Image Retrieval by Hierarchy-aware Deep Hashing Based on Multi-task Learning

Bowen Wang[1], Liangzhi Li[1], Yuta Nakashima[1], Takehiro Yamamoto[2], Hiroaki Ohshima[2], Yoshiyuki Shoji[3], Kenro Aihara[4], Noriko Kando[4]

[1]Institute for Datability Science (IDS), Osaka University, Japan

[2]University of Hyogo, National Institute of Informatics

[3]Aoyama Gakuin University

[4]National Institute of Informatics, The Graduate University for Advanced Studies

bowen.wang@is.ids.osaka-u.ac.jp,{li,n-yuta}@ids.osaka-u.ac.jp,t.yamamoto@sis.u-hyogo.ac.jp

ohshima@ai.u-hyogo.ac.jp,shoji@it.aoyama.ac.jp,{kenro.aihara,kando}@nii.ac.jp

ABSTRACT

Deep hashing has been widely used to approximate nearest-neighbor search for image retrieval tasks. Most of them are trained with image-label pairs without any inter-label relationship, which may not make full use of the real-world data. This paper presents deep hashing, named HA^{2}SH, that leverages multiple types of labels with hierarchical structures that an ethnological museum assigns to their artifacts. We experimentally prove that HA^{2}SH can learn to generate hashes that give a better retrieval performance. Our code is available [1].

CCS CONCEPTS

• **Computing methodologies → Visual content-based indexing and retrieval**.

KEYWORDS

image retrieval, hierarchical label structure, real-world dataset

ACM Reference Format:

Bowen Wang[1], Liangzhi Li[1], Yuta Nakashima[1], Takehiro Yamamoto[2],, Hiroaki Ohshima[2], Yoshiyuki Shoji[3], Kenro Aihara[4], Noriko Kando[4]. 2021. Image Retrieval by Hierarchy-aware Deep Hashing Based on Multi-task Learning. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3460426.3463586

1 INTRODUCTION

Museums have large image database to record their collections of artifacts. For example, the British Museum made their database with 1.9M images available online[2]. Each artifact (or equivalently image) often comes with rich metadata, including codes encoding

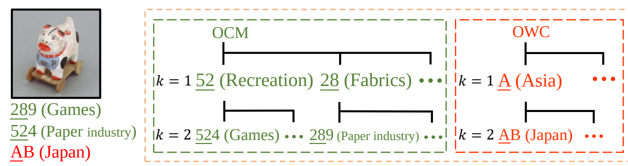

Figure 1: An example of artifact (or image) in our dataset. It comes with OCM and OWC codes to roughly represent the functionalities of the artifact and where it originates. k is the hierarchical level of the labels.

taxonomic classification as shown in Figure 1. Such an image database can facilitate the experience in the museum by, e.g., providing a handy and easy-to-use app to retrieve relevant artifacts (or images) by taking an image of the artifact in an exhibition, allowing exploration of relevant artifacts on the artifact for visitors.

In order to implement such an app, a powerful and efficient approach for image retrieval is necessary. Because of its high computational and storage efficiency, hashing [6, 15, 18, 20] can be the possible choice. Deep hashing [2, 4, 8, 12, 14, 23–29] adopts deep convolutional neural networks (CNNs) [9, 10] as base network to learn a nonlinear hash function. It allows large-scale retrieval of images [3, 24] and videos [1, 7, 13, 19].

Previous works often use existing datasets, such as ImageNet and COCO [5, 11, 21, 22], to train the image retrieval models; however this may not be very coherent with actual image retrieval scenario for, e.g., the museum uses. In real-world data, an artifact (or an image) can have multiple taxonomic classifications to describe different aspects of the artifact. For example, the artifact in Figure 1 is originated from Japan, used as a toy, and made of paper, which are encoded into multiple classification codes. Furthermore, such codes can also encode taxonomic hierarchy, e.g., Japan is in Asia. This inherently leads to the multi-task, multi-label, and hierarchical nature of this image retrieval task.

