Deep supervised hashing for gait retrieval [version 1; peer review: awaiting peer review]

Shohel Sayeed, Pa Pa Min, Thian Song Ong

Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, Melaka, 75450, Malaysia

First published: 12 Oct 2021, 10:1038
https://doi.org/10.12688/f1000research.51368.1
Latest published: 12 Oct 2021, 10:1038
https://doi.org/10.12688/f1000research.51368.1

Abstract

Background: Gait recognition is perceived as the most promising biometric approach for future decades especially because of its efficient applicability in surveillance systems. Due to recent growth in the use of gait biometrics across surveillance systems, the ability to rapidly search for the required data has become an emerging need. Therefore, we addressed the gait retrieval problem, which retrieves people with gaits similar to a query subject from a large-scale dataset.

Methods: This paper presents the deep gait retrieval hashing (DGRH) model to address the gait retrieval problem for large-scale datasets. Our proposed method is based on a supervised hashing method with a deep convolutional network. We use the ability of the convolutional neural network (CNN) to capture the semantic gait features for feature representation and learn the compact hash codes with the compatible hash function. Therefore, our DGRH model combines gait feature learning with binary hash codes. In addition, the learning loss is designed with a classification loss function that learns to preserve similarity and a quantization loss function that controls the quality of the hash codes.

Results: The proposed method was evaluated against the CASIA-B, OUI-SIR-LP, and OUI-SIR-MVLP benchmark datasets and received the promising result for gait retrieval tasks.

Conclusions: The end-to-end deep supervised hashing model is able to learn discriminative gait features and is efficient in terms of the storage memory and speed for gait retrieval.

Keywords

Gait Retrieval, Deep Supervised Hashing, Convolutional Neural Network, Binary codes

This article is included in the Research Synergy Foundation gateway.
Corresponding author: Shohel Sayeed (shohel.sayeed@mmu.edu.my)

Author roles: Sayeed S: Supervision, Writing – Original Draft Preparation, Writing – Review & Editing; Min PP: Methodology, Resources, Validation, Visualization, Writing – Original Draft Preparation; Ong TS: Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Sayeed S et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Sayeed S, Min PP and Ong TS. Deep supervised hashing for gait retrieval [version 1; peer review: awaiting peer review] F1000Research 2021, 10:1038 https://doi.org/10.12688/f1000research.51368.1

First published: 12 Oct 2021, 10:1038 https://doi.org/10.12688/f1000research.51368.1
**Introduction**

Gait recognition is perceived as the most promising biometric approach among behavioural biometric approaches, especially because of its efficient applicability in surveillance systems. However, gait recognition and identification tasks are becoming increasingly difficult due to the large-scale images and videos generated from surveillance systems. To ease the burden of real-world large dataset problems, researchers have applied person re-identification and retrieval approaches for surveillance video analysis or identify the person of interest. The re-identification approach finds the targeted person given by a query image from different cameras and a different time. Similar to re-identification, gait retrieval is also used to retrieve people with similar gait from a large-scale dataset given by the query subject. In contrast, gait re-identification considers the one-to-one problem with the top-ranked item within the dataset and from the dataset, and the retrieval normally considers all the top items from the ranked list (from one to many). The retrieval problem is addressed in many biometric applications, such as face retrieval and other large-scale content searches, because of the efficiency in tracking and locating similar content.

To retrieve visually or semantically similar content, the traditional approach is to search for similar contents by ranking the contents from the database based on the similarity with the query features, and the nearest contents are returned. Nevertheless, this approach affects the computation time and memory of large-scale databases. To address these speed and storage issues, hashing methods have been proposed for use in different text, video, and image retrieval tasks.

