

# Cross-Scale Context Extracted Hashing for Fine-Grained Image Binary Encoding

Xuetong Xue \*

XUEXUETONG@CORP.NETEASE.COM

Jiaying Shi \*

SHIJIAYING@CORP.NETEASE.COM

Xinxue He

HEXINXUE@CORP.NETEASE.COM

Shenghui Xu †

XUSHENGHUI@CORP.NETEASE.COM

Zhaoming Pan

PANZHAOMING@CORP.NETEASE.COM

*No.7 Building, Zhongguancun Software Park West, No.10 Xibeiwang East RD, Beijing, China*

**Editors:** Emtiyaz Khan and Mehmet Gönen

## Abstract

Deep hashing has been widely applied to large-scale image retrieval tasks owing to efficient computation and low storage cost by encoding high-dimensional image data into binary codes. Since binary codes do not contain as much information as float features, the essence of binary encoding is preserving the main context to guarantee retrieval quality. However, the existing hashing methods have great limitations on suppressing redundant background information and accurately encoding from Euclidean space to Hamming space by a simple sign function. In order to solve these problems, a Cross-Scale Context Extracted Hashing Network (CSCE-Net) is proposed in this paper. Firstly, we design a two-branch framework to capture fine-grained local information while maintaining high-level global semantic information. Besides, Attention guided Information Extraction module (AIE) is introduced between two branches, which suppresses areas of low context information cooperated with global sliding windows. Unlike previous methods, our CSCE-Net learns a content-related Dynamic Sign Function (DSF) to replace the original simple sign function. Therefore, the proposed CSCE-Net is context-sensitive and able to perform well on accurate image binary encoding. We further demonstrate that our CSCE-Net is superior to the existing hashing methods, which improves retrieval performance on standard benchmarks.

**Keywords:** Hashing algorithms, Image retrieval, Cross-scale, Binary encoding

## 1. Introduction

Hash encoding refers to compressing high-dimensional image pixels or feature points into binary codes instead of continuous features Wang et al. (2015). Due to its fast retrieval speed and low storage cost, deep hashing methods Lai et al. (2015); Shen et al. (2015); Li et al. (2015); Liu et al. (2016); Xia et al. (2014) have attracted lots of attention and been applied in the large-scale image and video retrieval to search similar ones from millions, which is quite common in today’s world Liong et al. (2016). Recently, with the help of CNN’s powerful representation capability, deep hashing methods Yuan et al. (2020); Yang et al. (2021) have shown excellent performance on benchmarks. The existing methods often

---

\*. These authors contributed equally to this work.

†. Corresponding author.

§. Code: <https://github.com/NetEase-Media/CSCE-Net>

learn to utilize pairwise or triplet data similarity to encode images. In order to retain the discrimination of the original data, the concepts of “Hash Center” Yuan et al. (2020) and “Proxy” Wicczorek et al. (2021) are proposed, which can reduce the computational complexity and simplify the problem to optimize the intra-class distance only by finding a set of orthogonal vectors as hash centers.

The most existing deep hashing algorithm Liu et al. (2016); Hoe et al. (2021); Yuan et al. (2020) firstly obtains continuous float features through deep network’s (eg. Alexnet Krizhevsky et al. (2012) or ResNet50 He et al. (2016)) last fully connect layer before classification layer, then computes binary codes (0, 1) by a simple sign function as post-processing. Therefore, network can be optimized by pairwise or triplet based loss Cao et al. (2017); Zhu et al. (2016); Li et al. (2015) as classification tasks or cosine similarity loss as metric learning tasks. With the emergence of deeper and more complex networks, such as Transformer Vaswani et al. (2017), deep learning begins to pay attention to finding stronger feature representations. For example, Balntas et al. (2016); Noh et al. (2017a); Revaud et al. (2019); Yang et al. (2021) introduced discriminated local features to guide the network to classify difficult samples through fine local information.

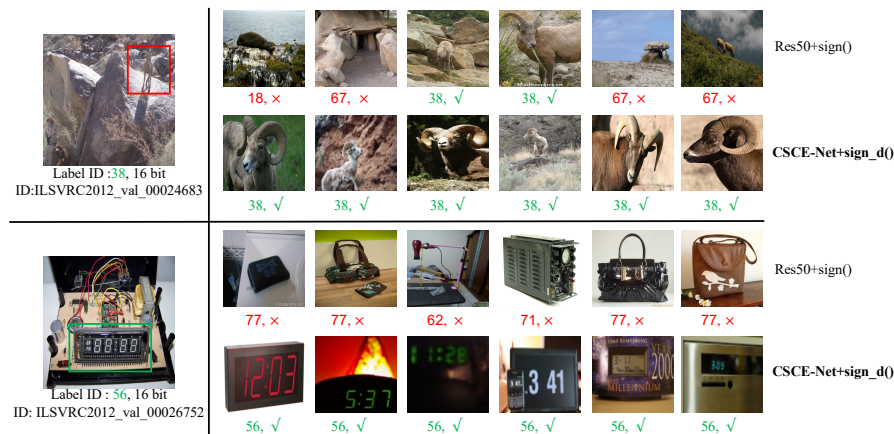


Figure 1: Visualization of two cases for *standard* hashing method and proposed CSCE-Net. Simple *CNN + sign()* methods encode entire image with no difference resulting in error recalls with similar background, for example label 18, 67 in case one. In contrast, the proposed CSCE-Net can encode key subject rather than background accurately.

