buildings

MDPI

Article

Image Retrieval for Local Architectural Heritage Recommendation Based on Deep Hashing

Kai Ma [1,2,*], Bowen Wang [3], Yunqin Li [2,4] and Jiaxin Zhang [2,4]

[1] School of Architecture, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, China
[2] School of Civil Engineering and Architecture, Nanchang University, No. 999, Xuefu Avenue, Honggutan New District, Nanchang 330031, China; li@it.see.eng.osaka-u.ac.jp (Y.L.); zhang@it.see.eng.osaka-u.ac.jp (J.Z.)
[3] Science and Technology, Graduate School of Information, Osaka University, 1-1, Yamadaoka, Osaka 565-0871, Japan; bowen.wang@is.ids.osaka-u.ac.jp
[4] Division of Sustainable Energy and Environmental Engineering, Graduate School of Engineering, Osaka University, 1-1, Yamadaoka, Osaka 565-0871, Japan
* Correspondence: makai@ncu.edu.cn

Abstract: Propagating architectural heritage is of great significance to the inheritance and protection of local culture. Recommendations based on user preferences can greatly benefit the promotion of local architectural heritage so as to better protect and inherit historical culture. Thus, a powerful tool is necessary to build such a recommendation system. Recently, deep learning methods have proliferated as a means to analyze data in architectural domains. In this paper, based on a case study of Jiangxi, China, we explore a recommendation system for the architectural heritage of a local area. To organize our experiments, a dataset for traditional Chinese architecture heritage is constructed and a deep hashing retrieval method is proposed for the recommendation task. By utilizing a data fine-tuning strategy, our retrieval method can realize high-accuracy recommendation and break the model training restriction caused by insufficient data on local architectural heritage. Furthermore, we analyze the retrieval answers and map the data into a two-dimensional space to reveal the relationships between different architectural heritage categories. An *image-to-location* application is also provided for a better user experience.

Keywords: deep learning; architectural heritage; image retrieval

check for updates

Citation: Ma, K.; Wang, B.; Li, Y.; Zhang, J. Image Retrieval for Local Architectural Heritage Recommendation Based on Deep Hashing. *Buildings* **2022**, *12*, 809. https://doi.org/10.3390/buildings12060809

Academic Editor: Dirk H.R. Spennemann

Received: 10 May 2022
Accepted: 8 June 2022
Published: 12 June 2022

1. Introduction

Architectural heritage is a type of cultural heritage that is material and immovable. All physical objects created by various construction activities during the evolution and development of human society belong to architectural heritage. From small buildings and structures to large villages and cities, whether or not they are currently intact does not affect their basic material properties of being immovable. As a product of history, architectural heritage is marked by the era, reflecting the level of science and technology of the corresponding era, as well as the social, political, military, and cultural conditions of the time for a local area. Therefore, it is essential to let more people know the story behind architectural heritage, which is of great significance to the promotion and inheritance of local culture. To achieve this, we need a powerful tool to recommend existing architectural heritage based on user preferences.

Recently, digital technology has become popular, which has great significance for cultural heritage [1–4]. A comprehensive digital archive is established for heritages by means of photography, scanning, and 3D printing. The rise of digital information technology has greatly broken through the limitations of traditional conservation means. It provides richer technical options for recognition, protection, presentation, and finding a breakthrough for solving the problems of heritage conservation. Digital tools also have the advantage of increasing opportunities in terms of cultural mediation and proposing interactive tools that involve the public in discovery cultures, thus making them actors in their own knowledge acquisition processes. Such kinds of new technologies allow us more ways to access and promote the local architectural heritage. In particular, the deep learning (DL) method is currently showing its power in this area. Recently, the work by Tejedor et al. [5] introduces the future perspectives of the DL method for the diagnosis of heritage buildings. The methods reviewed in the paper cover several important topics, e.g., object detection which can be integrated with UAVs and a GPS system [6], 3D models for the reconstruction of lost architectural heritage [7], and neural networks applied to heat flux meter (HFM) methods [8,9]. Some defects of the DL method are also given in the paper, especially the limitation of dataset and computation time cost. For the promotion of architectural heritage, we introduce another kind of DL method, namely, content-based image retrieval (CBIR), for a case study. A fine-tuning strategy is adopted to solve the data quantity problem, which is a common issue for real-world application of DL. We detail our target, method, and contributions in the continuous parts of the paper.

With the rapid development of convolutional neural networks (CNNs) [10,11], DL [12–14] methods have achieved many encouraging results in heritage image analysis [15,16]. Among them, CBIR [17,18], which searches for images from a large image collection, can serve as a recommendation tool for local architectural heritage. Suppose we have an image collection that contains a variety of architectural heritages from a local area. For one query image (e.g., an image of a temple) uploaded by the user, the retrieval method can return the top-ranked images from the image collection based on visual similarities. The retrieved images are the recommendations for users. However, applying image retrieval to local architectural heritage faces an unavoidable problem—data insufficiency. The amount of architectural heritage in a local area may not have enough quantity. However, having adequate data is always an essential factor for the training of a DL model [19]. It is easy to gather images of some common classes (e.g., residential buildings) in a local area, while, for some landmark buildings (e.g., bridges), there may be only a few entities available. The lack of data and data imbalance among classes will lead to a decrease in retrieval accuracy or cause training failure. Therefore, solving the problem of data insufficiency is the first priority.

As we discussed above, it is almost impossible to collect enough data in a local area for the training of a DL retrieval model. However, can we collect data from a broader area for training purposes? For architectural heritages from the same class, there are of course diverse feature distributions among different areas. For example, the palaces from China tend to be majestic and magnificent, but the palaces in Japan are graceful and delicate. However, based on the definition of architectural functionality, the architectures from the same class always share many common characteristics. Thus, for better training of the retrieval model, we designed a data fine-tuning strategy similar to the thought of transfer learning [20]. We can collect architectural heritage images from a broader area, or even not limited to heritage, to form the *source data* and then transfer the shared knowledge to the local architecture heritage *target data*. Taking Jiangxi, China, as a local area example, our experiment proves that utilizing this strategy enables better retrieval performance and enhances the feature extraction ability of the model for target data.

In this paper, we explore a case study for the recommendation of local architectural heritage. The main issue we have to address is the data quantity restriction for local architectural heritage. For this purpose, a new dataset named Chinese Architectural Heritage 10 (CAH10) is constructed. CAH10 contains 3080 images from 10 classic classes of Chinese architectural heritage (detailed in Section 3). We also design an image retrieval system using hash coding, which aims at high accuracy and efficient image retrieval. A CNN backbone is adopted to extract image features and a hashing module is used to realize the retrieval. Considering the real-world demands of an architectural heritage retrieval task for Jiangxi, China, we utilize a data fine-tuning strategy by first training the model on source data and then transferring the learned knowledge to target data. A small number of local area data can train a high-performance retrieval model. Our experiment results show that this learning strategy can break the data quantity restriction and better specify the backbone CNNs' feature extraction for target data. Furthermore, by demonstrating the retrieval results and using the UMAP [21] embedding, we can further observe some interesting feature relations among different classes, which are meaningful for actual uses. We also apply the proposed method to a real-world application. Based on the coordinate information of CAH10, an application of *image-to-location* is provided. This application can provide the user with a better recommendation experience and extend the use of retrieval tasks.

