SwinFGHash: Fine-grained Image Retrieval via Transformer-based Hashing Network

Di Lu¹ Id19@mails.tsinghua.edu.cn Jinpeng Wang¹ wjp20@mails.tsinghua.edu.cn Ziyun Zeng¹ zengzy21@mails.tsinghua.edu.cn Bin Chen^{†2} cb17@tsinghua.org.cn

Shudeng Wu¹ wsd19@mails.tsinghua.edu.cn

Shu-Tao Xia¹ xiast@sz.tsinghua.edu.cn

- ¹Tsinghua Shenzhen International Graduate School, Tsinghua University Shenzhen, China
- ² Harbin Institute of Technology Shenzhen, China

Abstract

Fine-grained image retrieval is a fundamental and challenging problem in computer vision due to the intra-class diversities and inter-class confusions. Existing hashingbased approaches employed convolutional neural networks (CNNs) to learn hash codes for fast fine-grained image retrieval, which are limited by the inherent locality constrain of the convolution operations and yield sub-optimal performance. Recently, transformers have shown colossal potential on vision tasks for their excellent capacity to capture longrange visual dependencies. Therefore, in this paper, we take the first step to exploit the vision transformer-based hashing network for fine-grained image retrieval. We propose the SwinFGHash, which takes advantage of transformer-based architecture to model the feature interactions among the spatially distant areas, e.g., the head and the tail of a bird on an image, thus improving the fine-grained discrimination of the generated hash codes. Besides, we enhance the critical region localization ability of SwinFGHash by designing a Global with Local (GwL) feature learning module, which preserves subtle yet discriminative features for fine-grained retrieval. Extensive experiments on benchmark datasets show that our SwinFGHash significantly outperforms existing state-of-the-art baselines in fine-grained image retrieval.

1 Introduction

Learning to hash has become an effective way for fast image retrieval [2], 2], which maps high-dimensional data into compact binary codes and efficiently computes pair-wise similarity with Hamming distance. With the development of deep learning, deep hashing methods [2, 3, 1], 13, 24, 5] equipped with Convolutional Neural Networks (CNNs) have

^{© 2021.} The copyright of this document resides with its authors.

It may be distributed unchanged freely in print or electronic forms.

[†] Corresponding author.

In this paper, we take the first step to exploit the vision transformer-based hashing network for fine-grained image retrieval. Specifically, we propose a novel hashing-based method termed *SwinFGHash*. SwinFGHash is based on Swin Transformer [**1**] as it achieves better performance at a lower complexity with its self-attention mechanism within shifted windows. SwinFGHash consists of three modules, including intermediate feature learning, global with local (GwL) feature learning, and hash code learning, as shown in Figure 1. The first two modules are used to learn fine-grained image representation. The original Swin Transformer uses average pooling along the spatial dimension in the last layer to aggregate features, which leads to the loss of some fine-grained subtle information. To better adapt to fine-grained hashing scenarios, we add the GwL feature learning module to capture those subtle yet discriminative local features. We concatenate the local features and the global features as the final representation. In the hash code learning module, we add an auxiliary classification loss to optimize the hashing codes learning globally in Hamming space.

Our contribution can be summarized as follows:

- We propose a novel fine-grained image hashing retrieval model, SwinFGHash, in conjunction with the transformer model. To the best of our knowledge, this is the first exploration of fine-grained hashing via the transformer.
- We design a global with local feature learning module to further enhance fine-grained retrieval by generating various local features.
- Extensive experiments have shown that SwinFGHash achieves state-of-the-art results on fine-grained benchmark datasets.