Although these approaches attempted to make network learning a more compact representation, they ignored the relatively weak expressiveness of binary code. Hence, one challenge that needs to be addressed is how to maintain the most important information in binary codes from original float features. Besides, Fig. 1 shows two example cases, where the pure high-level semantic global representation learned by deep network is prone to recall error samples with unapparent targets (red box) or apparent targets (green box) with complex backgrounds. That is to say, network does not give enough representation for important targets when encoding images. Therefore, another challenge is how to make the

network automatically extract the most critical semantic context contained in images to produce better representations.

In this paper, we propose a Cross-Scale Context Extracted hashing Network (CSCE-Net) to extract the relative significant context of targets for image retrieval, which consists of two branches, aiming to jointly align global semantics and local details. Cooperated with sliding windows, an Attention guided Information Extraction module (AIE) is introduced between two branches to suppress low information areas by utilizing multi-scale semantic global context to filter fine-grained local information. Besides, instead of applying a simple sign function, CSCE-Net learns a content-related Dynamic Sign Function (DSF) to further ensure the accuracy of binary encoding. Generally speaking, our main contributions are as follows:

- We first consider the representation capability of binary code and propose CSCE-Net to automatically extract critical image information.
- An Attention guided Information Extraction module (AIE) is designed to further utilize global semantics to extract fine-grained local information, which improves the quality of handling unapparent targets and complex backgrounds.
- A content-related Dynamic Sign Function module (DSF) is designed to encode continuous features into binary codes, which reduces information loss adaptively.
- We conduct extensive experiments to demonstrate that the proposed CSCE-Net outperforms existing methods and achieves remarkable state-of-the-art performance on three public benchmarks.

## 2. Related Work

The deep hashing methods have been activately researched, such as CNNH [Xia et al. \(2014\)](#), DNNH [Lai et al. \(2015\)](#), DHN [Zhu et al. \(2016\)](#), HashNet [Cao et al. \(2017\)](#), DCH [Cao et al. \(2018\)](#), CSQ [Yuan et al. \(2020\)](#), DOLG [Yang et al. \(2021\)](#). Among these related works, the early methods usually adopt pairwise or triplet similarity learning to encode images. CNNH [Xia et al. \(2014\)](#), a two-stage method, decomposed the sample similarity matrix to obtain the binary code of samples, then used CNN to fit the binary codes. Furthermore, the deep architecture proposed by DNNH [Lai et al. \(2015\)](#) used a triplet ranking loss to preserve relative similarities which converted the input images into unified image representation and then encoded them into hash codes by divide-and-encode modules. DHN [Zhu et al. \(2016\)](#) simultaneously optimized the pairwise cross-entropy loss on semantic similarity pairs and the pairwise quantization loss on compact hash codes. HashNet [Cao et al. \(2017\)](#) generated binary hash codes by optimizing a novel weighted pairwise cross-entropy loss function in deep convolutional neural networks. However, due to the pair-wise data limitation, these pair-wise or triplet methods have insufficient coverage for data distribution and low efficiency across the entire training dataset.

To address the above limitations, CSQ [Yuan et al. \(2020\)](#) proposed a novel concept “Hash Center” and formulated the central similarity for deep hash learning which transformed hash code similarity learning to classification tasks. [Hoe et al. \(2021\)](#) unified training

objectives of deep hashing under a single classification objective by maximizing the cosine similarity between the continuous codes and binary orthogonal target under a cross-entropy loss. Besides, with the same basic idea of maximizing feature similarity, CosFace Wang et al. (2018) and Arcface Deng et al. (2019) are also mentioned in hashing tasks. The concept of above methods is to learn the high-level semantic features of images through deep networks, and then use a hash network to generate compact binary codes. However, for similar samples with much redundant background, the global representation obtained by deep learning cannot be accurately retrieved, requiring more refined local features for recognition.

Therefore, in recent years, several methods focus on how to represent an image with much richer information. Siméoni et al. (2019); Noh et al. (2017a); Cao et al. (2020) leveraged global features to select candidate images and then performed fine matching on candidate images according to local features. DELF Noh et al. (2017a) designed an attentive local feature descriptor and an attention mechanism for key point selection to identify usefully semantic local features for image retrieval. DELG Cao et al. (2020) proposed a unified model which leveraged generalized mean pooling to produce global features and attention-based key point detection to produce local features. DOLG Yang et al. (2021) attempted to fuse local and global features in an orthogonal manner for effective single-stage image retrieval.

Inspired by above concepts, the proposed method designs a two-branch feature extraction architecture among different scales. Interacted by an Attention guided Information Extraction module (AIE), our CSCE-Net aligns local and global spatial context to give enough representation for important targets when encoding images, which is contrary to DOLG Yang et al. (2021) focusing on feature fusion. Despite this, all the above methods use a sign function to encode features from Euclidean space to Hamming space, which leads to the quantization loss of binary code. Therefore, we consider the compactness between float and binary codes with a content-related Dynamic Sign Function to extend the ability from float features.