We summarize our main contributions as follows:

- We construct a new dataset CAH10 for traditional Chinese architectural heritage.
- We propose a deep-hashing-based retrieval method that can realize high recommendation accuracy. The analysis of the retrieved results reveals the relations among different architectural heritage categories.
- A data fine-tuning strategy is adopted to break the quantity restriction of local architectural heritage data. This strategy can also enable better image feature extraction of the retrieval model.
- For a better user experience, an application of *image-to-location* is provided for building a recommendation system.

The rest of the paper is organized as follows. Section 2 introduces the existing research. We detail the constructed dataset in Section 3. Section 4 presents notations and the model. It gives an overview of the proposed retrieval method and data specifying strategy. In Section 5, we present a performance comparison of different retrieval methods and analyze the retrieved results of the proposed method. In Section 6, we further emphasize the advantages of our method. Limitations and future works are also discussed. Finally, conclusions are drawn in Section 7.

2. Related Works

Content-based image retrieval (CBIR) is a research branch in the field of computer vision that focuses on large-scale digital image content retrieval. A typical CBIR system allows users to input an image to find other images with the same or similar content. In contrast, traditional image retrieval is text-based, i.e., the query function is implemented by the name of the image, textual information, and indexing relationships. The core of CBIR is image retrieval using the visual features of images. Essentially, it is an approximate matching technique that integrates the technical achievements of various fields, such as computer vision, image processing, image understanding, and database, in which feature extraction and indexing can be performed automatically by the computer, avoiding the subjectivity of manual description. The process of user retrieval is generally to provide a sample image (query), and then the system extracts the features of the query image, compares them with the features in the database, and returns the image similar to the query features to the user.

Remarkable progress has been made in image feature representations for CBIR [22]. Among different kinds of image retrieval methods, the main advantage of hashing retrieval is the quickness. Many hashing methods have been designed for approximate nearest-neighbor (ANN) search in Hamming space for image retrieval. Hashing-based methods map high-dimensional data into compact binary codes with a preset number of bits and generate similar binary code data items, which can greatly reduce the calculation consumption and storage space. In the early stages, hashing methods focused on data-independent methods, such as locality-sensitive hashing (LSH) and its variance [23,24]. The major drawback of LSH is that long code is necessary to achieve satisfactory search accuracy, which limits its application. Recently, the deep hashing method [25–29] has gained great performance in image hashing. CNNs are adopted as feature extractors and hash layers are applied to study hash codes. Pairwise similarity loss or triplet ranking loss are used for similarity learning. CNNH [30] adopts a two-stage strategy to maximize the Hamming distances between binary codes of dissimilar images and minimize the Hamming distances of similar images. HashNet [26] adopts the weighted maximum likelihood (WML) estimation to alleviate the severe data imbalance by adding weights in pairwise loss functions. The Hamming distance between hash codes of images is forced to be greater than a certain threshold. Works [26,30] such as HashNet directly make data pairs similar if they share at least one category.

Due to its many applications, deep-learning-based image retrieval has been applied to cultural heritage as well. Gupta et al. [17] curated a novel dataset consisting of monument images of historical, cultural, and religious importance from all over India. Their retrieval task is based on the architectural characteristic of each class and holistically infuses semantic-hierarchy-preserving embeddings to learn deep representations for retrieval in the domain of Indian heritage monuments. Sipiran et al. [18] present a benchmark of cultural heritage objects for the evaluation of 3D shape retrieval methods. Their experiments and results show that learning methods on image-based multiview representation are suitable for tackling 3D retrieval in a cultural heritage domain. Belhi et al. [31] explore the difference between CNN features and classical features for a large-scale cultural image retrieval task. Their method can quickly identify the closest clusters and then only matches images from the selected clusters. Liu et al. [32] use modern information technology to efficiently retrieve images of national costumes. This research lay a good foundation for the informatization of national costumes and is conducive to the inheritance and protection of intangible cultural heritage. Different from previous works where data are sufficient for training a deep model, in this paper, we want to deal with a real-world task in that data are minimal.

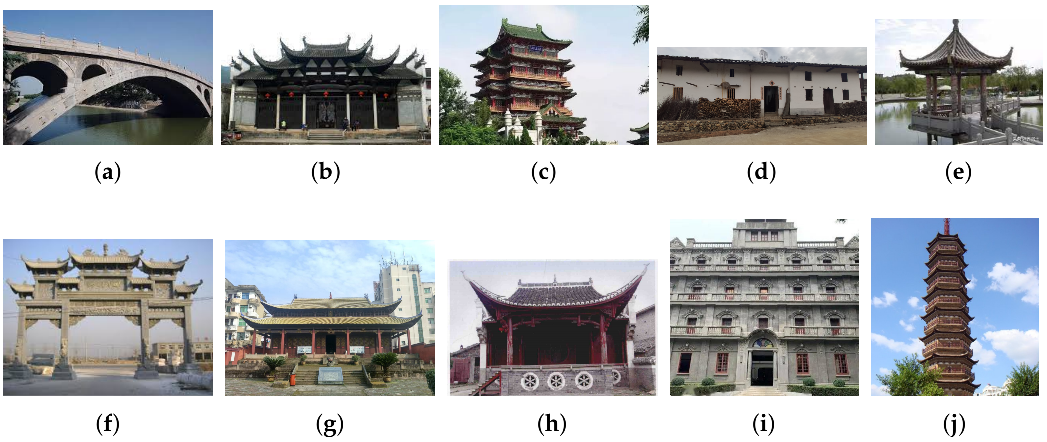
3. Data

CAH10 contains 3080 images of 10 classic classes belonging to Chinese architectural heritage. We demonstrate the image samples for each class in Figure 1a–j. The CAH10 dataset consists of three subsets: the target set, source set, and query set. Here, we briefly explain how we collect these data and what each subset serves.

- Source set: We use the images of the source set to pretrain our retrieval model. It contains 90% of the random split images from an image collection. The images from the collection are searched by using the keyword of each heritage class name and selected by a specialist based on the class definition. Selected images cover different regions and cultures; synthetic images are also included.
- Query set: The query set contains the remaining 10% of the image collection. All these images are used to evaluate the model retrieval accuracy on the source set or target set. This set also serves as the possible user input to demonstrate what will be retrieved by the model.
- Target set: This set contains the local architectural heritage images which are the entities that our retrieval system wants to recommend. We collect 285 images from Jiangxi, China. For the accessible *image-to-location* recommendation, each image is attached with its geographical coordinates. Notice that the images from the target set are excluded from both the source set and query set.