2 Related Works

Fine-Grained Image Hashing. Fine-Grained Image Retrieval (FGIR) aims at finding subcategory images belonging to a meta-category. The slight inter-class variance and significant intra-class variance make FGIR a challenging task. Learning to hash (L2H) is one of the most widely used Approximate Nearest Neighbor (ANN) search approach. There are two branches in L2H, namely quantization and binary hashing methods. The quantization methods [23, 29, 51, 53] divide the real data space into disjoint cells and the data points in each cell are approximately represented as the centroid. The hashing methods convert the data into Hamming space so that the distance can be quickly measured by bitwise operations. Currently, deep CNN-based hashing methods [2, 6, 51, 55] have achieved satisfactory performance on coarse-grained datasets, e.g., CIFAR-10, while the accuracy drops



Figure 1: Overview structure of our proposed SwinFGHash, which consists of three modules. (i) The intermediate feature learning module is used to learn obtain middle layer features from divided image patches. (ii) The global with local feature learning module is used to get both global and local features. (iii) The hash code learning module is used to generate binary hash codes. We define a learning objective consisting of hashing similarity part, classification part, and local feature distribution part, and we optimize the model by back-propagation.

Vision Transformer. Vision Transformer (ViT) [2] applies a pure transformer to a sequence of image patches and achieves state-of-the-art results in image classification. Since then, a flow of transformer-based models emerges and makes a remarkable breakthrough on various vision tasks [10, 6], [11], [12]. For instance, Swin Transformer [112] proposes a hierarchical structure while incorporating self-attention within shifted windows and improves accuracy on various tasks while reducing complexity. TransReID [112] utilizes side information and designs a novel jigsaw branch, surpassing previous works by a large margin on several object reidentification datasets. Nonetheless, the transformer has not yet been explored on FGIR. Our SwinFGHash is the first transformer-based model designed specifically under fine-grained image retrieval scenarios. The encouraging performance on several public benchmarks sheds light on the transformer application for FGIR.

3 Methodology

In this section, we describe the proposed SwinFGHash method in detail.

Problem Formulation. Given a fine-grained dataset consisting of N images that belong to N_c categories, i.e., $\{I_i, y_i\}_{i=1}^N$, where I_i denotes the *i*-th image and $y_i \in \{0, 1, \dots, N_c - 1\}$ denotes the label of the *i*-th image. For each image pair in the training set, we can construct a similarity matrix S, where $s_{ij} = 1$ when I_i has the same label as I_j ; and vice versa then $s_{ij} = 0$. The goal of deep hashing image retrieval is to learn a nonlinear hash function $\mathcal{F} : I_i \mapsto \{0, 1\}^B$ that takes the image I_i encoded as a binary hash code vector h_i of length B, and we expect the binary hash code vector to maintain the similarity relationship in matrix S. That is, if $s_{ij} = 1$, the Hamming distance between h_i and h_j should be small, while the Hamming distance should be large when $s_{ij} = 0$.

Model Overview. Our proposed model framework, SwinFGHash, is shown in Figure 1.



Figure 2: Swin Transformer Block

The network structure is based on the Swin Transformer [**D**] model, and the overall framework consists of three modules. (i) The intermediate feature learning module, which is used to learn global intermediate features. In this module, the input images are split into nonoverlapping image patches in the first. Each patch is treated as a token and fed into the stacked Swin Transformer blocks. (ii) The GwL feature learning module. We design this module to generate global and local intermediate features and feed them into a refinement network to obtain the final feature representation. (iii) The hash code learning module. This module goes from the feature representation to the binary hash code. The loss function of hash code learning consists of three parts: hashing similarity quantization loss, auxiliary classification loss, and local feature distribution loss.

3.1 Intermediate Feature Learning Module

The size of the fine-grained input image is $H \times W \times 3$. We partition the input images into nonoverlapping patches of 4×4 to form sequence embeddings. Each patch is considered a token, and by this partitioning method, the feature dimension of each patch becomes $4 \times 4 \times 3 = 48$. Afterward, a linear embedding layer is applied to project the feature dimension to an arbitrary size (denoted as *C*). The projected patch tokens are passed through several Swin Transformer blocks and patch merging layers to generate intermediate feature representations. Specifically, the patch merging layer is responsible for downsampling and increasing the dimensionality, and the Swin Transformer block is responsible for feature representation learning.