This paper presents the deep gait retrieval hashing (DGRH) model to address the gait retrieval problem for large-scale datasets. Due to the recent growth in gait data across surveillance systems, the ability to rapidly search for the required data has become an emerging need. Linearily searching for real-value features may affect the computation time and memory storage, so the approximate nearest neighbour (ANN) search approach via hashing has attracted increasing attention. The goal of the hashing approach is to represent the input images as hash codes and learn the similarity of the learned binary codes. Similarity searching can be efficiently implemented when the high-dimensional data are transformed into compact binary codes with hashing functions. Our proposed method used the supervised hashing method with a deep convolutional network. The supervised hashing approach takes label information of each gait to generate binary codes. To extract the discriminant features from the input gait, we use a deep convolutional neural network instead of manually extracting the features. We use the convolutional neural network (CNN) to capture the semantic gait features for feature representation and learn the compact hash codes with the compatible hash function. Therefore, our DGRH model is a supervised hashing model that combines gait feature extraction and binary hash code learning. The pipeline for our proposed (DGRH) model includes:

- The sub-network of convolutional layers and pooling layers is used to extract the gait features
- The binary hash codes are generated from the last fully connected layer
- The classification and quantization loss are used to optimize the network and learn the hash function

**Related work**

Similarity searching can be efficiently implemented with compact binary codes generated with hashing methods. The current hashing methods can be divided into unsupervised and supervised hashing. Unsupervised hashing methods use unlabelled data and only training data to learn the hash function and perform neighbourhood relation clustering in a Hamming space. For example, kernelized locality-sensitive hashing uses the random projection of the hash function and constructs the kernel function to perform the similarity search. Liu et al. proposed anchor graph hashing to build a neighbourhood graph that learns binary hash codes to map similarities in a Hamming space. Neighbourhood discriminant hashing (NDH) utilizes the local discriminant information from the neighbourhood structure so that the data labels can be predicted in a Hamming space.

To reduce the complexity and perform efficient semantic similarity searching, supervised hashing approaches are practised. Supervised hashing takes advantage of label information, pairwise similarity information, or data point similarity. Liu et al. proposed supervised hashing with kernels by using image pairs and converted them into binary code to map the data in Hamming distance. The kernel formulation minimizes the distance between similar pairs and maximizes the distance on dissimilar pairs. Shen et al. also introduced supervised discrete hashing, which combines linear classification with the generation of hash codes into a model and addresses supervised hashing problems. Then, Lin et al. proposed supervised hashing with a two-step model: binary code learning in the first step and hash function learning in the second step.

The above papers used manual feature learning, which has limitations on the diversity of the dataset and the performance. Therefore, current researchers utilize the advantages of deep networks, which can extract visual features from raw data.
with minimal pre-processing. Xia et al. introduced a two-step paradigm for supervised hash learning with a convolutional neural network. They pre-processed the input images with a pairwise similarity matrix to give the approximated hash codes for training images and then used a convolutional neural network (AlexNet) for the feature learning of input images as well as for learning the hash codes. However, a CNN has limitations for improving hash code learning, and the one-stage method became the norm of later deep supervised hashing methods. Other researchers proposed deep supervised hashing techniques for different kinds of inputs, such as single data, paired data, and triplet pairs of data. Lai et al. proposed a deep hashing architecture with triplet pair images as the input and convolutional layers to extract effective image features. They used the divide and encode module to extract the image features into branches that correspond to each hash code, and then the triplet ranking loss function was used to perceive the similarities. Thereafter, Zhang et al. were inspired to use this approach for their person re-identification problem because the optimization of triplet ranking is able to capture the variation differences between the intra-class and inter-class rankings. The supervised hashing of semantic similarity based on data pairs has also received attention because of improvements in the quality of hash coding. The deep hashing network (DNH) proposed by uses paired data for image representation and a convolutional neural network for feature extraction. The last layer of the deep network, a fully connected layer, is used to generate binary hash codes. To preserve the similarity between the pairs of images, the pairwise cross-entropy loss is adopted, and the pairwise quantization loss is adopted to control the quality of the hash code. The improved version of pairwise deep hashing is presented in DCH (deep Cauchy hashing), which uses the Cauchy distribution to design the pairwise cross-entropy loss. The Cauchy cross-entropy loss is adapted from the Bayesian framework and well designed for the Hamming space retrieval.