The specific data distribution of CAH10 is shown in Table 1. We can observe that the total image number of the source set is almost 10 times that of the target set. The classification of traditional Chinese architecture in this paper is based on the classification methods of architectural heritage mainly derived from the international arena, and the architectural heritage is classified according to the functions and functional categories of buildings. “Convention Concerning the Protection of The World Cultural and Natural Heritage”, “Hoi An Protocols for Best Conservation Practice in Asia”, “Burra Charter” and “International Symposium on the Concepts and Practices of Conservation and Restoration of Historic Buildings in East Asia” all require the classification of cultural relics according to the function and value of architectural heritage. In terms of the specific types of architectural heritage, we mainly refer to two laws, “Law of the Peopleís Republic of

China on Protection of Cultural Relics" and "The Regulation on the Protection of Famous Historical and Cultural Cities, Towns, and Villages". Combined with the local laws of Jiangxi Province, such as the "Regulations on the Protection of Cultural Relics in Jiangxi Province" and the "Regulations on the Protection of Revolutionary Cultural Relics in Jiangxi Province", the classification of building types was further supplemented and improved, thus summarizing the ten architectural classifications in CAH10. In these ten categories, seven categories, including bridges, residential buildings, palaces, temples, theatres, towers, and modern historic buildings, have been specifically classified in Western architecture, while ancestral halls, memorial archways, and pavilions are unique architectural forms in China. According to China's current cultural relics protection regulations, the archway is unique to China in form and similar in function to the column and triumphal arch. This classification method is based on international and Chinese cultural heritage protection and management regulations and further subdivides building types according to local regulations in Jiangxi Province.


(**a**) (**b**) (**c**) (**d**) (**e**)

(**f**) (**g**) (**h**) (**i**) (**j**)

Figure 1. (**a**–**j**) are the sample images from the classes of architectural heritage defined in CAH10. (**a**) bridge; (**b**) ancestral hall; (**c**) palace; (**d**) residential building; (**e**) pavilion; (**f**) memorial archway; (**g**) temple; (**h**) theatre; (**i**) modern historic; (**j**) tower.

Table 1. The data distribution of CAH10 dataset.

Class	Source Set	Target Set	Query Set	Total
Bridge	267	12	28	307
Ancestral hall	299	40	26	365
Palace	151	25	18	194
Residential building	463	73	51	587
Pavilion	208	13	19	240
Memorial archway	281	31	20	332
Temple	221	33	17	271
Theatre	160	14	15	189
Modern historic	256	12	28	296
Tower	246	32	21	299
Total	2552	285	243	3080

4. Methods

In order to realize a high accuracy and efficient retrieval system, we construct our retrieval model based on a deep hashing structure. We train our model by firstly using the source data and then transferring the learned knowledge to the target set.

4.1. Problem Definition and Model Structure

We denote $\mathcal{X} = \{x_i\}_{i=1}^{N}$ as the set of database images in our dataset (either source set or target set) from which images similar to an image from the query set are retrieved. The purpose of the retrieval model is to find a mapping from an image to a K-bit binary hash code $\mathcal{B} = \{b_i\}_{i=1}^{N}$, where $b_i \in \{-1, 1\}^K$. For differentiability, following previous works [26,28,29,33], instead of directly generating binary hash codes, we adopt continuous relaxation $e_i \in [-1, 1]^K$. By taking the $sign()$ of each element of e_i, the continuous hash code can be easily mapped to a binary hash code; therefore, the model learns a mapping f from x_i to e_i. The labels for image $x_i \in \mathcal{X}$ can be represented by a one-hot vector z_i, where $z_i \in C$ (C is the classes defined in CAH10) and the semantic similarity between a pair of images can be calculated by

$$s_{ij} = z_i \cdot z_j, \tag{1}$$

where "$\cdot$" is the operator for inner product. This definition quantifies a semantic similarity, taking the nature of our dataset by setting the similarity to be 0 (different class) or 1 (same class).

We demonstrate the pipeline of our model in Figure 2. Firstly, a shared CNN is used as the model backbone to extract the features for all input images. After the global average pooling, we set hash layers, which consist of two layers of full connection (FC) and are activated by $tanh(\cdot)$ nonlinearity. The hash layer serves as the module to generate continuous K-bit hash code e_i. Finally, the quantization loss will force the generated e_i close to the binary hash code b_i and the similarity loss will instruct the whole model's parameters for better feature extraction. We introduce these two losses in Section 4.2.

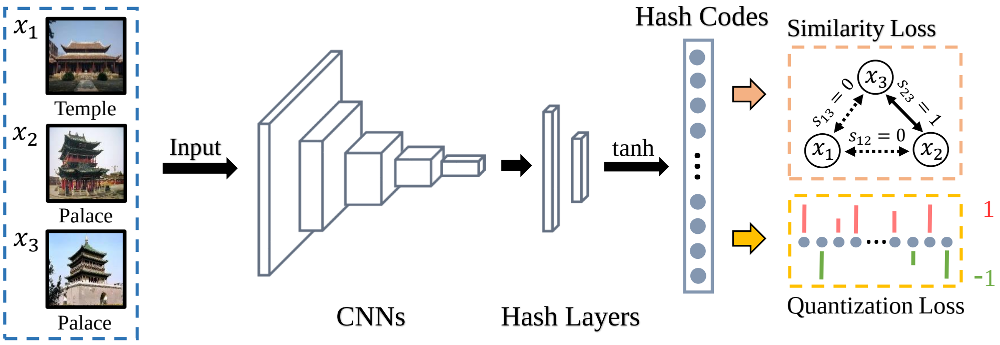


Figure 2. Structure of the proposed hashing-based retrieval model.

4.2. Learning Similarity

How to utilize the similarity between pairs is the major factor for training a retrieval model. For a pair of hash codes e_i and e_j, we use the inner product $e_i \cdot e_j$ to calculate the distance between them. As an alternative to the Hamming distance, the inner product has been proven to better quantify pairwise similarity for continuous hash codes [25,28,34,35]. In order to allow the generated hash codes e_i and e_j to well encode our label-based similarity s_{ij} for image pair (x_i, x_j), we train our mapping f for label class C.

4.2.1. Quantization Loss

By using $tanh(\cdot)$, we aim to squash the output of the hash layer to be in $[-1, 1]$. However, this activation alone does not promise that the generated hash code has values closer to either 1 or -1. We thus design the quantization loss to normalize this issue. It is given by

$$L_{\text{Quantization}} = \sum_i || |e_i| - \mathbf{1}_K ||^2, \tag{2}$$

where $|e_i|$ gives the absolute value element-wise and $\mathbf{1}_K$ is a vector whose all K elements are 1.