### 3. Method

In this paper, we propose a Cross-Scale Context Extracted Hashing Network (CSCE-Net), which is consist of an Attention guided Information Extraction module (AIE) and a content-related Dynamic Sign Function (DSF). The proposed framework is described in detail as follows:

#### 3.1. Problem Definition

As mentioned above, deep hash networks usually learn a compact continuous feature and then transform it into hash code by sign function. We define the deep hashing problems of  $N$  training samples and  $C$  categories as follows: we express the original  $D$ -dimensional image data as  $X = \{x_i\}_{i=1}^N \in \mathbb{R}^{N \times D}$  and denote the one-hot training corresponding semantic labels as  $Y = \{y_i\}_{i=1}^N \in \{0, 1\}^{N \times C}$ . For each training sample  $(x_i, y_i)$ , we express the compact continuous features generated by the deep network as  $f_{e_i}$ , and represent the  $K$ -bit binary codes  $b_i \in \{-1, 1\}^{N \times K}$  through sign function as shown in Fig. 2. Following previous work in CSQ Yuan et al. (2020) and ICS Zhang et al. (2021), our model also adopts the hash center  $\mathcal{H} = \{h_i\}_{i=1}^C \in \{-1, 1\}^{C \times K}$  to convert  $C$  class labels to the corresponding Hadamard

matrix rows with  $K$ -bit, which guarantees the orthogonality of any two class centers and facilitates high precision retrieval as previous work has proven.

The hash center can be generated by  $2k$ -order Hadamard matrix  $H$  simply using the Sylvester’s algorithm [Sylvester \(1867\)](#):

$$H_2^k = \begin{bmatrix} H_2^{k-1} & H_2^{k-1} \\ H_2^{k-1} & -H_2^{k-1} \end{bmatrix} = H_2 \otimes H_2^{k-1}, H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (1)$$

where  $\otimes$  represents the Kronecker product,  $2^k$  is the number of orthogonal centers.

### 3.2. Overview of CSCE-Net

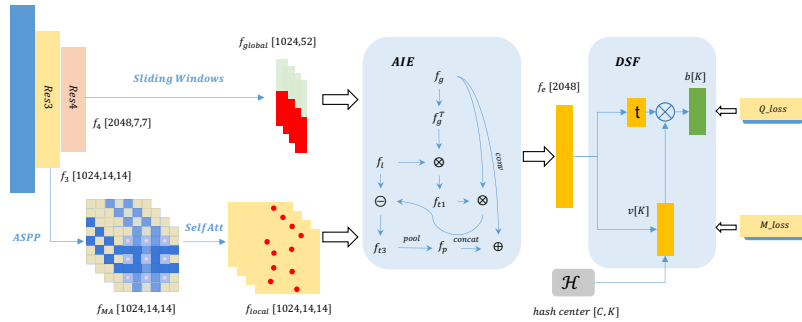


Figure 2: Overall structure of proposed CSCE-Net, built upon ResNet50. The global branch takes the output of *Res4* block as input to get multi-scale features  $f_{global}$ . The local branch uses ASPP and SelfAtt to model more local details  $f_{local}$  after *Res3* block. AIE takes both as input to generate final representation  $f_e$ .  $Q\_loss$  and  $M\_loss$  are quantization loss and metric loss respectively.

In order to solve the problem that the previous network does not give enough representation to unapparent targets when coding images, a two-branch framework CSCE-Net is designed. Following [Zhu et al. \(2016\)](#); [Cao et al. \(2017\)](#); [Yuan et al. \(2020\)](#), it is built upon state-of-the-art image recognition model ResNet50. As shown in Fig. 2: 1) The global branch keep the same as the original ResNet50 except that we remove all *pooling* and *fc* layers after *Res4* block  $f_4 \in R^{c_4 \times h \times w}$  to keep spatial information in feature maps, and the multi-scale sliding windows are introduced to get multi-scale spatial global information; 2) The major building blocks of our local branch (start from *Res3* block  $f_3 \in R^{c_3 \times h \times w}$ ) is based on ASPP [Chen et al. \(2017\)](#) and self-attention mechanism [Noh et al. \(2017b\)](#). Then AIE module is used to further model the semantics of global window features while helping to extract fine-grained local feature. The global features  $f_{global}$  abbreviated as  $f_g$  can be formulated as:

$$f_g = concatenate(torch.unfold(f_{4_1}, sw1), torch.unfold(f_{4_2}, sw2)), \quad (2)$$

where  $f_{4_1}$  and  $f_{4_2}$  have different feature size after *Conv2d* as shown in Fig. 3, and  $sw1$  and  $sw2$  are the hyper parameters of window size, which we use  $2 \times 2$  and  $4 \times 4$  respectively.

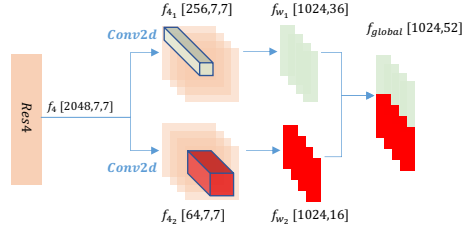


Figure 3: The structure of sliding window in global branch based on *Res4* block with two different window sizes.

Specifically, for the local branch, a multi-atrous dilated convolution layer is introduced in ASPP to handle scale variations among different image instances, which outputs feature maps with the different spatial receptive fields. Then a self-attention module can model the importance of each feature point in spatial. As is known to all, the shallow layers of the deep network focus on detailed information such as edges and corners, while the high layers perceive more overall and semantic information. Thus, the intuition behind two-branch design is to take advantage of the semantic context of *Res4* to align and filter better local features to suppress interference of noise. In terms of local structure, ASPP module contains three dilated convolution layers to obtain  $f_{MA}$  with different spatial receptive fields, then we compute  $f_{local}$  with an attention map produced by a  $1 \times 1$  convolution layer and the SoftPlus operation. Each feature size is shown in Fig. 2. Additionally, the proposed CSCE-Net keeps spatial information in both branches to align context adaptively which would be mentioned next.