Unlike conventional multi-head self-attention (MSA) modules, Swin Transformer blocks are built based on shifted windows. As shown in Figure 2, each Swin Transformer block consists of LayerNorm layer, multi-head self-attention module, residual connection, and two-layer MLP. Two consecutive Swin Transformer blocks apply the window-based multi-head self-attention (W-MSA) module and the shifted-window-based multi-head self-attention (SW-MSA) module. The W-MSA module models the locality of the image by performing multi-head self-attention operations only inside the local window. In contrast, SW-MSA shifts the windows, thus enabling the interaction of information between different windows. Based on such window division mechanism, the consecutive Swin Transformer blocks can be formulated as following for image I_i :

$$\hat{E}_{i}^{l} = W - MSA(LN(E_{i}^{l-1})) + E_{i}^{l-1}$$
(1)

$$E_i^l = MLP(LN(\hat{E}_i^l)) + \hat{E}_i^l \tag{2}$$

$$\hat{E}_i^{l+1} = SW - MSA(LN(E_i^l)) + E_i^l$$
(3)

$$E_i^{l+1} = MLP(LN(E_i^{l+1})) + \hat{E}_i^{l+1}$$
(4)

The patch merging layer concatenates the features of each group of 2×2 neighboring

patches, which reduces the number of tokens to a quarter and increases the dimensionality of the features by a factor of 4. Then a linear embedding layer is applied to unify the feature dimension to the $2\times$ of the original size.

Our intermediate feature learning module can be viewed as a three-stage network. The number of patch tokens after projecting is $\frac{H}{4} \times \frac{W}{4}$, and the feature dimension is *C*. By feeding these tokens into several Swin Transformer blocks, followed by a patch merging layer, the number of tokens becomes $\frac{H}{8} \times \frac{W}{8}$ and the feature dimension becomes 2*C*, which is the first stage. By a similar operation, after the second stage, the number of tokens becomes $\frac{H}{16} \times \frac{W}{16}$ and the feature dimension becomes 4*C*. Finally, after the end of the third stage, the number of tokens becomes $\frac{H}{32} \times \frac{W}{32}$ and the feature dimension becomes 8*C*, which are the desired global intermediate features.

3.2 Global with Local Feature Learning Module

The subtle and discriminative parts of images are critical to the learning of fine-grained image features. To better capture these part features, we propose the GwL feature learning module. This module consists of a local saliency learning submodule and a refinement feature learning submodule.

The image I_i is fed into the intermediate feature learning module and the intermediate features obtained are represented as

$$E_i = f_{ibl}(I_i) \in \mathbb{R}^{L \times C_f} \tag{5}$$

where $L = \frac{H}{32} \times \frac{W}{32}$ is the number of tokens and $C_f = 8C$ is the dimension of the features.

The local saliency map learning submodule consists of 2 Swin Transformer blocks and a linear embedding layer. The linear embedding layer maps the features from C_f -dimensions to *K*-dimensions, where *K* is the number of local saliency maps. The local saliency maps are

$$A_i = FC(f_{lsm}(E_i)) \in \mathbb{R}^{L \times K}$$
(6)

where $l_{lsm}(\cdot)$ is the Swin Transformer blocks and $FC(\cdot)$ is linear embedding layer.

Then we obtain K local intermediate features by A_i and E_i

$$E_i^{local} = A_i \otimes E_i \in \mathbb{R}^{K \times L \times C_f} \tag{7}$$

where \otimes is a special operation as we first make A_i and E_i have the same dimension by *repeat* operation and then multiply them element by element. Now we have global intermediate features and local intermediate features.