4.2.2. Similarity Loss

We let $\mathcal{S}$ and $\mathcal{D}$ as the sets of image indices pairs (i, j) whose labels are the same (i.e., $s_{ij} = 1$) or different (i.e., $s_{ij} = 0$), respectively. Pairs in these sets give a signal that corresponding hash codes e_i and e_j are far from or close to each other. To encode this, similar to HashNet [26], we define the probability of similarity given a pair of hash codes as

$$p(s_{ij} \mid e_i, e_j) = \begin{cases} \sigma(e_i \cdot e_j) & \text{for } (i,j) \in \mathcal{S} \\ 1 - \sigma(e_i \cdot e_j) & \text{for } (i,j) \in \mathcal{D} \end{cases} \tag{3}$$

$\sigma(\cdot) \in [0, 1]$ is the activation function of sigmoid. Generally, the quantity of image pairs with different labels is far more than those with same labels. We thus apply a weight w_{ij} that gives ϕ for $(i, j) \in \mathcal{S}$ and $1 - \phi$ for $(i, j) \in \mathcal{D}$ to relieve the pair imbalance. ϕ is simply calculated by

$$\phi = D/(D + S) \tag{4}$$

By using the cross-entropy, the final similarity loss function is defined as

$$L_{\text{Similarity}} = - \sum_{(i,j) \in \mathcal{S} \cup \mathcal{D}} w_{ij} \log p(s_{ij} \mid e_i, e_j). \tag{5}$$

4.2.3. Overall Loss

The overall loss is defined by the combination of the quantization loss and similarity loss as follows:

$$L = \lambda L_{\text{Quantization}} + L_{\text{Similarity}} \tag{6}$$

where λ is the bias to control the importance of the quantization loss.

4.3. Data Fine-Tuning and Image Retrieval

As shown in Figure 3's training part, we firstly train our model by the source set. Although the image number in the source set is almost ten times that of the target set, we adopt data augmentation for each input for better model training [36,37]. The augmentation includes random horizontal flip, color jitter, rotation, translation, cropping, Gaussian noise, and blurring. After the model is well trained by the source set, we use the target set to fine-tune the model starting with a small learning rate (also using data augmentation in this phase). In this step, the knowledge learned from the source set can be transferred to the training of the target set, which enables better retrieval performance for the target set.

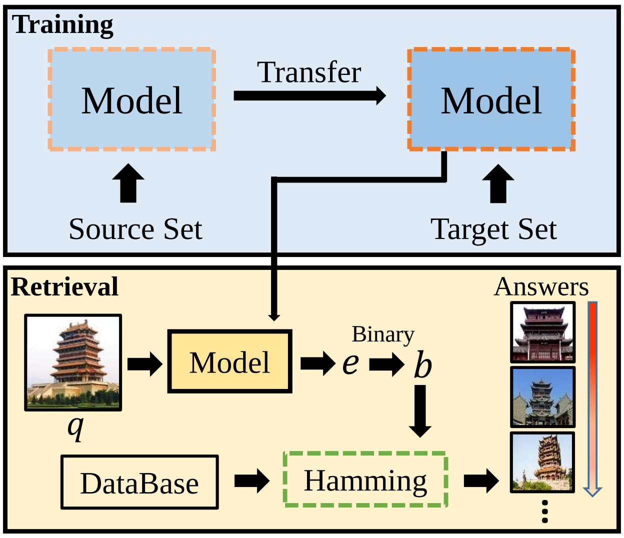


Figure 3. The flowchart of knowledge transfer and image retrieval.

In the retrieval part, considering the situation that the user uploads one query image q, the model will first extract the continuous hash codes e. In order to follow the definition of hash retrieval, e will be processed into binary code b using the threshold 0. Then, the Hamming distance will calculate the similarity between b and each image's binary code pre-extracted from the database. The database is the image collection for searching for similar images. It could be either the source set or target set in our experiments. With the similarity order, the model can return the top n similar images as the retrieval answers from the database. All these answers are recommendations to the user based on visual similarity.

5. Results

5.1. Implementation Details

We adopt ResNet-50 [38] as the CNN backbone. It is important to notice that a backbone with a larger size (e.g., ResNet-101) may improve the retrieval performance. However, we do not have much data (compared to ImageNet [39]) and a small backbone is enough for our dataset [40]. We also demonstrate the choice of backbone model for our method in Section 5.4.

For all the training settings, we resize the input image to 260 × 260 and adopt AdamW [41] as the optimizer. During the training of the source set, we randomly initialize the model parameters and run a total of 20 training epochs. The initial learning is default to 10^{-3} and this rate is decreased by a factor of 10 halfway through the training process. For the continuous fine-tuning of the target set, we initialize the model with the parameter pretrained on the source set. We set the learning rate to start with as small a value as 5^{-4} and totally run 10 training epochs. All the experiments are implemented by a Tesla V100 GPU.

5.2. Evaluation Metrics

For evaluating the performance of image retrieval, we adopt two widely used metrics: average cumulative gains (ACG) [42] and mean average precision (MAP) [43].

ACG is used to compute the average number of shared class labels between the top n retrieved images and the query image. Given a query image I_q, the ACG score is calculated by

$$ACG@n = \frac{1}{n}\sum_{i}^{n} C(q,i), \tag{7}$$

where $C(q,i)$ denotes the number of shared class labels between I_q and I_i. n denotes the number of the top n retrieval images.

MAP is used to calculate the mean of average precision for each query image by

$$MAP@n = \frac{1}{Q}\sum_{q}^{Q} AP(q), \tag{8}$$

where

$$AP(q) = \frac{1}{N_{Re}@n}\sum_{i}^{n}(Re(q,i)\frac{N_{Re}(q)@i}{i}), \tag{9}$$

and $Re_{(q,i)} \in \{0,1\}$ is an indicator. $Re(q,i) = 1$ when I_q and I_i have same class labels; otherwise, $Re(q,i) = 0$. $N_{Re}(q)@i$ denotes the number of images with respect to the query image I_q among the top i retrieval images. Q is the image number of the query set.

5.3. Retrieval Performance

In this section, we compare the performance of the proposed hashing retrieval method (using ResNet-50 as a backbone) with some previous works: unsupervised methods SH [44] and LSH [23]; supervised methods BRE [45], SDH [46], and ITQ-CCA [47]; and supervised deep methods CNNH [30], DNNH [48], DHN [25], and HashNet [26]. Both supervised and unsupervised DL methods try to discriminate the image similarity through mathematical functions or predefined kernels. The advantages of them are the simple model structure and implementation speed. Supervised deep methods are based on the deep neural network structure, which requires much more training efforts than other types. However, the retrieval performance is obviously improved. As shown in Table 2, we implement this comparison using the source set as a database (all methods only use the source set for training) and all images in the query set as a query. It can be observed that the proposed method outperforms all the previous works (0.018 more for MAP@50 48-bit than HashNet), which demonstrates the superiority of our method for the CAH10 dataset.

