Devised Deephashing Models through Self-supervised Learning for Image Retrieval

Sooin Kim
Dept. of Software Science
Dankook University
Yongin, South Korea
32190756@dankook.ac.kr

Youngjin Jeon
Research Center
AiDeep
Seoul, South Korea
yjjeon@aideep.ai

Won-gyum Kim
Research Center
AiDeep
Seoul, South Korea
wgkim@aideep.ai

Doosung Hwang
Dept. of Software Science
Dankook University
Yongin, South Korea
dshwang@dankook.ac.kr

Donghoon Kim
Dept. of Computer Science
Arkansas State University
Jonesboro, AR, USA
dhkim@astate.edu

***Abstract*—Many image database systems utilize content-based retrieval technology using perceptual hashing to facilitate image comparison and retrieval. However, conventional hashing functions generate different hash codes for the same image domain even with minor changes, making accurate searches in the image database impossible. To overcome these limitations, a method has been devised to enable the search for images similar to queries synthesizing visual feature information, including texture, shape, and color of images. This paper proposes deephashing models that incorporate both variational autoencoders and vision transformers, capable of capturing semantic patterns and preserving the semantic similarity between data points. To validate this proposal, we conduct a comparative evaluation by comparing its performance with widely-used supervised learning-based models. The experimental analysis shows that the proposed approach generates compressed and discriminative hash codes with fixed-length binary representations that maintain semantic similarity and enable the search even for similar images. When evaluating the performance of our proposed model in comparison to the other studies, noticeable improvements are observed across the selected metrics in deephashing based on self-supervised learning.**

***Index Terms*—deep learning, image retrieval, perceptual hashing, self-supervised learning, attention layer**

I. Introduction

To build a similar image retrieval system, it is essential to fulfill the requirements of search accuracy and time-space limitations for retrieval. Hash codes are a useful tool for swiftly and effectively retrieving data, including images and videos, and other forms of content. Perceptual hashing is a technique that enables the comparison and identification of similar images by learning distinctive features that represent the perceptual characteristics of images [1]. This method has a range of practical applications, including image identification, image and video retrieval, and digital watermarking. However, image manipulation presents a significant challenge for ensuring multimedia authentication and security [2], [3]. Image editing software facilitates operations like color correction, object modification, and duplication, emphasizing the need for a digital image protection system.

A conventional hashing method converts high-dimensional input data into a low-dimensional hash code of fixed length. Deephashing is a technique that involves training a deep neural network to learn a hash function capable of mapping high-dimensional input data into a low-dimensional binary code. However, the existing hashing function produces different hash codes for the same image domain, even with minor alterations like jpeg compression, making it impossible to calculate the distance between query and image hash codes to search for similar images in the database [4]. To overcome these limitations, a new approach has been developed that can search for images similar to queries by synthesizing visual feature information, including texture, shape, and color of images [5].

Supervised deephashing methods incorporate label in the training data, allowing for the preservation of pairwise similarity relationships between labeled data through the learning of a mapping function from the input space to the hash space [6]. Conversely, self-supervised deephashing involves learning to map hash codes from unlabeled data, with the aim of capturing semantic patterns that reveal the structure of data and generate simple binary codes that maintain similarities between data points [6], [7]. Autoencoders are a common approach to self-supervised deephashing, as they can reconstruct input data from a compressed representation [8].

This paper proposes deephashing models with variational autoencoder (VAE [9]) or vision transformer (ViT [10]). Figure 1 depicts an image retrieval framework with deephashing. An comprehensive evaluation is conducted to measure their performance in comparison to established models based on supervised learning. The key contributions are described below:

- This study presents an architecture that effectively extracts visual features from images by enhancing CNNs for deephashing through self-supervised learning.
- This study demonstrates that self-supervised learning of VAE, VTE and ViT models enables unique key mapping, generating hash codes without redefining the loss function for hash vector modification.
- The proposed method can generate small and efficient hash codes with a fixed and lower-dimensional hashing vector that can be used to quickly retrieve similar data points.
- The experimental analysis shows that the hash code

This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency(Project Number: 2021-EC-9500).