This paper presents a new approach for hierarchy-aware hashing, called HA^{2}SH, which can handle real-world data derived from an actual database provided by an ethnological museum. There are two label spaces associated with each artifact, where one of them can have multiple labels and both of them have hierarchical structures. In order to generate hashes dedicated for the two label spaces, HA^{2}SH uses a shared CNN followed by two branches with a respective hash layer to generate hashes. Multi-task learning

[1]Code is available at https://github.com/wbw520/minpaku.
[2]https://www.britishmuseum.org/collection

ICMR '21, August 21–24, 2021, Taipei, Taiwan.

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463586

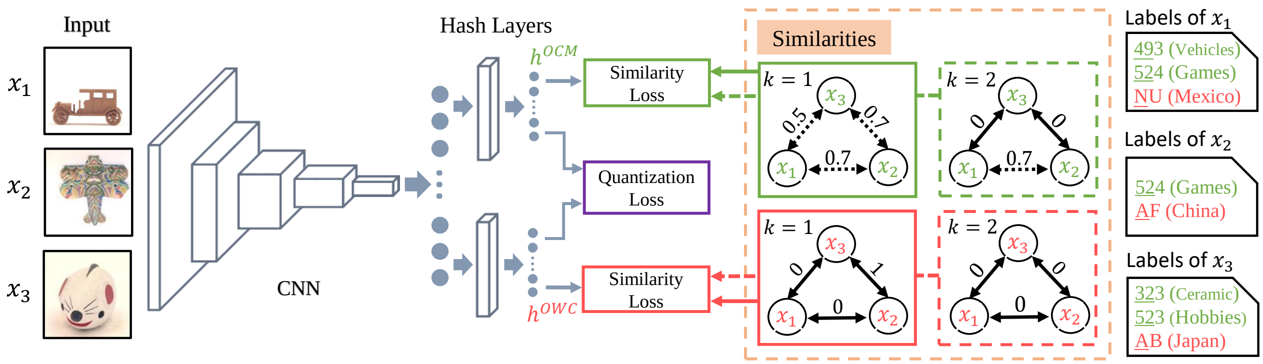

Figure 2: Overview of HA^{2}SH with OCM and OWC branches to generate respective hashes. The cosine similarity is used to define the hierarchical image-image similarities, which provide hard (solid lines) and soft (dotted lines) similarity losses.

with respective losses is adopted for better image representation, whereas the losses take the multiple labels and their hierarchy into account. Our three main contributions are as follows: 1) We propose HA^{2}SH, which is trained in a multi-task and multi-label paradigm for hierarchy-aware hashing. 2) We design a flexible retrieval system that allows controlling the importance of different hashes to meet actual users' needs. 3) We evaluate HA^{2}SH with a real-world dataset derived from an actual database provided by an ethnological museum.

2 OUR DATASET

Under our collaborative project with an ethnological museum, we were granted access to the database of its collection of artifacts, which contains images and metadata of each artifact. We extracted these images and associated metadata to build our dataset, containing 450,443 images (127,337 artifacts) in total. The metadata includes various information on the artifact, and we used as label the outline of cultural materials (OCM) and the outline of world cultures (OWC) defined by Human Relations Area Files[3], where OCM and OWC roughly describe the function and culture of the artifact.

One important aspect of OCM and OWC is their hierarchical structures. The semantics is encoded in a few-digit code, representing a certain category and its subcategory; for example, OCM's three-digit label 524 stands for "game", where the first two digits 52 means recreation. OWC label `AB06` stands for the culture of "Ainu," where the first digit and the first two digits mean "Asia" and "area of Japan." We used only the first two digits of the OWC label to identify the region that the artifact is originated. The first and second levels of OCM have 31 and 80 classes, where those of OWC have 8 and 50 classes. Each image has at least one OCM and OWC labels; many images have multiple OCM labels as an artifact can serve multiple functions. An example data is shown in Figure 1.

3 HIERARCHY-AWARE HASHING

Given the dataset above, we design a deep hash that takes into account the multi-task, multi-label, and hierarchical nature of our dataset for image retrieval.