Table 2. Comparison to previous retrieval methods using ACG@50 and MAP@50 on training set. The best results for each metric are marked in bold.

Method	**ACG@50**				**MAP@50**			
	16-bit	**32-bit**	**48-bit**	**64-bit**	**16-bit**	**32-bit**	**48-bit**	**64-bit**
SH [44]	0.421	0.514	0.585	0.602	0.447	0.539	0.605	0.621
LSH [23]	0.308	0.429	0.450	0.483	0.325	0.444	0.487	0.512
BRE [45]	0.332	0.395	0.522	0.551	0.356	0.422	0.545	0.562
SDH [46]	0.411	0.452	0.555	0.512	0.432	0.464	0.572	0.530
ITQ-CCA [47]	0.500	0.543	0.585	0.561	0.519	0.560	0.601	0.573
CNNH [30]	0.618	0.656	0.701	0.715	0.632	0.673	0.714	0.732
DNNH [48]	0.593	0.677	0.717	0.725	0.611	0.692	0.733	0.735
DHN [25]	0.580	0.674	0.688	0.705	0.595	0.693	0.702	0.726
HashNet [26]	0.720	0.775	0.762	0.745	0.728	0.795	0.770	0.742
Ours	**0.731**	**0.792**	**0.771**	**0.752**	**0.738**	**0.812**	**0.788**	**0.765**

5.4. Results Comparison with Different Model Settings

In this section, we discuss the performance difference of using a combination of the following four aspects: (1) backbone model default as ResNet-50, (2) the length of the hash code $K = 32$, (3) loss function bias $\lambda = 0.1$, and (4) the number of the top retrieval images $n = 10$. For all the experiments, we pretrain the model on the source set and adopt data fine-tuning on the target set. The evaluation metrics are calculated by the results retrieved from the target set using all the images from the query set as queries. We conduct this study by using the default setting except for the one to be explored.

We first analyze the best backbone for our model. As shown in Table 3, we compare the performance of different-capacity submodels of ResNet and some other types of backbone CNNs, such as DenseNet [49], EfficientNet [50], and Inception-V4 [51]. The best ACG and MAP for all bit settings are achieved by ResNet-50. It outperforms its bigger size variance ResNet-101 by about 0.01–0.03, which may imply that a medium-size model is enough for our dataset. ResNet-50 also obtains higher performance than other types of CNN backbones. We think the basic structure of ResNet can provide a better image feature for continuous hashing computing.

As shown in Table 4, we can find that the length of hash code K is an important setting for retrieval performance. It is designed as the index for searching similar images from the target set. The best results are achieved by using K as 32-bit. Generally, a larger K gives a better retrieval performance. However, for our dataset, a small length of hash code is enough. We can observe that there is an obvious performance drop (about 0.05) when setting K as 64-bit.

Table 3. Backbone selection for our model. The best results for each metric are marked in bold.

Backbone	ACG@10				MAP@10			
	16-bit	**32-bit**	**48-bit**	**64-bit**	**16-bit**	**32-bit**	**48-bit**	**64-bit**
ResNet-18	0.753	0.788	0.755	0.724	0.767	0.801	0.770	0.736
ResNet-50	**0.811**	**0.837**	**0.822**	**0.801**	**0.816**	**0.847**	**0.838**	**0.795**
ResNet-101	0.792	0.803	0.788	0.770	0.809	0.813	0.799	0.783
DenseNet-121	0.712	0.749	0.740	0.710	0.723	0.765	0.758	0.721
EfficientNet-b4	0.795	0.821	0.791	0.758	0.808	0.833	0.812	0.787
Inception-V4	0.704	0.741	0.733	0.702	0.711	0.758	0.741	0.713

Table 4. ACG@10 and MAP@10 performance of retrieval model with different settings of λ and K bits. The best results for each metric are marked on bold.

λ	ACG@10				MAP@10			
	16-bit	**32-bit**	**48-bit**	**64-bit**	**16-bit**	**32-bit**	**48-bit**	**64-bit**
0	0.771	0.804	0.795	0.772	0.787	0.809	0.805	0.781
0.01	0.808	0.829	0.820	0.783	0.795	0.814	0.817	0.788
0.1	0.811	**0.837**	0.822	0.801	0.816	**0.847**	0.838	0.795
1.0	0.800	0.824	0.817	0.785	0.801	0.821	0.810	0.780
10.0	0.721	0.755	0.741	0.718	0.733	0.763	0.748	0.711

λ is another value that affects model performance. By using a larger λ, the generated continuous hash code e will be close to the binary code b. However, due to the activation function $tanh()$, getting close to 1 or -1 will cause the disappearance of the gradient, which makes model training much more difficult. Based on the experiment results, we can find that setting $\lambda = 0.1$ gives the best retrieval performance on both evaluation metrics. A very small value of λ will cause a drop of about 0.03 for MAP, but it will not vary a lot. When we increase the value to 10, the MAP drops drastically (about 0.08 down in performance).

By the analysis above, we can conclude that $K = 32$ and $\lambda = 0.1$ are the best parameters for our dataset. Thus, within these parameters, we explore the results of different numbers for top retrieval answer n. As counted in Table 1, the smallest class (bridge) only owns 12 images. Thus, we set the n maximum as 10. As shown in Table 5, both ACG and MAP metric results demonstrate no obvious fluctuation with the change in n (from $n = 2$ to $n = 10$). There is only a 0.026 MAP performance drop, which shows the robustness of our proposed method.

Table 5. The performance of different numbers of top n retrieval answers (using model trained with λ = 0.1 and K = 32).

n	ACG@n	MAP@n
2	0.807	0.821
4	0.816	0.825
6	0.824	0.833
8	0.830	0.838
10	0.832	0.847

5.5. Top Retrieval Results

In Figure 4, we show the retrieval results of ten samples (one sample per class) randomly selected from the query set. The images in the left blue box are the samples of each class and the images in the left green box are the corresponding top 10 retrieval answers. The images colored red are the wrong retrieval answers (we attach the actual classes under the images). For the query image of landmark buildings (e.g., pavilion), the retrieval answers are all correct. They are easy to be discriminated against due to their unique structures. However, we can also observe some confusing classes. Ancestral hall, temple, and theatre are classes similar to each other in both appearance (pointed eaves) and functionality (used for cultural events). Most of these three types of architectural heritage are located in Chinese rural complexes, which mainly serve for cultural activities such as rituals. Moreover, all of them are designed in architectural design with flying eaves, so it is difficult to distinguish them by only access to the image taken from the exterior appearance. Even humans cannot directly tell the difference between them. However, we think it is not a drawback for the retrieval system. Sometimes users are not concerned about the actual class of the uploaded query image. They just want to have more recommendations of architecture that share similar vision characteristics. We further discuss the feature relations among classes in Section 6.1.