### 3.3. Attention based Information Extraction Module, AIE

Existing hashing methods often process features on the global branch, by *pooling* or *fc* layer, which results in information loss on spatial. As shown in Fig. 1, simple *CNN + sign()* lacks the ability to separate redundant context from images, especially for unapparent targets. Therefore, binary code inevitably encodes a lot of unnecessary information. In order to effectively capture important areas in spatial, an Attention guided Information Extraction module (AIE) follows after multi-atrous ASPP and *Sel fAtt*. The structure of AIE is shown in Fig. 4.

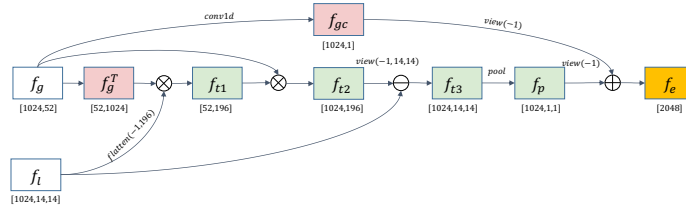


Figure 4: Detailed structure of Attention guided Information Extraction module, AIE

The inputs of AIE include two parts: sliding windows based multi-scale global features  $f_{global}$  abbreviated as  $f_g$  and attention based multi-atrous local features  $f_{local}$  abbreviated as

$f_l$ . First, through dot product operations (*torch.mm*), we align two features and compute the repetitive context from global to local which can be formulated as:  $f_{t2} = f_g \otimes (f_g^T \otimes f_l)$ . Because of the fact that global features have more semantic context as well as local features have more detailed context, for unapparent targets in complex background, the repetitive context would contain much more useless semantics. Therefore, it is necessary to separate them from local details, so secondly, we remove redundant features using *minus* to extract fine-grained local feature with multi-scale spatial context from global information. The whole process can be formulated as:

$$f_{t3} = f_l - f_g \otimes (f_g^T \otimes f_l). \quad (3)$$

Then, we obtain extracted ‘‘critical’’ information  $f_{t3}$  from correlation modeling, and the final extraction feature  $f_e$  is concated with global branch:

$$f_e = \text{concat}(\text{conv1d}(f_g).\text{view}(-1), \text{pool}(f_{t3}).\text{view}(-1)). \quad (4)$$

In this way, the final extracted  $f_e$  aligns the detailed local features with the global spatial context to focus on critical instance areas.

### 3.4. Content-related Dynamic Sign Function, DSF

As shown in Formula 5, the common binarization method sign function jumps around 0, which is equivalent to having a fixed threshold value of 0. The threshold cannot be changed according to the image content.

$$\text{sign}(x) = \begin{cases} 1, x > 0 \\ -1, \text{else} \end{cases}, \quad (5)$$

where we use 1 and -1 to represent binary code which also can be 1 and 0. Therefore, we propose a hypothesis that if the threshold can be changed according to the content of the image, images which have similar float features in original space will have a higher chance of being closer in Hamming space. Therefore, we propose content-related Dynamic Sign Function namely DSF to learn a dynamic threshold for *sign*().

Although it is better for hash code to have a balanced bit of being +1 or -1 [Hoe et al. \(2021\)](#), there is no guarantee that encoded hash code through *sign* function also maintains this balance. Therefore, we propose a learnable threshold  $t(t \geq 0)$ , and dynamically determine the direction of the jump to -1 or 1. The threshold  $t$  is proposed by a *fc* layer with output size of 1 shown in ‘‘DSF’’ in Fig. 2. For more details, according to learned  $t$ , we set an interval  $[-t, t]$ , then count the number of 1 and -1 outside the interval to dynamically encode the value inside  $[-t, t]$  to -1 or 1 for code balance. This rule ensures that *sign*() function maintains balance as much as possible. The definition of the new dynamic *sign* function is as follows:

$$\text{sign}_d(x) = \begin{cases} 1, x > t \\ \text{dynamic}, -t \leq x \leq t \\ -1, x < -t \end{cases}, \quad (6)$$

$$\text{dynamic} = \begin{cases} -1, \frac{\text{count}(x=1)}{\text{count}(x=-1)} > 1 \\ 1, \text{else} \end{cases}, \quad (7)$$

when the number of values bigger than  $t$  is more than the number of values smaller than  $-t$ , all inside interval  $[-t, t]$  will jump to -1, and vice versa. As shown in Fig. 2, the threshold  $t$  is learnt from extracted hash feature, which is content-related and context-sensitive. Note that for more stable training, the upper bound of  $t$  was set to 0.005 as a hyper-parameter.