[3]https://ehrafworldcultures.yale.edu/ehrafe/

3.1 Problem Formulation

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ be the set of images in our dataset, from which images similar to a query image are retrieved. HA^{2}SH finds a mapping from an image to a Q-bit binary code $\mathcal{B} = \{b_i\}_{i=1}^N$, where $b_i \in \{-1, 1\}^Q$. Code b_i is trained to be locality-sensitive, and its neighbors may be semantically similar to each other. Following the previous work [4, 26, 27], instead of generating binary codes, we adopt continues relaxation $h_i \in [-1, 1]^Q$, which can be easily mapped to a binary code by taking the sign of each element of h_i; therefore, HA^{2}SH learns mapping f from x_i to h_i.

The semantic similarity between a pair of images is defined based on their labels. The labels for image $x_i \in \mathcal{X}$ can be represented by a multi-hot vector $z_i^{(u,k)}$, where $u \in \{\text{OCM}, \text{OWC}\}$ (for OWC, the vector is often reduced to a one-hot vector) and k is the level in the label hierarchy (k is either 1 or 2 for both OCM and OWC). We adopt the same strategy as IDHN [27] to define similarity $s_{ij}^{(u,k)}$ between images x_i and x_j with labels $z_i^{(u,k)}$ and $z_j^{(u,k)}$ using the cosine similarity, i.e.,

$$s_{ij}^{(u,k)} = z_i^{(u,k)} \cdot z_j^{(u,k)} / (\|z_i^{(u,k)}\| \, \|z_j^{(u,k)}\|), \tag{1}$$

where "$\cdot$" is the operator for inner product. This definition quantifies a fine-grained semantic similarity, taking the multi-label nature of our dataset by allowing similarity in-between 0 and 1. The similarities in the different levels are fused through the loss function to generate hierarchy-aware multi-level deep hashes. For notation simplicity, we omit u and k unless it is ambiguous.

Figure 2 shows the pipeline of our model. A CNN is used as the backbone for feature extraction. HA^{2}SH branches after the global average pooling to generate different hashes for OCM and OWC. Each branch has a hash layer, consisting of an fc layer and the $tanh(\cdot)$ nonlinearity, to generate hash $h_i^{(u)}$.

3.2 Learning from Similarities

For a pair of hashes h_i and h_j, we use the inner product $h_i \cdot h_j$ to measure the distance between them, which is proved to be a good alternative of the Hamming distance used for binary hashes to quantify the pairwise similarity [3, 27, 29]. We train our mapping f

for label category u so that generated hashes h_i and h_j well encode our label-based similarity s_{ij} for image pair (x_i, x_j).

Hard similarity loss. Let $\mathcal{S}_1$ and $\mathcal{S}_0$ be the sets of image indices pairs (i, j) whose (multiple) labels are exactly the same (i.e., $s_{ij} = 1$) or completely different (i.e., $s_{ij} = 0$), respectively. Pairs in these sets give a strong signal that corresponding hashes h_i and h_j are close to or far from each other. To encode this, similar to HashNet [4], we define the probability of the similarity given a pair of hashes as

$$p(s_{ij} \mid h_i, h_j) = \begin{cases} \sigma(h_i \cdot h_j) & \text{for } (i,j) \in \mathcal{S}_1 \\ 1 - \sigma(h_i \cdot h_j) & \text{for } (i,j) \in \mathcal{S}_0 \end{cases} \tag{2}$$

where $\sigma(\cdot) \in [0, 1]$ is the sigmoid function. Generally, the number of image pairs with the same set of labels is far less than those of completely different set of labels. We therefore introduce a weight w_{ij} that gives γ for $(i, j) \in \mathcal{S}_1$ and $1 - \gamma$ for $(i, j) \in \mathcal{S}_0$ to mitigate the imbalance and define the loss function as

$$\ell_\mathrm{H} = -\sum_{(i,j) \in \mathcal{S}_1 \cup \mathcal{S}_0} w_{ij} \log p(s_{ij} \mid h_i, h_j). \tag{3}$$

Soft similarity loss. For pairs (i, j) that have partially matched sets of labels, we use the loss defined in IDHN [27]. Let $\mathcal{S}'$ denote the set of indices pair (i, j) such that $s_{ij} < 1$. The soft similarity loss is given by:

$$\ell_\mathrm{S} = -\sum_{(i,j) \in \mathcal{S}'} \left(\frac{h_i \cdot h_j + Q}{2} - s_{ij} Q \right)^2. \tag{4}$$

This loss enforces the correlation between $h_i \cdot h_j$ and s_{ij} to take into account the multiple labels assigned to a single image.