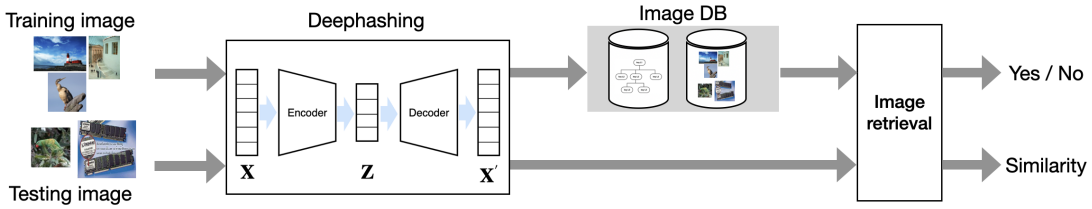

Fig. 1: A deephashing framework for image retrieval.

generated by the proposed method can map the expressions that preserve semantic similarity of input images, and provides high generalization performance just by changing the model structure.

This paper is organized as follows. Section II discusses related work on deephashing technology. Section III proposes various deephashing methods with detailed explanations for detecting database images. Section IV discusses the experimental results and Section V summarizes this work towards future directions.

II. Related works

The deephashing has been studied through supervised and unsupervised hashing techniques, depending on whether the dataset includes labels or not. Venkateswara *et al.* [7] proposed a supervised deephashing model that can extract semantic hash codes for images. Their dataset consisted of labeled and unlabeled images of various objects. The learning algorithm employed the VGG-F [11] network as a backbone, with two output layers configured to extract hash codes and labels. Dubey *et al.* [12] designed a model for image retrieval, called Vision Transformer (ViT) [13]-based hashing (VTS). They utilized a pre-trained ViT architecture with ImageNet, fine-tuned with a hashing head to extract hash values through pre-trained weights. To evaluate the performance, they used standard datasets such as CIFAR10, ImageNet, NUS-WIDE, and MS-COCO, and mAP was used as the evaluation metric. Gattupalli *et al.* [14] proposed a supervised learning-based deephashing model that generates both tag information and hash codes by passing image features extracted from AlexNet [15] through two FCN layers.

Unsupervised deephashing is based on the distribution or structure of the data to learn effective binary representation. Hoe *et al.* [16] proposed a technique to enhance deephashing performance by exploiting the cosine similarity between the continuous neural network output and the quantized binary hash codes. Liu *et al.* [17] proposed a deephashing model using a convolutional neural network to generate binary codes from images and optimized the codes using a loss function that approximates the labeled data. En *et al.* [18] proposed an unsupervised binary code learning algorithm based on a Stacked Convolutional AutoEncoder that maps input images into a low-dimensional space and generates sparse binary codes through binary relaxation.

Deephashing generates binary codes from the collected dataset as opposed to traditional hashing algorithms. Autoencoder-based deephashing utilizes many advantages over deephashing, such as non-linear mapping, dimensionality reduction, data reconstruction, end-to-end learning, and more. The optimization algorithm is operated through a binary loss function, fine-tuning it with transfer learning, and reconstruction loss in a self-supervised manner. The hash code is generated by hashing the output of the hash output layer or the intermediate layer. The models were evaluated with CIFAR-10, NUS-WIDE, ImageNet, and MS COCO datasets.