To address the gait retrieval problem, Zhou et al. presented the kernel-based semantic hashing method and used the Gaussian kernel function to map the gait data into the hashing function. The learned hashing function is later optimized by the triplet ranking loss, and the binary codes of gait data are stored in a database. To retrieve the given query data, the semantic ranking list is obtained based on the Hamming distance between the query data and the gait database. Rauf et al. also proposed deep supervised hashing for gait retrieval using triplet pair gait data. They designed the hash model with a three-channel convolutional neural network sharing the same parameter. The hash layer is added after fully connected layers to generate the hash code. The triplet ranking loss is used for optimization, and the associated ranking list is based on labels. Their method outperformed other traditional methods because of the robustness of the CNN in visual feature learning.

**Proposed method**

**Gait representation**

As shown in Figure 1, we use the gait energy image (GEI). The gait energy image is a spatiotemporal gait representation that represents gait features in a single image. GEIs convert a sequence of gait silhouettes into a two-dimensional image that preserves the human motion. GEIs were first introduced by to reduce the burden of limited gait training templates. Since the GEI can capture both temporal and spatial information, it has become the most popular gait representation. In addition, GEIs also include information on both the silhouette shape and the dynamic walking motion. GEIs can be obtained by extracting the silhouette of the human and averaging the sequence of silhouettes. The details of computing the gait energy image can be found in.

**Proposed deep gait retrieval hashing (DGRH) model**

The idea of similarity preservation in a hashing method is that similar codes are generated from semantically similar data. Therefore, mathematically, the goal of the hash function is to learn the image space $X$ to the mapping with $H: X \rightarrow \{ -1, 1 \}^K$ into the Hamming space. Here, $K$ is the k-bit binary hash code generated from the input image $X$.

![Figure 1. Sample gait energy image from the OUISIR-Large Population dataset.](image-url)
Suppose $N$ training GEI images are given, denoted by $\{x_i\}_{i=1}^N$, and each $x_i$ is represented by the dimension feature vector $D$ so that $x_i \in \mathbb{R}^D$. Therefore, $X = [x_1, x_2, x_3, \ldots, x_N] \in \mathbb{R}^{D \times N}$. The label matrix $Y$ in $N$ training GEIs is denoted as $Y = [y_1, y_2, y_3, \ldots, y_N] \in \mathbb{R}^{C \times N}$, and the number of classes is denoted as $C$. If the $i$th GEI belongs to the $j$th class, $y_{ij} = 1$, and if not, $y_{ij} = 0$. Therefore, our proposed hashing method will learn the hash codes $h_i \in \{-1, 1\}^K$ from each input GEI, where $K$ is the length of the binary codes.

As shown in Figure 2, our proposed model takes the gait energy image (GEI) as the input gait data and extracts the features using the deep convolutional neural network (CNN) presented by. The architecture of the CNN is composed of convolutional and pooling layers, a fully connected layer, and a hashing layer; the last layer of the CNN network generates the binary hash code. There are four convolutional layers with different numbers of filters for each layer (16, 32, 64, 128), and the filter size for all layers is $3 \times 3$. After each convolutional layer, there is a max-pooling layer with a stride of 2. The first fully connected layer also takes the number of hidden neurons (1024) as the parameter, and the last fully connected layer serves as the fully connected hash (FCH) layer for hash function learning. The detailed architecture of the CNN proposed by is shown in Table 1. We used the leaky rectified linear unit (LeakyReLU) activation function for all the layers except the hash layer. To learn the hash function, we used a hyperbolic tangent (tanh) activation function to compress the output of the last fully connected layers to the range of $[-1, 1]$. The tanh function is defined as:

$$
\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
$$

(1)