### 3.5. Loss Functions for Image Hashing

**Metric Loss.** Generally speaking, for any two  $K$ -dimension binary codes  $b_1, b_2$ , the Hamming distance estimates the similarity of the two binary codes by XOR operation per bit. As we all know that XOR is non-differentiable, which cannot be used for training networks by back propagation algorithm Rumelhart et al. (1986). Therefore, cosine similarity is adopted instead to approximate the distance between continuous float code  $v_1, v_2$  for calculating the loss. Besides, the binary code can be computed by *sign* function from continuous code, which is expressed in the formula as:

$$b_i = \text{sign}_d(v_i), \quad (8)$$

where *sign<sub>d</sub>* function can be formulated as 6. Therefore, the relationship between cosine similarity and Hamming distance is formulated as Xu et al. (2021):

$$\mathfrak{D}_{ham}(b_i, b_j) = \frac{K}{2} \left(1 - \frac{b_i \cdot b_j}{\|b_i\| \cdot \|b_j\|}\right) \approx -\cos(v_i, v_j), \quad (9)$$

where  $K$  is the dimension of hash code. As is known, metric loss aims at clustering features by pushing different category samples as far as possible and pulling samples of the same category as close as possible. Following Hoe et al. (2021), Deng et al. (2019), Wang et al. (2018), the training of our CSCE-Net is also based on hash center and angular margin penalty-based cosine softmax loss (metric loss). The continuous code  $v_i$  also called logits can be reformulated as:  $v_i = f_{e_i} \cdot h_i^T = \|f_{e_i}\| \|h_i\| \cos(\theta_{h_i})$ , where  $\theta_{h_i}$  is the angle between the hash center  $h_i$  and the extracted hash feature  $f_{e_i}$  as shown in ‘‘DSF’’ part of Fig. 2. By the norm operation of extracted hash feature and hash center, we can get two equations,  $\|f_{e_i}\| = 1$  and  $\|h_i\| = 1$ . Therefore, it is explained that we can set  $\|v_i\|$  to a constant parameter  $S$ . The softmax cross-entropy loss is formulated as:

$$L_{ce} = \frac{1}{N} \sum_{i \in N} -\log \frac{e^{v_i}}{\sum_{j=1}^C e^{v_j}}, \quad (10)$$

where  $C$  is the category number. Taking the above formula of  $v_i$  into account and setting a constant bias, we can get a transformed formula for  $L_{ce}$ ,

$$L_{ce} = \frac{1}{N} \sum_{i \in N} -\log \frac{e^{f_{e_i} \cdot h_i^T + \text{bias}}}{\sum_{j=1}^C e^{f_{e_j} \cdot h_j^T + \text{bias}}} = \frac{1}{N} \sum_{i \in N} -\log \frac{e^{S \cdot \cos(\theta_{h_i})}}{e^{S \cdot (\cos(\theta_{h_i}))} + \sum_{j=1, j \neq i}^C e^{S \cdot \cos(\theta_{h_j})}}, \quad (11)$$

For simplicity, we fix bias offset equal to 0.  $\theta_{h_j}$  is the angle between extracted hash feature  $f_{e_i}$  and the other hash center  $h_j$  for negative computation. Therefore, the general angular margin penalty-based metric loss  $L$  is defined as follows:



$$L = \frac{1}{N} \sum_{i \in N} -\log \frac{e^{S \cdot (\cos(m_1 \cdot \theta_{h_i} + m_2) - m_3)}}{e^{S \cdot (\cos(m_1 \cdot \theta_{h_i} + m_2) - m_3)} + \sum_{j=1, j \neq i}^C e^{S \cdot \cos(\theta_{h_j})}}. \quad (12)$$

The angular margin penalty is different in SphereFace Liu et al. (2017) ( $m_1 = \alpha, m_2 = 0, m_3 = 0, \alpha > 1.0$ ), CosFace Wang et al. (2018) ( $m_1 = 1, m_2 = 0, m_3 = \alpha, 0 < \alpha < 1 - \cos(\frac{\pi}{4})$ ) and Arcface Deng et al. (2019) ( $m_1 = 1, m_2 = \alpha, m_3 = 0, 0 < \alpha < 1.0$ ). We use CosFace margin loss in this paper to train the whole CSCE-Net as formulated as:

$$M\_loss = \frac{1}{N} \sum_{i \in N} -\log \frac{e^{S \cdot (\cos(\theta_{h_i}) - \alpha)}}{e^{S \cdot (\cos(\theta_{h_i}) - \alpha)} + \sum_{j=1, j \neq i}^C e^{S \cdot \cos(\theta_{h_j})}}. \quad (13)$$

**Quantization Loss.** Metric loss only guarantees the continuous codes have favorable intra-class compactness and inter-class separability in original space. In order to mitigate the loss of information caused by *sign()* operation, quantization loss Zhu et al. (2016) is proposed to constrain latent codes. By adding quantization loss Li et al. (2017) Yuan et al. (2020) Zhe et al. (2019) Wang et al. (2021), retrieval performance is demonstrated to achieve promising improvement. Quantization loss *Q\_loss* can be formalized as:

$$Q\_loss = \frac{1}{N} \sum_{i=1}^N \|v_i - b_i\|_2^2 = \frac{1}{N} \sum_{i=1}^N \|v_i - \text{sign}_d(v_i)\|_2^2, \quad (14)$$

where  $v_i$  is continuous codes and  $b_i$  is corresponding binary codes formulated by Equation 8. By combining metric loss and quantified loss, our learning objective can be expressed as:

$$\min(M\_loss + \lambda Q\_loss), \quad (15)$$

where we use  $\lambda = 1$  to train CSCE-Net in an end-to-end manner.

## 4. Experiments

In this section, the proposed CSCE-Net would be evaluated and compared with several state-of-the-art hashing methods. We firstly introduce three popular datasets we used, namely ImageNet100, MS COCO, and NUS-WIDE, and then describe the evaluation protocols, implementation details, experiment comparisons, and ablation studies respectively.