III. Proposed Approach

Each image of a dataset $\mathcal{D} = \{\mathbf{x}^{(i)} | i = 1, 2, \ldots, N\}$ was transformed into a grayscale image of size 128×128 and scaled to a value between $[0, 1]$ with the minmax scalar. The proposed deephashing model consists of an encoder and decoder: $\mathbf{M} = (\mathbf{M}_e, \mathbf{M}_d)$. The encoder learns to map the input $\mathbf{x} \in \mathcal{D}$ to a low-dimensional latent vector $\mathbf{z}$ ($|\mathbf{z}| \ll |\mathbf{x}|$) while the decoder learns to generate the reconstructed input $\hat{\mathbf{x}}$ from the latent vector $\mathbf{z}$: $\mathbf{z} = \mathbf{M}_e(\mathbf{x})$ and $\hat{\mathbf{x}} = \mathbf{M}_d(\mathbf{z})$. For all $\mathbf{x} \in \mathcal{D}$, a deephashing model proceeds with learning to minimize the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$ as much as possible. The latent vector $\mathbf{z}$ is binarized to generate the hash code $\mathbf{h}$ of $\mathbf{x}$ with threshold $\boldsymbol{\theta}$. The threshold vector $\boldsymbol{\theta}$ utilizes the average of latent vectors from the training dataset. The hashing function $\mathbf{H}(\mathbf{x}^{(i)})$ provides the binary code $\mathbf{h}^{(i)} = [\mathrm{h}_j^{(i)}]_d$ for image $\mathbf{x}^{(i)}$ of size d.

$$\boldsymbol{\theta} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}^{(i)} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{M}_e(\mathbf{x}^{(i)})$$

$$\mathbf{h}^{(i)} = \mathbf{H}(\mathbf{x}^{(i)}) = [\mathrm{h}_1^{(i)}, \mathrm{h}_2^{(i)}, \ldots, \mathrm{h}_d^{(i)}]$$

$$\mathrm{h}_j^{(i)} = \begin{cases} 1 & \text{if } \mathrm{z}_i^{(i)} > \theta_i; \\ 0 & \text{otherwise.} \end{cases}$$

Variational Autoencoder (VAE): As a generative model, VAE can learn a compressed representation of the input data. In the context of generating hash vectors, the VAE can be used to learn a compressed representation of an input data point, such as an image or a text document. To generate a hash vector using a VAE, the encoder map an input image to a point in the latent space, and then use a hash function to map that point to a hash vector. The hash function maps the continuous-valued point in the latent space to a binary vector, which serves as a hash code for the input.

By using a VAE to generate hash vectors, we can learn a compressed representation of the input data that captures

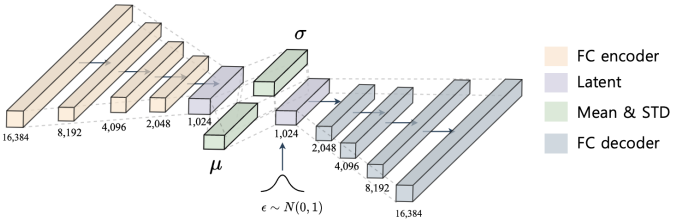

Fig. 2: VAE-based deephashing model.

important features, while still producing binary hash codes that are efficient to store and compare. Figure 2 illustrates a VAE model used for deephashing. The activation value $\mathbf{z}$ of the mean layer for the input image $\mathbf{x}$ is used to generate its hash code $\mathbf{h}$. Both the encoder and decoder consist of four fully connected layers, the input image is flattened and the layer size decreases by half for the potential layer and increases by two for the output.

Convolutional Variational Autoencoder (CVA): The performance of convolutional neural network outperform fully connected networks (FCN), particularly when feature extraction layers are utilized [11], [15]. CNN extracts features from images by applying multiple convolutional filters. These filters can extract localized features from the input image with varying filter sizes. These localized features provide detailed information on specific regions of the image, and the network learns to synthesize these features into global features. On the other hand, FCNs lose spatial information in the process of flattening an image to one dimension. It is also configured to perform pixel-by-pixel prediction, so it is not effective in recognizing features of complex images [18].

CVAS is a deephashing model that replaces the encoder and decoder of VAE with convolution layers. CVASE's encoder reduces a two-dimensional image to the vector in the latent space. The convolution layer characterizes the information of local area of pixels using several filters. It uses four layers to learn complex patterns and expressions. Reconstruction of the input from the latent vector is performed in the reverse order of the encoder layer through upsampling. Figure 3 shows the architecture of CVAE.