The refinement feature learning submodule also consists of 2 Swin Transformer blocks, and the obtained output, after average pooling along the token's direction, gives the final feature representation. We denote the Swin Transformer blocks in this module as $f_{rfl}(\cdot)$ and average pooling layer as $Avg(\cdot)$. We first input the global intermediate features into the refinement feature learning module to obtain a global representation of the image

$$F_i^{global} = Avg(f_{rfl}(E_i)) \in \mathbb{R}^{C_f}$$
(8)

Then, we input the local intermediate features into the refinement feature learning module to obtain the local feature representation

$$F_i^{local} = Avg(f_{rfl}(E_i^{local})) \in \mathbb{R}^{K \times C_f}$$
(9)

We concatenate the local features and the global features to get the final feature representation

$$F_i^{GwL} = [F_i^{local}; F_i^{global}]_{cat} \in \mathbb{R}^{(K+1)C_f}$$
(10)

3.3 Hash Code Learning Module

The purpose of the hash code learning module is to convert the real-valued features into binary hash codes that maintain the similarity among the images in Hamming space. For each image I_i , we use a linear embedding layer as the hash layer $f_{hash}(\cdot)$ to map its final feature representation to *B*-dimension feature \hat{h}_i , i.e.,

$$\hat{h}_i = f_{hash}(F_i^{GwL}) \in \mathbb{R}^B \tag{11}$$

Then the *B*-bit binary hash codes h_i can be generated from the approximate hash representation by the $sign(\cdot)$ function: $h_i = sign(\hat{h}_i)$. Instead of directly optimizing discrete hash codes h_i , we choose to optimize the continuous approximate hash representation \hat{h}_i . We approximate the distance $d_H(h_i, h_j)$ with $d_H(\hat{h}_i, \hat{h}_j)$:

$$d_{H}(\hat{h}_{i},\hat{h}_{j}) = \frac{B}{2} \left(1 - \frac{\langle \hat{h}_{i},\hat{h}_{j} \rangle}{\|\hat{h}_{i}\|\|\hat{h}_{j}\|} \right)$$
(12)

where $\langle \cdot, \cdot \rangle$ denotes inner product and $\|\cdot\|$ is the length of a vector.

To maintain the similarity relation during the training of hashing codes, inspired by DCH [**D**], we design the similarity reconstruction loss consists of hash similarity loss \mathcal{L}_s and quantization loss \mathcal{L}_q :

$$\mathcal{L}_{s} = \sum_{s_{ij} \in S} w_{ij} \left(s_{ij} \log \frac{\mathrm{d}_{H}(\hat{h}_{i}, \hat{h}_{j})}{\gamma} + \log \left(1 + \frac{\gamma}{\mathrm{d}_{H}(\hat{h}_{i}, \hat{h}_{j})} \right) \right)$$
(13)

where γ is the scale parameter of the *Cauchy* distribution and γ controls the compactness of the distribution of the learned hash codes. Besides, weight w_{ij} is used to balance positive $(s_{ij} = 1)$ and negative sample pairs $(s_{ij} = 0)$. The quantization loss defined as follow is to ensure the output of approximate hashing representations to be close to 1 or -1.

$$\mathcal{L}_q = \sum_{i=1}^N \log\left(1 + \frac{\mathrm{d}_H(|h_i|, \mathbf{1})}{\gamma}\right) \tag{14}$$

To make good use of supervised information, we take the output of the hash layer to a classification layer that directs the learning of hashing representation by an auxiliary softmax cross-entropy loss \mathcal{L}_{cls} , i.e.,

$$\mathcal{L}_{cls} = \sum_{i=1}^{N} l_{ce}(y_i, \hat{y}_i), \tag{15}$$

where y_i denotes the real label of image I_i , while \hat{y}_i denotes the predicted label.