Quantization loss. We use $tanh(\cdot)$ to squash the output of the hash layer to be in $[-1, 1]$, but this does not guarantee that the resulting hash has values closer to either 1 or -1. We thus use the quantization loss, given by

$$\ell_\mathrm{Q} = \sum_i \| |h_i| - \mathbf{1}_Q \|^2, \tag{5}$$

where $|h_i|$ gives the absolute value element-wise and $\mathbf{1}_Q$ is a vector with all its Q elements being 1.

Overall loss for hierarchical training (HT). Due to the hierarchical structure of our labels, the hard and soft similarity loss can be defined for respective levels of the hierarchies. Therefore, the loss for branch u is given by combining the losses as:

$$L^{(u)} = \sum_k \ell_\mathrm{H}^{(u,k)} + \delta \sum_k \ell_\mathrm{S}^{(u,k)} + \lambda \ell_\mathrm{Q} \tag{6}$$

where δ and λ are weights to control the soft similarity and quantization losses, respectively. The model is trained in the multi-task learning framework, in which the following loss is used to train mappings f^OCM and f^OWC:

$$L = \sum_u L^{(u)} = L^\mathrm{OCM} + L^\mathrm{OWC}. \tag{7}$$

Table 1: Similarity learning experiments with 32-bits and 64-bits of hash codes. Evaluated with mAP@1000.

Branch			OCM		OWC	
OCM	OWC	Soft Loss	32-bit	64 bit	32 bit	64-bit
	✓		0.233	0.256	0.585	0.625
	✓	✓	0.227	0.254	0.588	0.623
✓			0.629	0.701	0.300	0.342
✓		✓	0.644	0.713	0.319	0.334
✓	✓		0.645	0.711	0.600	0.632
✓	✓	✓	0.652	**0.726**	0.597	**0.635**

Table 2: Performance of Hierarchy-aware hashing in both branches and hierarchical level k.

	OCM		OWC	
Label Setting	$k = 1$	$k = 2$	$k = 1$	$k = 2$
Only First-level	0.791	-	0.834	-
Only Second-level	0.754	0.726	0.769	0.634
All Levels	0.788	0.712	0.829	0.601

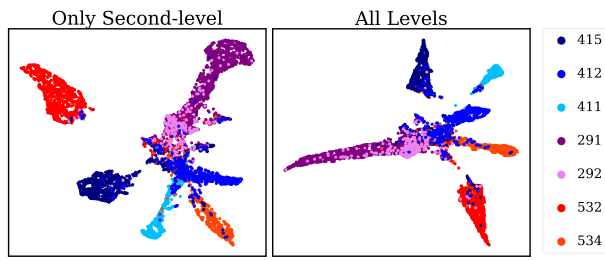

Figure 3: The visualization of UMAP for classes share similar semantic meanings

3.3 Retrieval

Given query image q, we retrieve similar images in $\mathcal{X}$, for which we preliminary compute the set $\mathcal{H}^{(u)} = \{h_i^{(u)}\}_{i=1}^N$ of hashes. The pairwise distance between q and x_i can be given by

$$D_i^{(u)}(q) = f^{(u)}(q) \cdot h_i^{(u)}. \tag{8}$$

We combine the distances for OCM and OWC with weight $\alpha \in [0, 1]$ to provide flexible retrieval that can take both aspects of images into account:

$$D = \alpha D^\mathrm{OCM} + (1 - \alpha) D^\mathrm{OWC}. \tag{9}$$

Images in $\mathcal{X}$ are ranked according to D.

4 EXPERIMENTS

We implemented HA^{2}SH with PyTorch, using a CNN backbone ResNet-50 [9] pre-trained on the ImageNet classification task [10]. The model was trained for 30 epochs with AdamW [16], which started with a learning rate 10^{-4}, decreased by a factor of 10 at the 20-th epoch. The learning rate of hash layers is set to be 5 times

Figure 4: Demonstration of top 10 retrieval answers with different interest factors α.

greater than the backbone network. Based on the data statistic, we set γ and δ to 0.9 and 1. λ is set to 0.1. For learning the hierarchical structures in the labels, we only used the first-level ($k = 1$) for the first 10 epochs and then added the second-level's loss.