### 4.1. Dataset

To verify the performance of our proposed method, we compare our method with the state-of-the-art methods by conducting extensive experiments on three widely-used benchmarks following prior works Cao et al. (2017, 2018); Yuan et al. (2020), which includes ImageNet100 Russakovsky et al. (2015), MS COCO Lin et al. (2014), and NUS-WIDE Chua et al. (2009). In terms of the hash center for classification, we generate each category center referring to CSQ Yuan et al. (2020) for both single-label and multi-label datasets.

**ImageNet100** is a large-scale benchmark for the visual recognition challenge, which contains over 1M images with a single label totally. For a fair comparison, we use the same data and settings as Cao et al. (2017); Yuan et al. (2020), containing 10K, 5K, 120K for training, testing and retrieval respectively in 100 categories.

**MS COCO** is a multi-label benchmark containing about 120K images belonging to 80 category types. Following [Cao et al. \(2017\)](#), after pruning images with no category information, we randomly sample 10K, 5K, 110K for training, testing and retrieval respectively.

**NUS-WIDE** is a public Web image dataset consisting of 269,648 multi-label images. Actually, this dataset contains 81 ground truth concept labels and we selected 21 most frequent categories following [Zhu et al. \(2016\)](#) to sample 10K, 2K, 140K for training, testing and retrieval respectively.

## 4.2. Evaluation Protocols and Implementation Details

We evaluate the retrieval performance based on the widely used metric mean average precision (mAP). For a fair comparison, we follow the prior methods [Zhu et al. \(2016\)](#); [Yuan et al. \(2020\)](#) that adopt mAP@1000 for ImageNet and mAP@5000 for the other datasets. The mAP is able to measure retrieval quality reliably which provides an average result of recall performance across all samples.

Following previous works, all the experiments of the proposed method in this paper are trained based on ResNet50 initialized from ImageNet pretrained weights and we fine-tune convolutional layers with proposed AIE and DSF modules through back propagation. As the proposed modules are trained from scratch, we use RMSProp optimizer with  $1e^{-5}$  initial learning rate and  $1e^{-5}$  weight decay factor. For the CosFace margin loss, we empirically set the margin  $\alpha$  as 0.15 and the CosFace scale  $S$  as 10. The images are first resized to  $256 \times 256$  resolution and then center cropped to 224. All experiments use batch size of 128 trained on a single A10 GPU with 24G memory for 100 epochs.

## 4.3. Experiment Results

### 4.3.1. COMPARISON WITH STATE-OF-THE-ART METHODS

Table 1: Comparison in mAP of Hamming Ranking for different bits on image retrieval.

Method	Backbone	ImageNet (mAP@1000)			MS COCO(mAP@5000)			NUS-WIDE (mAP@5000)		
		16bits	32bits	64bits	16bits	32bits	64bits	16bits	32bits	64bits
TransHash	ViT	0.785	0.873	0.892	-	-	-	0.726	0.739	0.749
CIBHash	VGG16	0.718	0.756	0.783	0.737	0.760	0.775	0.790	0.807	0.815
CIBHash	GCViT-XXT	0.860	0.886	0.897	0.807	0.836	0.851	0.795	0.817	0.825
CNNH	ResNet50	0.315	0.473	0.596	0.599	0.617	0.620	0.655	0.659	0.647
DNNH	ResNet50	0.353	0.522	0.610	0.644	0.651	0.647	0.703	0.738	0.754
DHN	ResNet50	0.367	0.522	0.627	0.719	0.731	0.745	0.712	0.759	0.771
HashNet	ResNet50	0.622	0.701	0.739	0.745	0.773	0.788	0.757	0.775	0.790
CSQ	ResNet50	0.851	0.865	0.873	0.796	0.838	0.861	0.810	0.825	0.839
<b>CSCE-Net</b>	ResNet50	0.869	0.887	0.897	0.807	0.852	0.888	0.794	0.827	0.839
<b>CSCE-Net</b>	VGG16	0.729	0.764	0.800	0.751	0.803	0.832	0.806	0.823	0.839
<b>CSCE-Net</b>	GCViT-XXT	0.896	0.910	0.911	0.844	0.902	0.919	0.839	0.865	0.878

The mean average precision (mAP) results comparing with state-of-the-art methods are shown in Table 1. The CNNH [Xia et al. \(2014\)](#), DNNH [Lai et al. \(2015\)](#), DHN [Zhu et al. \(2016\)](#), HashNet [Cao et al. \(2017\)](#), CSQ [Yuan et al. \(2020\)](#) and our proposed CSCE-Net are based on ResNet50 backbone; TransHash [Chen et al. \(2021\)](#), CIBHash [Qiu et al. \(2021\)](#) use the Vision Transformer (ViT) as backbone. From Table 1, it can be observed that our proposed CSCE-Net substantially outperforms all ResNet50 based comparison methods

even better than transformer-based TransHash by up 8.4% on ImageNet(16bits) and 14.4% on NUS-WIDE(64bit). Specifically, compared with the state-of-the-art ResNet50 based methods CSQ, our CSCE-Net achieves average absolute boosts of 1.5% in mAP for different bits on three benchmarks except for NUS-WIDE(16bit). Especially, the performance boost on ImageNet100 dataset is much larger than that on others, which is very impressive. Note that ImageNet100 has the most categories with variations among three datasets. Therefore, our CSCE-Net has the strong capacity for extracting critical local detailed information guided by semantic global context.