To generate diverse local features for fine-grained image retrieval, inspired by Exch-Net [\square], we design the local feature distribution loss \mathcal{L}_{lfd} based on the Hellinger distance:

$$\mathcal{L}_{lfd} = \max\left(t - \frac{1}{\sqrt{2}\binom{K}{2}}\sum_{k,l=1}^{K} \|\sqrt{p_i^k} - \sqrt{p_i^l}\|_2, 0\right)$$
(16)

where $t \in [0, 1]$ is a hyper-parameter to adjust the diversity, and $p_i^k = softmax(F_{i,k}^{local})$. We can obtain the overall loss \mathcal{L}_{TFGH} as following:

$$\mathcal{L}_{TFGH} = \mathcal{L}_s + \lambda_q \mathcal{L}_q + \lambda_{cls} \mathcal{L}_{cls} + \lambda_{lfd} \mathcal{L}_{lfd}$$
(17)

where λ_q , λ_{cls} , λ_{lfd} are hyper-parameters that adjust the overall loss.

4 Experiments

4.1 Datasets and Experimental Settings

Datasets. We conduct experiments on two widely used fine-grained benchmark datasets: CUB-200-2011 and Stanford Dogs . **CUB-200-2011** [26] is a fine-grained bird dataset, which consists of 11,788 images belonging to 200 categories. Following [15, 22, 54, 56], we adopt the official split strategy. We use the test set containing 5,794 images as the test query set, while the train set containing 5,994 is used to train the hash models and is also used as the retrieval database. **Stanford Dogs** [16] is a fine-grained dog dataset consisting of 20,580 images in 120 categories. Each category has approximately 150 images. We use official partitions according to [15, 12, 54], 56], and 100 images are selected from each category to form the training set, while the remaining 8,580 images are used as the test query set. Similar to CUB-200-2011, the training set is also used as a retrieval database.

Metrics and Baselines. To measure the performance of hashing, we use three evaluation metrics: the Mean Average Precision (mAP), the Precision-Recall (PR) curves, and the Precision curves w.r.t. different numbers of top-returned samples (P@N). We compare the retrieval performance of the proposed SwinFGHash with state-of-the-art baselines, including: 1) seven generic deep hashing methods, DSH [$[\basel{lmatrix}]$, DTSH [$[\basel{lmatrix}]$, HashNet [$\basel{lmatrix}]$, GreedyHash [$[\basel{lmatrix}]$, DCH [$\basel{lmatrix}]$, DPN [$[\basel{lmatrix}]$, and CSQ [$\basel{lmatrix}]$, 2) six fine-grained hashing methods, DSaH [$[\basel{lmatrix}]$, PWH [$\basel{lmatrix}]$, ExchNet [$\basel{lmatrix}]$, SRLH [$\basel{lmatrix}$] and CFH [$\basel{lmatrix}]$.

Experimetal Details. We implement SwinFGHash with PyTorch [23] and adopt Swin-Tiny [13] as the backbone. Because Swin-Tiny (29M parameters & 4.5 GFLOPS) has a comparable model capacity to ResNet-50 (26M parameters & 4.1 GFLOPS) [13], we adopt ResNet-50 as CNN backbone for those CNN-based methods for fair comparision. All backbones are pre-trained on ImageNet [3] and finetuned on fine-grained datasets. We adopt an AdamW [13] optimizer for 300 epochs with a weight decay of 5e-4 and gradient clipping with a max norm of 1 and fix the batch size as 64. We use a cosine decay learning rate scheduler with an initial learning rate of 2.5e-4 and 20 epochs of linear warm-up. All input images are resized to 224×224 and augmented with RandAugment [5] strategies in training. The intermediate feature learning module weights are initialized by the first three stages of Swin-Tiny pre-trained on ImageNet. The fourth stage initializes the Swin Transformer blocks of the GwL feature learning module. We set $\gamma = 20$, t = 0.4, and $\lambda_q = \lambda_{cls} = \lambda_{lfd} = 0.1$.