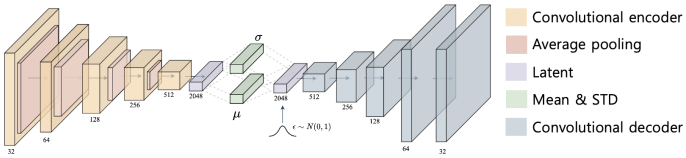

Fig. 3: CVAE-based deephashing model.

Convolutional and Simple Attention Variational Autoencoder (CSVAE): SimAM [19] is a deep learning architecture for attention-based CNN. The SimAM model combines attention mechanisms with simulated annealing, which is a heuristic optimization algorithm for finding the global minimum of a cost function. Attention mechanisms can enable the model to selectively focus on different regions of an image.

CSVAE replaces the encoder of CVAE only for deephashing. SimAM layers of the same size are inserted into the second and third layers of the CVAE encoder. The structure of the decoder does't change. With the introduction of SimAM, a hash code is generated that uses both the overall feature and the partial feature formed by the convolution layer. The hash vector thus formed was mapped to a more distinguishable latent vector in the hash space.

Variational Transformer Encoder (VTE): Vision Transformer (ViT) is a popular deep learning architecture for image classification tasks that uses self-attention mechanisms to capture long-range dependencies between image patches [10]. While ViT was not specifically designed for generating hash codes, it is possible to use the features learned by the model to generate a hash code for an input image. Its binary vector is acquired by thresholding. ViT uses a self-attention mechanism to capture global information about the input image. This feature is useful for handling larger image sizes where the global-context is important for accurate image processing. The output of ViT can be considered as the latent vector of the input image.

Variational transformer encoder (VTE) is an extension of ViT-based variational autoencoder (Figure 4). VTE enforces to learn both global feature and local feature from an input image. In addition, the latent vector converted considering the global context can generate a reconstructed image very similar to the input image while passing through the decoder.

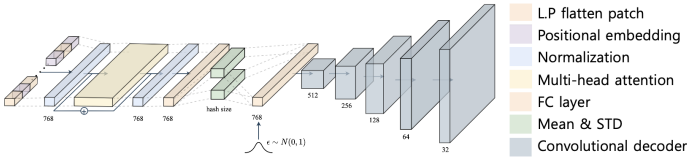

Fig. 4: VTE-based deephashing model.

Figure 5 are examples of images reconstructed from the deephashing model. The comparison of the input and output images is evaluated with the mean squared error: 8.680 ± 0.821 for VAE, 7.355 ± 1.226 for CVAE, 6.750 ± 1.623 for CSVAE and 6.664 ± 1.698 for VTE. In MSE analysis, the reconstructed output image is evaluated as similar to the input image in the order of VTE, CSVAE, CVAE, and VAE. Therefore, a deephashing model can be improved by the effect of the convolutional layer and the attention layer.

IV. Experiments

Data preparation: We populated an image database including CIFAR-10, ImageNet, and NUS-WIDE to analyze the proposed deephashing models. The CIFAR-10 dataset comprises 60,000 32×32 color images classified into 10 different categories, with each category containing 6,000 images. ImageNet is a massive image recognition dataset containing over 1.4 million images across 1,000 class categories. Images have a minimum size of 256×256 pixels, and the classes are various object categories, such as animals, plants, and household