On the other hand, our ResNet50 based comparisons are conducted on both single-label and multi-label datasets. As shown in results for multi-label MS COCO and NUS-WIDE, the performance boost of the proposed CSCE-Net is as obvious as that on ImageNet100 except for NUS-WIDE(16bit). We achieve 1.1%, 1.4% and 2.7% higher mAP than the best competitor CSQ at 16-, 32- and 64-bit. In summary, our CSCE-Net performs well in most cases of different bits, which confirms the representation capability of binary code for image retrieval.

































Bits	Query	Top 10 retrieval results on ImageNet										
16 bit	 label: 10 ID: ILSVRC2012_val_00019045											CSCE-Net
		16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	16. ✓	CSQ
32 bit	 Label: 27 ID: ILSVRC2012_val_00046759											CSCE-Net
		46. x	27. ✓	27. ✓	27. ✓	27. ✓	97. ✓	27. ✓	27. ✓	27. ✓	27. ✓	CSQ
64 bit	 Label: 55 ID: ILSVRC2012_val_00005995											CSCE-Net
		55. ✓	64. x	55. ✓	55. ✓	55. ✓	64. x	55. ✓	55. ✓	55. ✓	55. ✓	CSQ

Figure 5: Examples of top 10 retrieved images for three bits on Imagenet-100 dataset. Below each image, we mark the corresponding label in green for accurate recall and red for wrong recall.

As shown in Fig. 5, each query has recalled a list of relative images, and our results are named by CSCE-Net. From the qualitative comparison, it is observed that the popular method CSQ and our CSCE-Net both can get accurate recalls in most cases. However, when the query image has a complex background or small main target like the first example of Fig. 5, our method performs much better than the other. Especially, the recall results of CSCE-Net also have redundant noise marked with green dashed boxes, which shows our method can deal with hard samples of unapparent targets in complex backgrounds. To validate this point, we visualize the last non-reduced convolutional layer of CSCE-Net using TorchCAM Fernandez (2020) shown in Fig. 6.

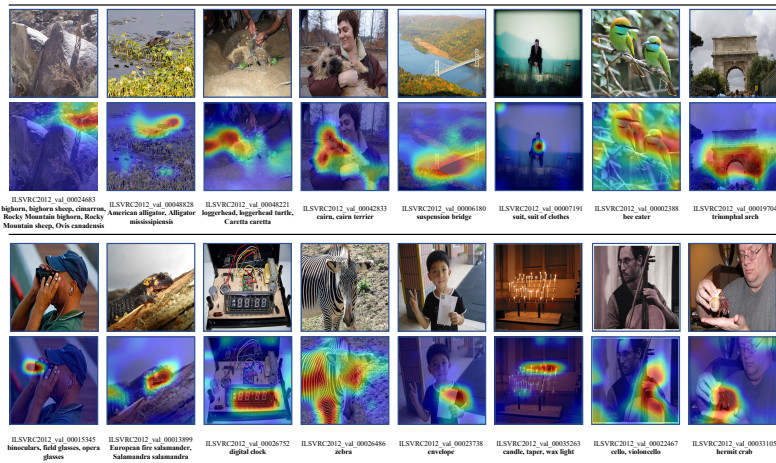


Figure 6: Visualization of class activation map of the last non-reduced convolutional layer in CSCE-Net. The image id and bolded **label** are at the bottom of each image.

#### 4.3.2. ABLATION STUDIES

**Analysis on the Effects of AIE.** We conduct ablation study experiments to analyze which part of the network benefits our proposed method mainly. As shown in Table 2, using the same loss as mentioned in Section 3.5, the baseline model is constructed on the 4-th layer of ResNet50, which represents the global context. Besides, we consume that the 3-rd layer feature namely “local” has more detailed information compared to the 4-th layer with a much bigger feature map. In terms of that, the proposed AIE aligns global semantics and local details to suppress redundant context, which evaluates the performance on the final row in Table 2. The second row of “Baseline + local” means the concatenation of local and global features shown as  $f_{local}$  and  $f_{global}$  in Fig. 2 to output continuous features. Compared with baseline result, it has an average 0.3% absolute improvement benefiting from local details.

In addition, neither using a single sliding window nor concating sliding window features with different scales as shown in the 3rd-5th rows of Table 2 has improved effectively. Compared with all conditions, Our AIE module achieves relative boosts of 3%, 1%, 1% in average mAP for each bit code on ImageNet100 and 7%, 3%, 2% on MS COCO dataset, which confirms that the detailed feature from local branch is aligned and filtered by global semantic context to focus on important instances area. Especially for 16 bit code, the proposed AIE module strongly improves performance with fewer hash values. Therefore, in the case where only less information can be retained, our model still has a strong representation of critical content.

**Analysis on the Effects of DSF and loss functions.** We have explained that the target of hash encoding is to cluster all samples in the same category. Besides, it is desirable that the output hash code is balanced with 0 and 1 as much as possible. Therefore, several experiments with different loss settings were conducted as shown in Table 3. “CE” means using a cross-entropy function to measure the loss between continuous features and

Table 2: Experimental results of CSCE-Net with variants structure.