4.2 Experimental Results

We have tried to reproduce open-sourced methods, including the seven generic hashing methods. Nevertheless, it is more difficult to reproduce those six more complicated fine-grained hashing methods (marked with the superscript †), as the codes are not publicly available. We directly quote the results from the original papers [2, 15, 22, 52, 53, 56] only when the experimental settings keep consistent with ours, i.e., the same dataset with the same partition and

	CUB-200-2011				Stanford Dogs			
Method	16bits	32bits	48bits	64bits	16bits	32bits	48bits	64bits
HashNet	11.95	44.92	49.89	54.36	18.77	59.35	62.65	63.51
DSH	21.05	56.51	72.72	73.68	16.84	25.60	48.97	64.86
DTSH	69.35	71.98	72.70	73.56	66.81	70.57	72.44	73.86
DCH	67.17	71.10	72.84	75.25	74.33	75.41	76.21	79.14
GreedyHash	66.82	74.35	76.94	78.02	68.84	75.45	77.78	79.59
CSQ	71.03	74.72	-	77.01	72.25	74.81	-	77.57
DPN	73.83	75.23	77.85	78.25	73.97	75.50	76.53	77.92
DSaH [†]	14.08	28.17	34.28	43.13	39.76	52.83	59.50	64.52
FPH^\dagger	51.69	58.32	61.24	62.33	63.40	69.09	70.60	71.30
PWH^{\dagger}	-	62.79	69.56	-	-	-	-	-
ExchNet [†]	-	67.74	71.05	-	-	-	-	-
\mathbf{SRLH}^\dagger	69.08	69.22	70.87	70.10	66.68	74.02	75.16	75.38
CFH^{\dagger}	72.08	73.41	73.21	72.71	79.33	79.96	79.47	79.34
SwinFGHash(ours)	77.71	84.26	83.73	83.55	82.03	83.74	82.42	81.80

Table 1: mAP(%) Results on CUB-200-2011 & Stanford Dogs Dataset of Different Bits. † means we did not reproduce these methods, but we quote the results from original papers under the same settings. - indicates that the method does not support a specific number of bits or there are no results in the literature.

the same metrics. The mAP results on two datasets, CUB-200-2011 and Stanford Dogs, are shown in Table 1, which shows that our proposed SwinFGHash substantially outperforms all the methods. To be more specific, on the CUB-200-2011 dataset, we achieve a 6.02% improvement in average mAP compared with the current state-of-the-art method DPN. Similarly, on the Stanford Dogs dataset, SwinFGHash outperforms CFH, the best method among the compared methods, by 2.97% in terms of average mAP. The results of mAP show that our proposed SwinFGHash method successfully captures the discriminative part features while achieving good interaction between different part features to generate better binary hash codes, thus obtaining superior retrieval results.



(a) PR-CUB(b) P@200-CUB(c) PR-Dogs(d) P@200-DogsFigure 3: (a) (b) Precision-Recall curves & Precision@Top-N curves on the CUB-200-2011datasets with binary codes@32 bits.(c) (d) Precision-Recall curves & Precision@Top-Ncurves on the Stanford Dogs datasets with binary codes@32 bits.

Figure 3 shows the Precision-Recall curves and the Precision curves w.r.t. a different number of top returned items on CUB-200-2011 and Stanford Dogs, respectively, with binary codes at 32 bits. SwinFGHash essentially achieves higher precision than other baselines, especially in the case of a low recall. In addition, SwinFGHash consistently finds more relevant images in the returned top-N (N < 100) list, which is advantageous because

users will be more concerned with the top-ranked results in the retrieval system.

Backbone	GwL	16bits	32bits	48bits	64bits	-	K	32bits mAP(%)
Swin-Tiny	X	76.40	83.18	83.03	82.26		2	83.32
Swin-Tiny	\checkmark	77.71	84.26	83.73	83.55	_	4	84.26
Swin-Base	×	81.13	85.95	85.61	84.02		6	83.76
Swin-Base	\checkmark	84.05	87.40	87.11	84.92		8	83.61
		(a)				-		(b)

4.3 Ablation Study

Table 2: (a) mAP(%) Results of with/without GwL Feature Learning Module on CUB-200-2011 with Swin-Tiny & Swin-Base. K means replacing the GwL module. (b) mAP(%) Results of different Local Feature Number *K* on CUB-200-2011 with Swin-Tiny with binary codes@32 bits.