The tanh activation function maps the data into the range of $[-1, 1]$, and the number of neurons ($K$) for the fully connected layer is the desired length of binary hash codes (such as 16 bits, 32 bits, 64 bits, and so on). The learned hash function from the deep convolutional network is $h_i \in \{-1, 1\}^K \times N$ with $N$ GEI images. To calculate the binary hash code ($b_i$) from the output of the hash layer, we utilize the $sign(.)$ function; $b_i = sign(h_i)$. Here, the $sign(.)$ function on a vector and a matrix is expressed as

![Convolutional layers+Pooling layer](image)

**Figure 2.** The architecture for the proposed deep gait retrieval hashing (DGRH) model.

**Table 1.** The convolutional neural network (CNN) architecture for the proposed model.

| Network layer          | Filter size | No of filters | Stride | Size of output volume |
|------------------------|-------------|---------------|--------|-----------------------|
| 1. Input image         |             |               |        |                       |
| 2. Conv1               | $3 \times 3$| 16            | 1      | $240 \times 240 \times 1$ |
| 3. Pool1 (Max)         | $2 \times 2$| 32            | 2      | $238 \times 238 \times 16$ |
| 4. Conv2               | $3 \times 3$| 64            | 1      | $119 \times 119 \times 16$ |
| 5. Pool2 (Max)         | $2 \times 2$| 124           | 2      | $117 \times 117 \times 32$ |
| 6. Conv3               | $3 \times 3$| 1             | 2      | $58 \times 58 \times 32$ |
| 7. Pool3 (Max)         | $2 \times 2$| 1             | 2      | $56 \times 56 \times 64$ |
| 8. Conv4               | $3 \times 3$| 1             | 1      | $28 \times 28 \times 64$ |
| 9. Pool4 (Max)         | $2 \times 2$| 2             | 2      | $26 \times 26 \times 124$ |
| 10. Fully connected layer 1 (tanh) |               |               |        | $13 \times 13 \times 124$ |
|                        |             |               |        | $1 \times 1 \times 1048$ |
|                        |             |               |        | $1 \times 1 \times K$    |
Supervised loss function and optimization

We introduced the loss function to learn the ability of the hash codes. The designed loss functions measure the similarity-preserving ability of the hash codes. Since the well-learned hash codes have a solid classification ability, we expected that the supervised hash function can generate similar hash codes for the gait inputs with the same label. To preserve the similarity of the hash codes, we used the softmax function, which is a multinomial logistic regression function and suitable for predicting the class labels. The formulation for the softmax function is

\[
P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}} \quad \text{where} \quad s = f(x_i; W)
\]

Here, we denote \(W_h\) as the linear weight vectors that connect the output from the last hash layers, so \(W_h \in \mathbb{R}^{k \times N}\). The softmax linear function for the learned representation \(h_i\) can be rewritten as

\[
P(y \in C_j | h_i; W_h) = \frac{e^{w_j^T h_i}}{\sum_{j=1}^{C} e^{w_j^T h_i}}
\]

To minimize the loss across the training sample to further reduce the classification error, we used the cross-entropy loss function. The function of the cross-entropy loss measures the dissimilarity of the predicted label distribution with the true label probability distribution. Therefore, \(\hat{Y}_{ij}\) is the vector of the predicted label that sample \(i\) belongs to class \(j\). The ground truth vector \(Y_{ij}\) is 1 if \(i\) belongs to class \(j\) and 0 otherwise. Therefore, the cross-entropy loss for \(\hat{Y}_{ij}\) and \(Y_{ij}\) is

\[
L(\hat{Y}_{ij}, Y_{ij}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} Y_{ij} \log \hat{Y}_{ij} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} Y_{ij} \log \frac{e^{w_j^T h_i}}{\sum_{k=1}^{C} e^{w_k^T h_i}}.
\]