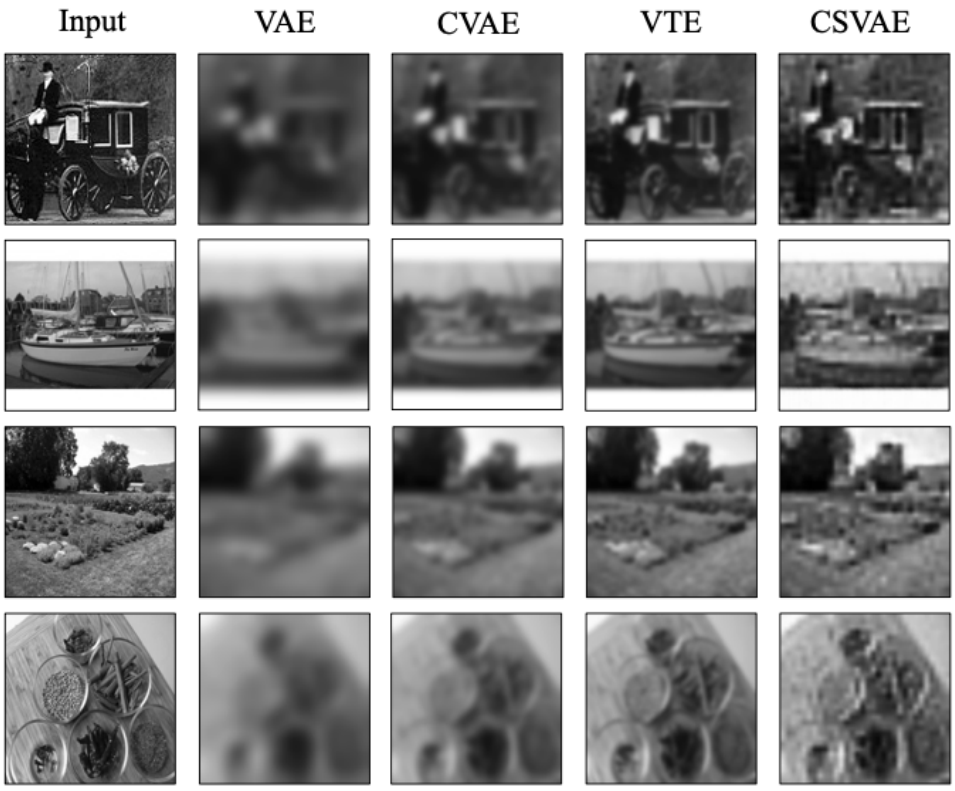

Fig. 5: Examples of reconstructed images for the deephashing models.

TABLE I: Evaluation of the proposed deephashing models

Method		CIFAR10@10000				ImageNet@10000				NusWide@10000			
		16b	32b	64b	128b	16b	32b	64b	128b	16b	32b	64b	128b
VAE	f1	0.624	0.647	0.662	0.647	0.543	0.532	0.662	0.647	0.537	0.555	0.564	0.583
	precision	0.607	0.647	0.663	0.647	0.512	0.534	0.663	0.647	0.538	0.555	0.565	0.584
	recall	0.608	0.648	0.664	0.648	0.527	0.515	0.664	0.648	0.536	0.554	0.563	0.587
	mAP	**0.410**	**0.455**	**0.475**	**0.456**	**0.319**	**0.394**	**0.475**	**0.456**	**0.348**	**0.356**	**0.407**	**0.425**
CVAE	f1	0.895	0.980	0.979	0.978	0.850	0.835	0.836	0.837	0.869	0.873	0.873	0.873
	precision	0.896	0.976	0.979	0.978	0.832	0.819	0.820	0.822	0.844	0.867	0.864	0.879
	recall	0.894	0.981	0.978	0.978	0.894	0.870	0.870	0.872	0.884	0.890	0.884	0.869
	mAP	**0.812**	**0.962**	**0.960**	**0.959**	**0.807**	**0.819**	**0.820**	**0.822**	**0.810**	**0.821**	**0.822**	**0.815**
VTE	f1	0.888	0.970	0.979	0.977	0.854	0.832	0.824	0.833	0.869	0.873	0.880	0.882
	precision	0.890	0.954	0.979	0.976	0.846	0.814	0.814	0.819	0.844	0.867	0.881	0.881
	recall	0.874	0.979	0.980	0.979	0.897	0.872	0.830	0.861	0.884	0.890	0.878	0.883
	mAP	**0.826**	**0.953**	**0.959**	**0.959**	**0.810**	**0.816**	**0.820**	**0.818**	**0.809**	**0.813**	**0.823**	**0.823**
CSVAE	f1	0.925	0.979	0.981	0.982	0.851	0.840	0.842	0.843	0.870	0.872	0.871	0.873
	precision	0.924	0.974	0.980	0.981	0.833	0.821	0.820	0.821	0.845	0.866	0.865	0.880
	recall	0.874	0.926	0.983	0.982	0.870	0.859	0.868	0.868	0.885	0.889	0.886	0.862
	mAP	**0.842**	**0.966**	**0.969**	**0.970**	**0.784**	**0.820**	**0.821**	**0.822**	**0.809**	**0.821**	**0.824**	**0.824**