To validate the effectiveness of the proposed GwL feature learning module, we conducted experiments on the CUB-200-2011 dataset after replacing this module with the fourth stage of the Swin Transformer. As shown in Table 2a, despite the already relatively high baseline, the GwL module still improves the average mAP by about 1.10%. As shown in Table 2a, we also conducted experiments on a larger model, Swin-Base, to verify the scalability of our approach and the effectiveness of the GwL module on larger models. The results show that the retrieval performance further improves as the model capacity becomes larger, and the GwL module improves the average mAP on Swin-Base by about 1.69%. The result shows that our proposed GwL module can capture discriminative but subtle part features, thus improving retrieval performance on fine-grained datasets.

We use *K* to denote the number of local features in the GwL module. As shown in Table 2b, we have experimented with different *K* on the CUB-200-2011 dataset with a bit number of 32. As we can see, the mAP increases and then decreases as *K* increases, but the performance is always better than replacing the GwL module. This result suggests that the number of *K* is essential to the GwL module. A smaller *K* may not be sufficient to capture complete fine-grained information, while a larger *K* may be challenging to optimize. In all other experiments in this paper, we set K = 4 to get the best performance.

4.4 Qualitative Analysis

Because of the multi-head self-attention (MSA) mechanism with dynamic attention and global information fusion based on sequences, Transformer has a strong ability of modeling long-range dependencies $[\Box, \Box]$. Intuitively, the Transformer model is suitable for finegrained retrieval scenarios, as fine-grained feature extraction often requires the interaction of features at spatially distant locations, such as the head and tail of a bird.

The GradCAM++ [**D**] visualization results of the ResNet and the SwinFGHash on the CUB-200-2011 dataset are shown in Figure 4. For the proposed SwinFGHash, we separate the global branch and several local branches of the GwL module to show more clearly the role of each part. Comparing the result of ResNet with the SwinFGHash global branch, we find that SwinFGHash captures more global contextual information and more discriminative



Figure 4: GradCam++ [2] visualization of attention maps on CUB-200-2011 datasets. From left to right are the original image, ResNet result, GwL Global branch result, and the last four are GwL Local branch results.

parts by modeling long-range visual dependencies. We also find that the local branch of GwL focuses on different discriminative local parts, which is crucial for fine-grained retrieval tasks.

5 Conclusion

This paper proposes SwinFGHash, the first use of a vision transformer-based hashing network for fine-grained image retrieval. The self-attention mechanism gives Transformer a solid ability to model long-range dependencies, which is suitable for fine-grained retrieval scenarios. We use the first three stages of the Swin Transformer to extract intermediate features. Besides, we propose a global with local feature learning module to capture those subtle yet discriminative local features. Comprehensive experiments demonstrate the superiority of SwinFGHash over state-of-the-art methods. With the promising results achieved by SwinFGHash, we believe that the transformer-based models have significant potential on fine-grained image retrieval scenarios.

Acknowledgement

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204 and the National Natural Science Foundation of China under Grant 61771273.

References

- [1] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation, 2021.
- [2] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1229–1237, 2018. doi: 10.1109/CVPR.2018.00134.

- [3] 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 (ICCV)*, Oct 2017.
- [4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 839–847, 2018. doi: 10.1109/WACV.2018.00097.
- [5] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306, 2021.
- [6] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 3008–3017, 2020. doi: 10.1109/CVPRW50498.2020.00359.
- [7] Quan Cui, Qing Yuan Jiang, Xiu Shen Wei, Wu Jun Li, and Osamu Yoshie. Exchnet: A unified hashing network for large-scale fine-grained image retrieval. In *Computer Vision – ECCV 2020*, pages 189–205, 2020. doi: 10.1007/978-3-030-58580-8_12.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009. doi: 10.1109/CVPR.2009. 5206848.
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [10] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *CoRR*, abs/2102.05644, 2021.
- [11] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, and Chee Seng Chan. Deep polarized network for supervised learning of accurate binary hashing codes. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 825–831. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.
- [12] Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Yuille. Transfg: A transformer architecture for fine-grained recognition. *arXiv preprint arXiv:2103.07976*, 2021.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 45, pages 770–778. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90.

