shao L (2020) Auto-encoding twin-bottleneck hashing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, pp 2815–2824. Computer Vision Foundation / IEEE, Seattle. https://doi.org/10.1109/CVPR42600.2020.00289
29. Ng T, Balntas V, Tian Y, Mikolajczyk K (2020) SOLAR: second-order loss and attention for image retrieval. In: Computer Vision - ECCV 2020 - 16th European Conference. Lecture Notes in Computer Science, vol. 12370, pp 253–270. Springer, Glasgow. https://doi.org/10.1007/978-3-030-58595-2_16
30. Feng J, Karaman S, Chang S (2017) Deep image set hashing. In: 2017 IEEE Winter Conference on Applications of Computer Vision, WACV, pp 1241–1250. IEEE Computer Society, Santa Rosa. https://doi.org/10.1109/WACV.2017.143
31. Lin K, Lu J, Chen C, Zhou J (2016) Learning compact binary descriptors with unsupervised deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp 1183–1192. IEEE Computer Society, Las Vegas. https://doi.org/10.1109/CVPR.2016.133
32. Li Y, Wang Y, Miao Z, Wang J, Zhang R (2020) Contrastive self-supervised hashing with dual pseudo agreement. IEEE Access 8:165034–165043. https://doi.org/10.1109/ACCESS.2020.3022672
33. Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp 248–255. IEEE Computer Society, Miami. https://doi.org/10.1109/CVPR.2009.5206848
34. You Y, Gitman I, Ginsburg B (2017) Scaling SGD batch size to 32k for imagenet training. CoRR arXiv:abs/1708.03888
35. Adam Paszke FM Sam Gross, et al. (2019) Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, pp 8024–8035. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html