For evaluation, 2% of the images are randomly picked out as query images for evaluation and the rest are used as image database $\mathcal{X}$. We randomly sampled 50% of the image database for training. We use the mean Average Precision (mAP) for evaluating our model.

4.1 Effects of Multi-task and Multi-label Losses

To evaluate the collective effect of two branches (OCM and OWC) and multiple labels for 32-bit and 64-bit hashes, we used only second-level ($k = 2$) labels with removing some losses (the losses for OCM and OWC branches; and the soft similarity loss). As shown in Table 1, the performance by multi-task losses with the soft similarity loss was better than those of individual tasks'. Interestingly, the model trained only for OCM labels can still give relevant images for OWC, and vice versa. This implies the correlation between OCM and OWC labels. The soft similarity loss worked well for the OCM labels, while in the OWC space, there are not many multi-label cases and this loss serves slightly.

4.2 Effects of Hierarchy Awareness

The hard/soft similarity losses encourage images with the same label to form a cluster in the hash space. Our hierarchy-aware hashing is for learning a better hash space, which forces the model to put images with semantically similar (i.e., the first level $k = 1$) classes closer to each other. This gives an extra value for image retrieval. We used UMAP [17] to visualize the 64-bit hashes trained with OCM in a 2-D space. We sampled some images (not multi-label) with the chosen classes that share the first-level labels. For example, labels 532 (*Representative art*) and 534 (*Musical instruments*) belong to the first-level class 53 (*Art*).

Figure 3 shows the visualizations. When the model is only trained with second-level ($k = 2$) labels (left), the first-level labels appear to be randomly placed. With this hash space, retrieved images can return semantically irrelevant images. With hierarchical training, the first-level labels bring images with similar semantic meanings closer, and then training with the second-level labels refines the clusters. The distribution of images in the hash space roughly takes this hierarchical structure into account as shown in Figure 3 (right).

Table 2 shows the performances of two models: one only trained with second-level labels ($k = 2$) and the other with all levels labels. Both models have two branches and use the soft similarity loss. They are evaluated with OCM and OWC labels at both first ($k = 1$) and second ($k = 2$) levels. The results show that for both OCM and OWC, the performances improved for the first level with all levels training at the cost of the second-level performance.

Table 3: The performance for different α values, evaluated with respect to OCM, OWC, and their union, in mAP@1000.

α	0	0.25	0.5	0.75	1
OCM	0.313	0.681	0.690	0.698	0.712
OWC	0.601	0.598	0.591	0.584	0.392
Union	0.216	0.559	0.544	0.551	0.342

4.3 Evaluation of Retrieval Performance

α controls the users' preference on OCM and OWC. We evaluated our model with different values of α. Table 3 summarizes the performances for different α, evaluated with respect to OCM, OWC, and their union (an image is counted as correct if both OCM and OWC labels are the same as a query), where the model is trained with the two branches, the soft similarity loss, and all levels labels. The results demonstrate that, as expected, $\alpha = 0$ gives a higher OWC performance, while $\alpha = 1$ gives a higher OCM performance.

Figure 4 gives examples of retrieval results for different α. The query has OCM label *323* (*Ceramic technology*) and *534* (*Musical instruments*), as well as OWC label AB (*Japan*). Images marked with black boxes are with the exact same labels (of both OCM and OWC) as the query. The orange and blue boxes represent partial matches and complete mismatches, respectively. For $\alpha = 1$, all retrieved images are with OWC label AB. OCM labels are almost correct, which may imply a high correlation between OWC and OCM labels. Meanwhile, for $\alpha = 0$, which fully focuses on OWC, HA^{2}SH gave relatively diverse images. When $\alpha = 0.5$, all retrieved images are with the same labels as the query.

5 CONCLUSION

In this paper, we proposed HA^{2}SH for image retrieval, targeted at the ethnological museum database. Our results demonstrated that HA^{2}SH can leverage multiple labels and their hierarchical structures to learn a better hash. It fuses hash codes learned from different types of labels to offer a flexible retrieval system. We believe HA^{2}SH provides a good user experience in museum apps. Our future work includes a subjective evaluation to show the usability of the retrieval system in some application scenarios.