Struct	ImageNet(mAP@1000)			MS COCO(mAP@5000)		
	16bits	32bits	64bits	16 bits	32bits	64bits
Baseline	0.839	0.879	0.889	0.752	0.829	0.865
Baseline + local	0.840	0.883	0.894	0.760	0.828	0.869
Baseline + global-sw2	0.856	0.868	0.888	0.739	0.802	0.851
Baseline + global-sw4	0.857	0.870	0.882	0.754	0.812	0.841
Baseline + global-sw2 + global-sw4	0.854	0.879	0.894	0.768	0.814	0.854
Baseline + AIE	<b>0.869</b>	<b>0.887</b>	<b>0.897</b>	<b>0.807</b>	<b>0.852</b>	<b>0.888</b>

Table 3: mAP comparison with variants loss of our method.

Loss	Imagenet100(mAP@1000)		
	16 bits	32bits	64bits
CE	0.845	0.877	0.887
CE+Qua	0.860	0.885	0.889
CF	0.860	0.877	0.891
CF+Qua	0.862	0.884	0.894
CF+DSF	<b>0.869</b>	<b>0.887</b>	<b>0.897</b>

ground truth binary codes. “CF” and “Qua” means CosFace loss and Quantization loss with the simple *sign* function we mentioned at Section 3.5 respectively. DSF is the proposed learnable sign function for quantization loss. From Table 3, we can see that the proposed CF+DSF algorithm achieves the best performance on the Imagenet100 dataset with the average mAP of 0.869, 0.887 and 0.897 respectively. As mentioned previously, the softmax cross-entropy loss performs well on classification, but is not optimal for feature learning due to its large intra-class variations. In contrast, the proposed method using cosine similarity to measure code distance is more adaptive. Besides, the comparisons of “no quantization loss”, “simple quantization loss” and “DSF quantization loss” shown in the last 3 rows of Table 3 draw a conclusion that it is necessary for reducing information loss through sign function. As for mAP results, our learnable sign function has a stable improvement.

## 5. Conclusion

In this paper, we consider the representation capability of binary code with a novel feature extraction module for deep hashing. The proposed Cross-Scale Context Extracted Hashing Network (CSCE-Net) can improve the quality of handling unapparent targets and complex backgrounds by utilizing global semantics to extract fine-grained local information. Moreover, we leverage the concept of code balance to inherit the stability from continuous float features to retrieval binary codes through a learnable dynamic sign function. Extensive experiments validated the efficiency of our method in both single-label and multi-label retrieval benchmarks. As part of the future work, we are exploring how to learn better feature representations for image retrieval in an unsupervised way.

## References

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3, 2016.

- Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *European Conference on Computer Vision*, pages 726–743. Springer, 2020.
- Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1229–1237, 2018.
- Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017.
- Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- Yongbiao Chen, Sheng Zhang, Fangxin Liu, Zhigang Chang, Mang Ye, and Zhengwei Qi. Transhash: Transformer-based hamming hashing for efficient image retrieval. *arXiv preprint arXiv:2105.01823*, 2021.
- Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–9, 2009.
- Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.
- François-Guillaume Fernandez. Torchcam: class activation explorer. <https://github.com/frgfm/torch-cam>, March 2020.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- Jiun Tian Hoe, Kam Woh Ng, Tianyu Zhang, Chee Seng Chan, Yi-Zhe Song, and Tao Xiang. One loss for all: Deep hashing with a single cosine similarity based learning objective. *Advances in Neural Information Processing Systems*, 34, 2021.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3270–3278, 2015.
- Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. Deep supervised discrete hashing. *Advances in neural information processing systems*, 30, 2017.

- Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep video hashing. *IEEE Transactions on Multimedia*, 19(6):1209–1219, 2016.
- Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016.
- Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.
- Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017a.
- Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017b.
- Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. *arXiv preprint arXiv:2105.06138*, 2021.
- Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised discrete hashing. *IEEE*, 2015.
- Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11651–11660, 2019.
- James Joseph Sylvester. Lx. thoughts on inverse orthogonal matrices, simultaneous signsuccessions, and tessellated pavements in two or more colours, with applications to newton’s rule, ornamental tile-work, and the theory of numbers. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):461–475, 1867.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.
- Jinpeng Wang, Bin Chen, Qiang Zhang, Zaiqiao Meng, Shangsong Liang, and Shutao Xia. Weakly supervised deep hyperspherical quantization for image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2755–2763, 2021.
- Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to hash for indexing big data—a survey. *Proceedings of the IEEE*, 104(1):34–57, 2015.
- Mikołaj Wiczcerek, Barbara Rychalska, and Jacek Dabrowski. On the unreasonable effectiveness of centroids in image retrieval. In *International Conference on Neural Information Processing*, pages 212–223. Springer, 2021.
- Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.
- Chengyin Xu, Zhengzhuo Xu, Zenghao Chai, Hongjia Li, Qiruyi Zuo, Lingyu Yang, and Chun Yuan. Hhf: Hashing-guided hinge function for deep hashing retrieval. *arXiv preprint arXiv:2112.02225*, 2021.
- Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuotong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11772–11781, 2021.
- Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3083–3092, 2020.
- Zhiwei Zhang, Allen Peng, and Hongsheng Li. Instance-weighted central similarity for multi-label image retrieval. *arXiv preprint arXiv:2108.05274*, 2021.
- Xuefei Zhe, Shifeng Chen, and Hong Yan. Deep class-wise hashing: Semantics-preserving hashing via class-wise loss. *IEEE transactions on neural networks and learning systems*, 31(5):1681–1695, 2019.
- Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 30, 2016.