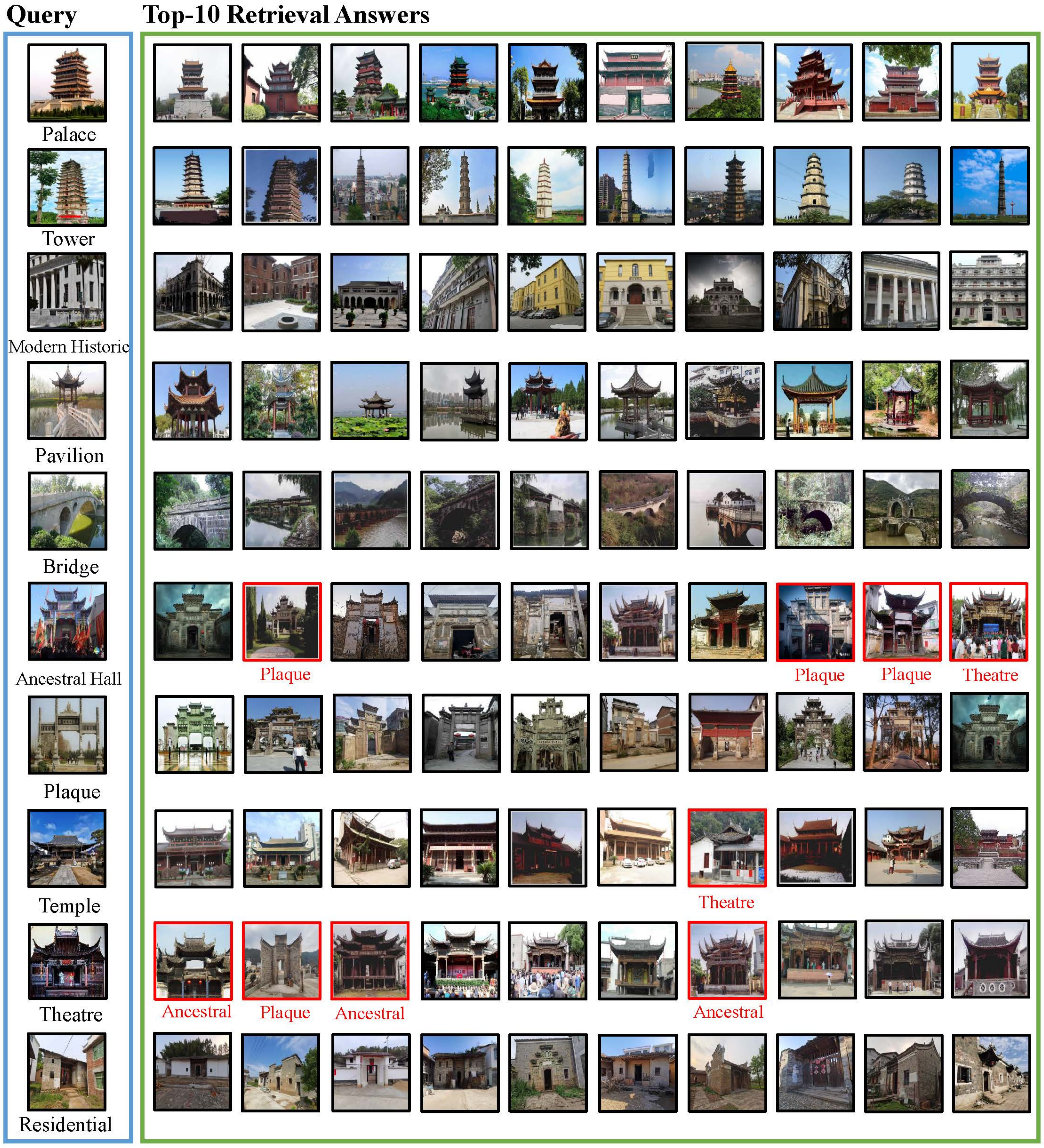


Figure 4. The demonstration of retrieved image samples.

6. Discussion

6.1. Advantages of Data Fine-Tuning

One contribution of this paper is applying data fine-tuning for better training of a local architecture heritage retrieval system. Thus, it is important to verify the impact of this method. We demonstrate the advantages of data fine tuning by a quantitative evaluation to show how fine-tuning can break thedata quantity restrictions of local data. We also make a feature space visualization in this section, which proves that the data fine-tuning can enable better image feature extraction of the retrieval model.

Break Data Quantity Restriction: We first analyze the results from the quantitative level. Here, we denote "*sole*" as the model only trained with the target set, "*w/o*" as the model only trained with the source set, and "*w/*" as the model trained with the source set and fine-tuned with the target set (the proposed strategy in this paper). As shown in Table 6, all the experiments are implemented by the setting of $\lambda = 0.1$ and $n = 10$. We can observe that the performance of *sole* is quite bad, which is consistent with the issue that we discussed in the introduction part (it is difficult to train a deep model when the quantity of data is small). In addition, there is also an obvious performance difference between "*w/o*" and "*w/*", which illustrates the diverse feature distribution between the source set and target set. Using knowledge transfer can train a retrieval model for a better recommendation of local architecture heritages.

Table 6. The performance of the model trained solo, *w/o* knowledge transfer, and *w/* knowledge transfer.

K **bits**	**ACG@10**			**MAP@10**		
	sole	***w/o***	***w/***	***sole***	***w/o***	***w/***
16	0.498	0.765	0.802	0.512	0.772	0.816
32	0.541	0.800	**0.834**	0.554	0.812	**0.847**
48	0.539	0.771	0.825	0.550	0.788	0.838
64	0.533	0.710	0.780	0.545	0.735	0.795

Better Image Feature Extraction: Mapping images into two-dimensional space is a commonly used way for human-understandable data visualization. In order to show the image features extracted by the backbone model, we use UMAP [21] to realize the downscaling for the output vector of the CNN backbone. Shown in Figure 5, we compare the mapping results between "*w/o*" and "*w/*". We can observe that for the visualization of "*w/o*", the dots of different classes are aggregated closely to each other, while "*w/*" provides a more distinguished feature distribution. This proves the effects of data fine-tuning for better feature extraction. We can also find the feature similarities of the confusing classes mentioned in Section 5.5. They overlap with each other and the model has difficulty precisely separating them apart. However, such close distribution gives the model a chance to retrieve diverse results, which may satisfy the users' demands.