- [14] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *arXiv preprint arXiv:2102.04378*, 2021.
- [15] Sheng Jin, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, Lei Zhang, and Xiansheng Hua. Deep Saliency Hashing for Fine-Grained Retrieval. *IEEE Transactions on Image Processing (TIP)*, 29:5336–5351, 2020. ISSN 19410042. doi: 10.1109/TIP.2020.2971105.
- [16] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.
- [17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.
- [18] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2064–2072, 2016. doi: 10.1109/CVPR.2016.227.
- [19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *International Conference on Computer Vision (ICCV)*, 2021.
- [20] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4905–4913, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.
- [21] Xiao Luo, Chong Chen, Huasong Zhong, Hao Zhang, Minghua Deng, Jianqiang Huang, and Xiansheng Hua. A survey on deep hashing methods. *CoRR*, abs/2003.03369, 2020.
- [22] Lei Ma, Xuan Li, Yu Shi, Jinmeng Wu, and Yaozhong Zhang. Correlation filteringbased hashing for fine-grained image retrieval. *IEEE Signal Processing Letters*, 27: 2129–2133, 2020. doi: 10.1109/LSP.2020.3039755.
- [23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. 2019.
- [24] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In Advances in Neural Information Processing Systems, volume 31, 2018.
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

- [26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.
- [27] Jingdong Wang, Ting Zhang, jingkuan song, Nicu Sebe, and Heng Tao Shen. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790, 2018. doi: 10.1109/TPAMI.2017.2699960.
- [28] Jinpeng Wang, Bin Chen, Tao Dai, and Shu-Tao Xia. Webly supervised deep attentive quantization. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2250–2254. IEEE, 2021.
- [29] 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.
- [30] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. Contrastive quantization with code memory for unsupervised image retrieval. *arXiv preprint arXiv:2109.05205*, 2021.
- [31] Xiaofang Wang, Yi Shi, and Kris M. Kitani. Deep supervised hashing with triplet labels. In *Computer Vision – ACCV 2016*, pages 70–84, Cham, 2017. Springer International Publishing. ISBN 978-3-319-54181-5.
- [32] Yimu Wang, Xiu-Shen Wei, Bo Xue, and Lijun Zhang. Piecewise hashing: A deep hashing method for large-scale fine-grained search. In *Pattern Recognition and Computer Vision*, pages 432–444. Springer International Publishing, 2020. ISBN 978-3-030-60639-8.
- [33] Jiacheng Yang, Bin Chen, and Shu-Tao Xia. Mean-removed product quantization for large-scale image retrieval. *Neurocomputing*, 406:77–88, 2020.
- [34] Yifan Yang, Libing Geng, Hanjiang Lai, Yan Pan, and Jian Yin. Feature Pyramid Hashing. In Proceedings of the 2019 ACM International Conference on Multimedia Retrieval (ICMR), pages 114–122, 2019. ISBN 9781450367653. doi: 10.1145/3323873. 3325015.
- [35] 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.
- [36] Haien Zeng, Hanjiang Lai, and Jian Yin. Simultaneous Region Localization and Hash Coding for Fine-grained Image Retrieval. *arXiv preprint arXiv:1911.08028*, 2019.
- [37] Ziyun Zeng, Jinpeng Wang, Bin Chen, Tao Dai, and Shu-Tao Xia. Pyramid hybrid pooling quantization for efficient fine-grained image retrieval. *arXiv preprint arXiv:2109.05206*, 2021.
- [38] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person re-identification. CVPR, 2020.