items. The NUS-WIDE dataset is a well-known benchmark dataset for image retrieval and tagging, consisting of 269,648 images of various sizes. Each image has class information that selects one or more tags from a vocabulary of 81. The tags encompass a broad range of concepts, including objects, scenes, and human activities.

A total of 20,000 training and 10,000 evaluation images are generated by uniform random sampling for each dataset. The evaluation dataset is divided into 5,000 images from the search database and 5,000 images from the query. The search database stores the original images, binary hash codes extracted from the hashing model, and labels. Search calculates the Hamming distance between the hash code generated from the query image and the hash code from the search database, and returns the image with the shortest distance.

The performance evaluations are precision, recall, f1-score, and mAP of query image labels and search image labels. Precision is the proportion of relevant images out of all images retrieved and evaluates whether the retrieved image has the same label as the query. Recall is the percentage of relevant images retrieved out of all images in the data set. The f1-score is the harmonic average of precision and recall and provides a single score that balances both measures. mAP is a metric that considers the precision and recall of a retrieval system across different thresholds. This provides a single value that represents an overall measure of the model's

TABLE II: Hash collision ratio.

Method	Collision Ratio(%)											
	CIFAR10@10000				ImageNet@10000				NusWide@10000			
	16b	32b	64b	128b	16b	32b	64b	128b	16b	32b	64b	128b
VAE	0.34	0.32	0.29	0.27	0.39	0.37	0.33	0.33	0.36	0.29	0.26	0.30
CVAE	0.28	0.23	0.21	0.21	0.24	0.16	0.14	0.13	0.26	0.18	0.18	0.15
VTE	0.24	0.23	0.21	0.21	0.22	0.15	0.13	0.11	0.23	0.20	0.18	0.15
CSVAE	0.23	0.23	0.21	0.21	0.22	0.23	0.15	0.13	0.13	0.22	0.19	0.16

TABLE III: Comparison of the proposed model and studied models.

Method	CIFAR10@10000			ImageNet@10000			Nus-Wide@10000		
	16b	32b	64b	16b	32b	64b	16b	32b	64b
DHN [20]	0.693	0.645	0.588	0.311	0.472	0.573	-	0.748	-
HashNet [21]	0.748	0.778	0.626	0.506	0.631	0.684	0.662	0.699	0.716
DPN [22]	0.774	0.803	0.812	0.608	0.691	0.727	0.810	0.822	0.839
TransH [23]	0.908	0.911	0.917	0.820	0.832	0.833	0.726	0.739	0.749
Our work	**0.842**	**0.966**	**0.969**	**0.784**	**0.820**	**0.821**	**0.809**	**0.821**	**0.824**