Acknowledgements This work is supported by JSPS KAKENHI Nos. JP18H03264, JP18K18161, JP18H03243, JP18H03494, and JP16H01756, as well as ROIS NII Open Collaborative Research 2020 (20S1001, 20AD03, and 20AD04). Our dataset is based on a database provided by National Museum of Ethnology, MINPAKU, together with OWC and OCM definitions by HRAF Association. Here we express our gratitude to their help.

REFERENCES

[1] Mahnaz Amiri Parian, Luca Rossetto, Heiko Schuldt, and Stéphane Dupont. 2020. Are you watching closely? Content-based retrieval of hand gestures. In *Proc. ICMR.* 266–270.

[2] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. 2018. Deep cauchy hashing for hamming space retrieval. In *Proc. CVPR.* 1229–1237.

[3] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. 2016. Deep quantization network for efficient image retrieval. In *Proc. AAAI.*

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. 2017. Hashnet: Deep learning to hash by continuation. In *Proc. ICCV.* 5608–5617.

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proc. ICMR.* 1–9.

[6] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*, Vol. 99. 518–529.

[7] Yun Gu, Chao Ma, and Jie Yang. 2016. Supervised recurrent hashing for large scale video retrieval. In *Proc. ACM Multimedia.* 272–276.

[8] Jie Gui, Tongliang Liu, Zhenan Sun, Dacheng Tao, and Tieniu Tan. 2016. Supervised discrete hashing with relaxation. *IEEE Transactions on Neural Networks and Learning Systems* 29, 3 (2016), 608–617.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR.* 770–778.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Proc. NeurIPS.* 25 (2012), 1097–1105.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Proc. ECCV.* 740–755.

[12] Yen-Liang Lin, Son Tran, and Larry S Davis. 2020. Fashion outfit complementary item retrieval. In *Proc. CVPR.* 3311–3319.

[13] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. 2016. Deep video hashing. *IEEE Transactions on Multimedia* 19, 6 (2016), 1209–1219.

[14] Qingshan Liu, Guangcan Liu, Lai Li, Xiao-Tong Yuan, Meng Wang, and Wei Liu. 2017. Reversed spectral hashing. *IEEE Transactions on Neural Networks and Learning Systems* 29, 6 (2017), 2441–2449.

[15] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. 2012. Supervised hashing with kernels. In *Proc. CVPR.* 2074–2081.

[16] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Proc. ICLR.* (2019).

[17] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint arXiv:1802.03426* (2018).

[18] Mohammad Norouzi and David J Fleet. 2011. Minimal loss hashing for compact binary codes. In *Proc. ICML.*

[19] Jie Qin, Li Liu, Mengyang Yu, Yunhong Wang, and Ling Shao. 2017. Fast action retrieval from videos via feature disaggregation. *Computer Vision and Image Understanding* 156 (2017), 104–116.

[20] Maxim Raginsky and Svetlana Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. *Proc. NeurIPS.* 22 (2009), 1509–1517.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[22] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. 2018. Learning type-aware embeddings for fashion compatibility. In *Proc. ECCV.* 390–405.

[23] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. 2017. Deep supervised hashing for multi-label and large-scale image retrieval. In *Proc. ICMR.* 150–158.

[24] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised hashing for image retrieval via image representation learning. In *Proc. AAAI.*

[25] Jiehao Xu, Chengyu Guo, Qingjie Liu, Jie Qin, Yunhong Wang, and Li Liu. 2019. DHA: Supervised deep learning to hash with an adaptive loss function. In *Proc. ICCV. Workshops.* 0–0.

[26] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central similarity quantization for efficient image and video retrieval. In *Proc. CVPR.* 3083–3092.

[27] Zheng Zhang, Qin Zou, Yuewei Lin, Long Chen, and Song Wang. 2019. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. *IEEE Transactions on Multimedia* 22, 2 (2019), 540–553.

[28] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. 2018. Graph convolutional network hashing. *IEEE Transactions on Cybernetics* 50, 4 (2018), 1460–1472.

[29] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep hashing network for efficient similarity retrieval. In *Proc. AAAI.*