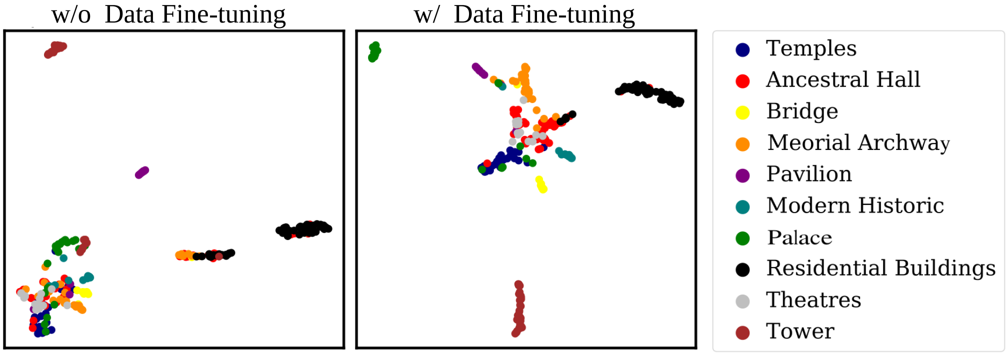


Figure 5. The visualization of UMAP embedding for the feature distribution of images from the target set. The image on the right is the result of the model using data fine-tuning and the image on the left is the result of the model only trained by source data.

6.2. Retrieval with Geographical Coordinates

How to apply the proposed method to a real-world application is important. Here, we provide one possible application of our method. In CAH10, all the images in the target set have their coordinates. Thus, we could build an online recommendation system that could return retrieved images and locate them on the geography map. For example, in Figure 6, we demonstrate the application of *image-to-location*. By analyzing one image (e.g., tower) uploaded by the user, the retrieval system can show similar architectures and mark them on the map. We believe this application can be easily transplanted into some popular online-map services, which enable better recommendation of the local architectural heritage to users.

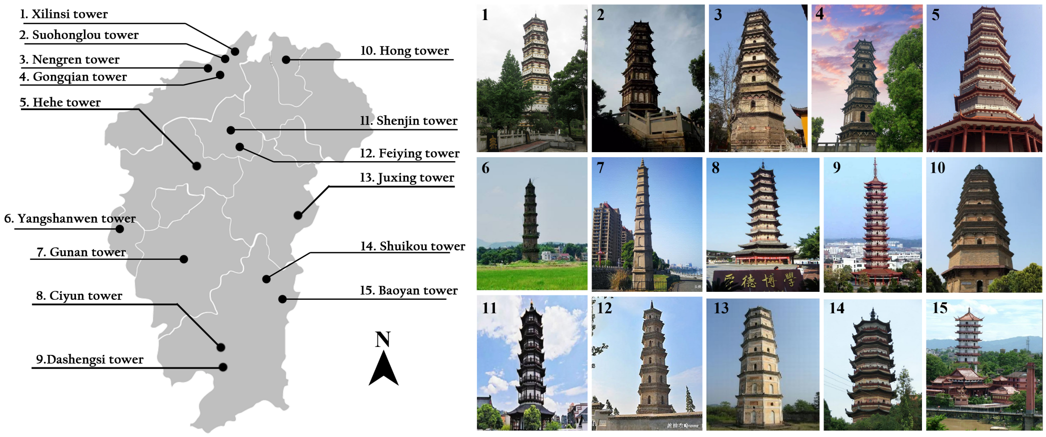


Figure 6. Application of image-to-location (Jiangxi, China).

6.3. Limitations and Future Works

CAH10 currently contains 10 classic Chinese architectural heritages. It would be interesting to include more classes to realize a large-scale dataset. Not only limited to traditional Chinese architecture, our future goal is also to collect more types of architectural heritages all over the world. By doing this, we can analyze the relationship between architectural styles in different regions and different cultures and then formulate more meaningful recommendations. Images can only stand as parts of architectural heritage, and the full description of a heritage entity will be much more complex. To realize a comprehensive recommendation system, we also plan to use text data (tourist comments) together with images. It is a VQA (vision question and answer) task with better interactivity.

As we discussed in Section 5.5, recognition difficulty exists among some classes. In our dataset, all the images are assigned with one certain class label; however, some images may belong to multiple classes. For example, some residential buildings in rural China also serve as ancestral halls. Such data characters will lead the current retrieval problem into a multilabel task which may enhance the performance of the recommendation. However, a lot of annotation contribution is required for a multilabel dataset. We will take it as our future research.

The hierarchy data structure of architectural heritages is not discussed in this paper (e.g., the class bridge has some subclasses). Based on the results of previous works [33], we think this is important information to regulate feature extraction during training. We will also complement our dataset with a hierarchy label and explore the impact of hierarchy in future research.

7. Conclusions

In this paper, we designed a hashing-based image retrieval method for local architectural heritage recommendations. Our work extends the DL research in the architectural heritages area, which helps interested researchers follow and pursue related research. In addition, our method aims at a real-world scene where only a few data values are available for local architectural heritage. By utilizing the data fine-tuning strategy, even a small quantity of data can realize the training of a high-performance model. Taking Jiangxi, China, as a case study, a new dataset CAH10 was proposed to evaluate the reliability of the proposed methods. We also compared our method with several popular retrieval methods and the results demonstrate our model's superiority. Furthermore, we analyzed the character of classic Chinese architectural heritages with the proposed dataset using a UMAP dimensionality reduction. The experiment results show interesting relations among classes which could be further related to the recommendation task. An advanced retrieval model can be built based on this finding. Instead of image-level labeling, the retrieval task can be regularized by some common concepts among different classes of architectural heritage. Our research is meaningful for the promotion and protection of local architectural heritages. We will continue our work by enlarging the dataset and introducing more state-of-the-art technologies.

Author Contributions: Conceptualization, K.M., B.W., Y.L. and J.Z.; coding, K.M. and B.W.; writing—review and editing, K.M. and B.W.; project administration, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available by contacting corresponding author. Our dataset and code are available at https://github.com/wbw520/CAH10.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