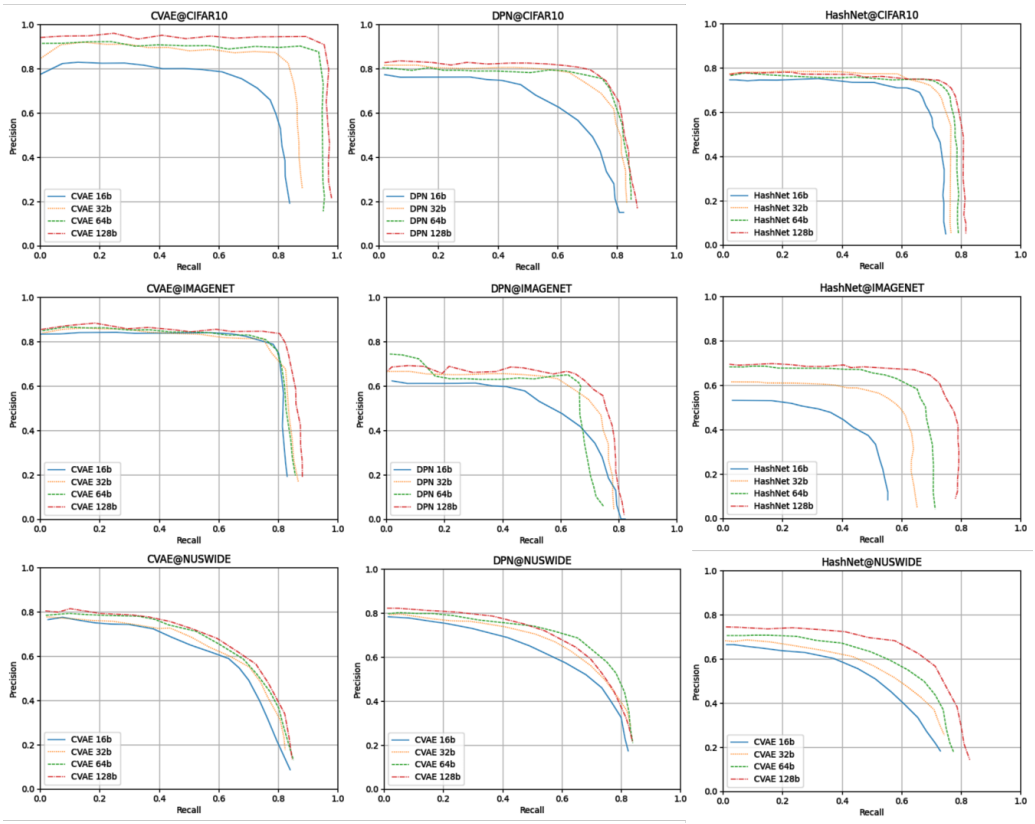

Fig. 6: mAP comparison of CVAE, DPN and HashNet.

performance in detecting objects across different categories. Alternatively, when mapping arbitrary-size inputs to fixed-size outputs, conflicts may arise due to the inherent characteristics of the model and its learning method. The hash collision rate is:

$$\text{Collision rate} = \frac{\text{No. of collided hashes}}{\text{No. of test images}} \times 100.0$$

Experimental results: Table I is the search performance result for CIFAR-10, ImageNet and NUS-WIDE. The evaluation metrics are calculated by label comparison between the searched image and the query image through Hamming distance in 16-bit, 32-bit, 64-bit, and 128-bit hash spaces. Except for the VAE model, the evaluation indicators of other models are excellent. It is analyzed that the introduction of the convolution layer and the attention layer helped to learn the deephashing model. It shows the highest score at 64 hash bits, so a latent vector size of the evaluation dataset is 4 bytes.

The hash collision rate of the deephashing model was compared (Table II). On the 10,000 ImageNet test dataset, the VTE model had 0.11% hash collisions at a hash size of 128 bits, resulting in hash collisions for 1 test image. The hash collision rate of the VAE model was $0.26\tilde{0}.37$ across the three datasets, showing a higher collision rate. At the same hash

size, the hash collision rate of the other models excluding the VTE model was similar.

