ConceptHash: Interpretable Fine-Grained Hashing via Concept Discovery

Supplementary Material

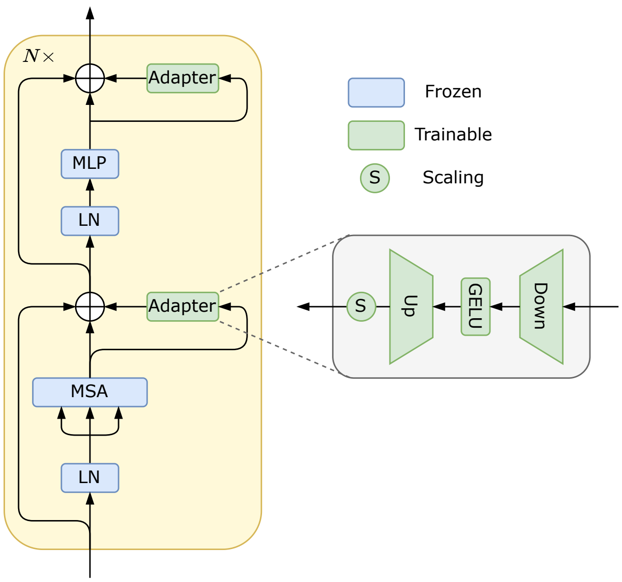

Figure 1. Two adapters are added after the multi-head self-attention layer (MSA) and the feedforward network (MLP). LN denotes layer normalization for each block of a standard vision transformer.

A. Implementation of adapter

To increase training efficiency, we add adapters to the vision transformer instead of fine-tuning all parameters. We adopt the architecture in AdaptFormer [3] and define our adapter as:

$$\text{adapter}(z) = s \cdot W_{\text{up}} \cdot \text{GELU}(W_{\text{down}} \cdot \text{LN}(z)), \quad (1)$$

where LN is a layer normalization layer [1], $W_{\text{down}} \in \mathbb{R}^{D_{\text{down}} \times D}$ is the weights of down projection and $W_{\text{up}} \in \mathbb{R}^{D \times D_{\text{down}}}$ is the weights of up projection, GELU is the non-linear activation function [6], and $s \in \mathbb{R}$ is a learnable scaling factor. D_{down} is set as 384.

We added two adapters for each block of the vision transformer, one after multi-head self-attention (MSA) layer and one after feedforward network (MLP). The output of l-th block of the vision transformer is computed as:

$$\begin{aligned}
\hat{Z}^{(l)} &= \text{MSA}(\text{LN}(Z^{(l-1)})), \\
\hat{\hat{Z}}^{(l)} &= \text{adapter}(\hat{Z}^{(l)}) + \hat{Z}^{(l)} + Z^{(l-1)}, \\
\widetilde{Z}^{(l)} &= \text{MLP}(\text{LN}(\hat{\hat{Z}}^{(l)})), \\
Z^{(l)} &= \text{adapter}(\widetilde{Z}^{(l)}) + \widetilde{Z}^{(l)} + \hat{\hat{Z}}^{(l)}. \quad (2)
\end{aligned}$$

Table 1. Performance (mean average precision) of retrieval by family species on CUB-200-2011.

	CUB-200-2011		
Methods	16	32	64
ITQ [5]	20.00	23.46	27.09
HashNet [2]	24.40	35.62	38.13
DTSH [11]	36.96	37.81	39.49
GreedyHash [10]	44.46	55.62	60.98
CSQ [13]	31.62	34.47	35.25
DPN [4]	34.09	36.28	36.84
OrthoHash [7]	34.16	36.95	37.61
A^2-Net [12]	45.62	50.93	52.95
SEMICON [9]	43.10	53.24	56.80
ConceptHash (Ours)	**60.54**	**63.44**	**67.20**

See Fig. 1 for the detail of the computational graph. The way we insert the adapters is also similar to [8].

B. Retrieval on family species

In this section, we evaluate the methods by replacing the fine-grained labels with family labels in order to assess the semantic ability of the hash codes. The CUB-200-2011 dataset is chosen as the benchmark. Table 1 presents two key observations: (i) Our ConceptHash outperforms previous methods by a significant margin, highlighting the effectiveness of our approach. This result underscores the superiority of our methods in capturing the semantic information encoded within the hash codes. (ii) Random-center-based hashing methods like CSQ [13] perform worse than older hashing methods such as DTSH [11], even though they outperform them in fine-grained retrieval (Table 1 in the main paper). A likely explanation is that the training objective of random-center-based hashing primarily focuses on learning to generate the fixed target hash codes, thereby ignoring the semantic relationships (such as family information) between the fine-grained classes.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. Hashnet: Deep learning to hash by continuation. In *International Conference on Computer Vision*, 2017. 1

[3] Shoufa Chen, Chongjian GE, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 1

[4] 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 *International Joint Conference on Artificial Intelligence*, 2020. 1

[5] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 1

[6] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[7] 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. In *Advances in Neural Information Processing Systems*, 2021. 1

[8] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019. 1

[9] Yang Shen, Xuhao Sun, Xiu-Shen Wei, Qing-Yuan Jiang, and Jian Yang. Semicon: A learning-to-hash solution for large-scale fine-grained image retrieval. In *European Conference on Computer Vision*. Springer, 2022. 1

[10] 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*, 2018. 1

[11] Xiaofang Wang, Yi Shi, and Kris M Kitani. Deep supervised hashing with triplet labels. In *Asian Conference on Computer Vision*, 2016. 1

[12] Xiu-Shen Wei, Yang Shen, Xuhao Sun, Han-Jia Ye, and Jian Yang. A^2-net: Learning attribute-aware hash codes for large-scale fine-grained image retrieval. In *Advances in Neural Information Processing Systems*, 2021. 1

[13] 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 *Computer Vision and Pattern Recognition*, 2020. 1