References

1. Marty, P.F. Digital convergence and the information profession in cultural heritage organizations: Reconciling internal and external demands. *Libr. Trends* **2014**, *62*, 613–627. [CrossRef]
2. Yilmaz, H.M.; Yakar, M.; Gulec, S.A.; Dulgerler, O.N. Importance of digital close-range photogrammetry in documentation of cultural heritage. *J. Cult. Herit.* **2007**, *8*, 428–433. [CrossRef]
3. Navarrete, T. Digital cultural heritage. In *Handbook on the Economics of Cultural Heritage*; Edward Elgar Publishing: Cheltenham, UK, 2013.
4. Calvanese, V.; Zambrano, A. A Conceptual Design Approach for Archaeological Structures, a Challenging Issue between Innovation and Conservation: A Studied Case in Ancient Pompeii. *Buildings* **2021**, *11*, 167. [CrossRef]
5. Tejedor, B.; Lucchi, E.; Bienvenido-Huertas, D.; Nardi, I. Non-Destructive Techniques (NDT) for the diagnosis of heritage buildings: Traditional procedures and futures perspectives. *Energy Build.* **2022**, *263*, 112029. [CrossRef]
6. Zou, Z.; Zhao, X.; Zhao, P.; Qi, F.; Wang, N. CNN-based statistics and location estimation of missing components in routine inspection of historic buildings. *J. Cult. Herit.* **2019**, *38*, 221–230. [CrossRef]
7. Condorelli, F.; Rinaudo, F.; Salvadore, F.; Tagliaventi, S. A Neural Networks Approach to Detecting Lost Heritage in Historical Video. *Isprs Int. J.-Geo-Inf.* **2020**, *9*, 297. [CrossRef]
8. Gumbarević, S.; Milovanović, B.; Gaši, M.; Bagarić, M. Application of Multilayer Perceptron Method on Heat Flow Meter Results for Reducing the Measurement Time. *Eng. Proc.* **2020**, *2*, 29.
9. Bienvenido-Huertas, D.; Rubio-Bellido, C.; Pérez-Ordóñez, J.L.; Moyano, J. Optimizing the evaluation of thermal transmittance with the thermometric method using multilayer perceptrons. *Energy Build.* **2019**, *198*, 395–411. [CrossRef]
10. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
11. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
13. Zhou, Y.; Liang, Y.; Pan, Y.; Yuan, X.; Xie, Y.; Jia, W. A Deep-Learning-Based Meta-Modeling Workflow for Thermal Load Forecasting in Buildings: Method and a Case Study. *Buildings* **2022**, *12*, 177. [CrossRef]
14. Kim, J.; Yum, S.; Son, S.; Son, K.; Bae, J. Modeling Deep Neural Networks to Learn Maintenance and Repair Costs of Educational Facilities. *Buildings* **2021**, *11*, 165. [CrossRef]
15. Llamas, J.; M Lerones, P.; Medina, R.; Zalama, E.; Gómez-García-Bermejo, J. Classification of architectural heritage images using deep learning techniques. *Appl. Sci.* **2017**, *7*, 992. [CrossRef]

16. Yoshimura, Y.; Cai, B.; Wang, Z.; Ratti, C. Deep learning architect: Classification for architectural design through the eye of artificial intelligence. In *Computational Urban Planning and Management for Smart Cities. CUPUM 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 249–265.
17. Gupta, R.; Mukherjee, P.; Lall, B.; Gupta, V. Semantics Preserving Hierarchy based Retrieval of Indian heritage monuments. In Proceedings of the 2nd Workshop on Structuring and Understanding of Multimedia Heritage Contents, Seattle, WA, USA, 12–16 October 2020; pp. 5–13.
18. Sipiran, I.; Lazo, P.; Lopez, C.; Jimenez, M.; Bagewadi, N.; Bustos, B.; Dao, H.; Gangisetty, S.; Hanik, M.; Ho-Thi, N.P.; et al. SHREC 2021: Retrieval of cultural heritage objects. *Comput. Graph.* **2021**, *100*, 1–20. [CrossRef]
19. Oyedare, T.; Park, J.M.J. Estimating the required training dataset size for transmitter classification using deep learning. In Proceedings of the 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 11–14 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–10.
20. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *arXiv* **2014**, arXiv:1411.1792.
21. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
22. Chen, W.; Liu, Y.; Wang, W.; Bakker, E.M.; Georgiou, T.; Fieguth, P.W.; Liu, L.; Lew, M.S. Deep Image Retrieval: A Survey. *arXiv* **2021**, arXiv:2101.11282.
23. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. In Proceedings of the 25th VLDB Conference, Edinburgh, UK, 7–10 September 1999; Volume 99, pp. 518–529.
24. Raginsky, M.; Lazebnik, S. Locality-sensitive binary codes from shift-invariant kernels. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1509–1517.
25. Zhu, H.; Long, M.; Wang, J.; Cao, Y. Deep hashing network for efficient similarity retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
26. Cao, Z.; Long, M.; Wang, J.; Yu, P.S. Hashnet: Deep learning to hash by continuation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5608–5617.
27. Cao, Y.; Long, M.; Liu, B.; Wang, J. Deep cauchy hashing for hamming space retrieval. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1229–1237.
28. Zhang, Z.; Zou, Q.; Lin, Y.; Chen, L.; Wang, S. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. *IEEE Trans. Multimed.* **2019**, *22*, 540–553. [CrossRef]
29. Yuan, L.; Wang, T.; Zhang, X.; Tay, F.E.; Jie, Z.; Liu, W.; Feng, J. Central similarity quantization for efficient image and video retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 3083–3092.
30. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. In Proceedings of the AAAI, QuEbec City, QC, Canada, 27–31 July 2014.
31. Belhi, A.; Bouras, A. CNN Features vs. Classical Features for Largescale Cultural Image Retrieval. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 95–99. [CrossRef]
32. Liu, E. Research on image recognition of intangible cultural heritage based on CNN and wireless network. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–12. [CrossRef]
33. Wang, B.; Li, L.; Nakashima, Y.; Yamamoto, T.; Ohshima, H.; Shoji, Y.; Aihara, K.; Kando, N. Image Retrieval by Hierarchy-aware Deep Hashing Based on Multi-task Learning. In Proceedings of the 2021 International Conference on Multimedia Retrieval, Taipei, Taiwan, 21–24 August 2021.
34. Cao, Y.; Long, M.; Wang, J.; Zhu, H.; Wen, Q. Deep quantization network for efficient image retrieval. In Proceedings of the Thirtieth AAAI Conference, Phoenix, AZ, USA, 12–17 February 2016.
35. Zhang, J.; Fukuda, T.; Yabuki, N. Development of a City-Scale Approach for Façade Color Measurement with Building Functional Classification Using Deep Learning and Street View Images. *ISPRS Int. J.-Geo-Inf.* **2021**, *10*, 551. [CrossRef]
36. Mikołajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujscie, Poland, 9–12 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 117–122.
37. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA , 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
40. Zhang, Y.; Ling, C. A strategy to apply machine learning to small datasets in materials science. *NPJ Comput. Mater.* **2018**, *4*, 25. [CrossRef]
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Järvelin, K.; Kekäläinen, J. IR Evaluation Methods for Retrieving Highly Relevant Documents. Available online: https://dl.acm.org/doi/abs/10.1145/3130348.3130374 (accessed on 9 May 2022).

43. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 1999; Volume 463.
44. Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 1753–1760.
45. Kulis, B.; Darrell, T. Learning to hash with binary reconstructive embeddings. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1042–1050.
46. Shen, F.; Shen, C.; Liu, W.; Tao Shen, H. Supervised discrete hashing. In Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 37–45.
47. Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 2916–2929. [CrossRef]
48. Lai, H.; Pan, Y.; Liu, Y.; Yan, S. Simultaneous feature learning and hash coding with deep neural networks. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3270–3278.
49. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
50. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
51. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.