Comparison with other studies: We compare and evaluate the search performance of hash codes extracted using previously studied deephashing models. Table III is the mAP comparison between the VTE model, which showed the highest performance among the proposed models, and the supervised deephashing models that have been studied previously. Figure 6 compares the mAP graph of CVAE, DPN and HashNet. The performance of the proposed deephashing model is relatively high, and it is analyzed that a simple model modification can yield distinguishable hash vectors. TransH [23] showed the highest performance among the comparison models, and is compared with the performance of the proposed model. The result shows that VTE is very competitive to TransH even though VTE is a VAE model with ViT and CNN modules.

V. Conclusion

In this study, we proposed deephashing models using attention, transformer, and CNN modules for an image retrieval system. These deephashing models can produce hash codes that can maintain semantic similarity of input images to overcome the limitation of conventional hash methods that often produce different hash codes for a single image domain, even with minor changes. The proposed model performs unsupervised variation inference learning and self-supervised learning to output the same input image. Through comparative experiments, we achieved competitive performance with supervised learning-based models on the CIFAR-10, ImageNet, and NUS-WIDE datasets. The proposed deephashing methods can generate compact and efficient hash codes with low-dimensional hashing vectors that can lead to better search performance. Consequently, deephashing will enhance an image retrieval system by generating binary hash that represent visual features of images, enabling fast and efficient searches within large image databases.

References

[1] Ling Du, Anthony T.S. Ho, and Runmin Cong, "Perceptual hashing for image authentication: A survey," *Signal Processing: Image Communication*, vol. 81, pp. 115713, 2020.

[2] Jiafa Mao, Danhong Zhong, Yahong Hu, Weiguo Sheng, Gang Xiao, and Zhiguo Qu, "An image authentication technology based on depth residual network," *Systems Science & Control Engineering*, vol. 6, pp. 57–70, 01 2018.

[3] Doyoung Kim, Suwoong Heo, Jiwoo Kang, Hogab Kang, and Sanghoon Lee, "A photo identification framework to prevent copyright infringement with manipulations," *Applied Sciences*, vol. 11, pp. 9194, 10 2021.

[4] Shuqin Zhu, Congxu Zhu, and Wenhong Wang, "A new image encryption algorithm based on chaos and secure hash sha-256," *Entropy*, vol. 20, no. 9, pp. 716, 2018.

[5] Li-Wei Kang, Chun-Shien Lu, and Chao-Yung Hsu, "Compressive sensing-based image hashing," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 1285–1288.

[6] Xiao Luo, Haixin Wang, Daqing Wu, Chong Chen, Minghua Deng, Jianqiang Huang, and Xian-Sheng Hua, "A survey on deep hashing methods," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, pp. 1–50, 2023.

[7] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[8] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*. Springer, 2011, pp. 44–51.

[9] Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] 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," *ArXiv*, vol. abs/2010.11929, 2020.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Shiv Ram Dubey, Satish Kumar Singh, and Wei-Ta Chu, "Vision transformer hashing for image retrieval," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] Vijetha Gattupalli, Yaoxin Zhuo, and Baoxin Li, "Weakly supervised deep image hashing through tag embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10375–10384.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[16] 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*, vol. 34, pp. 24286–24298, 2021.

[17] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen, "Learning multifunctional binary codes for both category and attribute oriented retrieval tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3901–3910.

[18] Sovann En, Bruno Crémilleux, and Frédéric Jurie, "Unsupervised deep hashing with stacked convolutional autoencoders," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3420–3424.

[19] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie, "Simam: A simple, parameter-free attention module for convolutional neural networks," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 11863–11874, PMLR.

[20] 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*, 2016, vol. 30.

[21] 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*, 2017, pp. 5608–5617.

[22] 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 *IJCAI*, 2020, pp. 825–831.

[23] Yongbiao Chen, Sheng Zhang, Fangxin Liu, Zhigang Chang, Mang Ye, and Zhengwei Qi, "Transhash: Transformer-based hamming hashing for efficient image retrieval," in *Proceedings of the 2022 International Conference on Multimedia Retrieval*, 2022, pp. 127–136.