To reduce overfitting and the variance of the network, we introduced \(L_2\) regularization terms to generalize the training of the deep network. Here, the regularization term is

\[
R_w = \left( \sum_{l=1}^{L} \|W_l\|^2_F \right) \frac{\lambda}{2m}
\]

where \(\lambda\) is the regularization parameter and \(\|A\|^2_F = \sum_i \sum_j (A_{ij})^2\) indicates the Frobenious norm defined as \(\|W_l\|^2_F = W_l^T W_l\). Therefore, the final classification loss for our proposed model is as follows:

\[
L_C = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} Y_{ij} \log \frac{e^{w_j^T h_i}}{\sum_{k=1}^{C} e^{w_k^T h_i}} + \frac{\lambda}{2m} \left( \sum_{l=1}^{L} \|W_l\|^2_F \right)
\]

Quantization loss

The real-value features \((h_i)\) from the hash layer need to be converted into binary codes to perform the retrieval in the Hamming space. To control the learned hash codes’ quality, we introduced the quantization loss. Discrete optimization of the classification loss function \(L_C\) is very challenging because of the binary constraints \(h_i \in \{-1, 1\}^{K \times N}\). To reduce the quantization errors, existing hashing methods apply discrete optimization to the similarity-preserving loss (classification loss) and continuous optimization to the quantization loss. The quantization error \((Q = \|h_i - \text{sgn}(h_i)\|_1)\) is optimized by applying continuous relaxation to the binary constraint. The optimization of the quantization error is very difficult due to the computation intensity and lack of compatibility with the training of the deep network with back-propagation because of the non-differentiable \(\text{sgn}\) function. Therefore, we employed the quantization loss function proposed by, which is suitable to control the quantization error. The quantization loss is defined as follows:

\[
L_Q = \|h_i - 1\|_1
\]
is an elementwise vector, and we applied the smooth surrogate function to the $L_1$ norm.

$$\|\cdot\|_1 \approx \log \cosh(\cdot)$$ to ease the differentiation during back-propagation. The optimized quantization loss can be derived as

$$L_Q = \sum_{i=1}^{N} \log \cosh(|h_i| - 1)$$ (8)

Therefore, by taking equations (7) and (9), we achieve the DGRH optimization loss:

$$\min_{W_k} L = L_C + \beta L_Q$$

$$\min_{W_k} L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} Y_{ij} \log \frac{e^{W_k^T h_i}}{\sum_{k=1}^{L} e^{W_k^T h_i}} + \frac{\lambda}{2m} \left( \sum_{l=1}^{L} \|W_l\|_F^2 \right) + \beta \sum_{i=1}^{N} \log \cosh(|h_i| - 1)$$ (9)

By optimizing the classification loss ($L_C$), we can learn similarity-preserving quality hash codes and control the learned hash codes with the joint optimization function quantization loss ($L_Q$). The k-bit binary codes are achieved using the $\text{sgn}$ function, and the final DGRH optimization loss function is able to reduce the quantization error and increase the retrieval performance. Our proposed DGRH model is trained using adaptive moment estimation (Adam) via back-propagation. In the testing stage, the new query image is converted into hash codes by the trained network. The k-bit binary codes can be obtained using the $\text{sgn}$ function. Once we obtain the k-bit binary codes from the gallery and query sets, we can compute the Hamming distance between them to obtain the result.

**Methods**

**Implementation of proposed method**

The proposed model is developed using the Python programming language. For the data-pre-processing, the Pandas library was used to create and analyse the training and testing gait dataset, and Pickle library was used to import and load the data. The proposed model uses the deep learning approach i.e. the convolutional neural network. To build the convolutional neural network, the deep learning framework known as Keras library with the TensorFlow backend was used. The Keras library allows to construct the neural network easily, and can perform the training and testing of the proposed model. The other important libraries that were used in the development of the model are Numpy and Math libraries for array data manipulation and mathematical formula construction. Finally, the Matplotlib and Seaborn libraries were used for the data visualization of the performance. The source code used for the analysis can be found at Zenodo.25

**Datasets and experimental setting**

We evaluated our proposed deep gait retrieval with public benchmark datasets (CASIA-B, OUISIR-LP, and OUISIR-MVLP). Since we are dealing with the retrieval problem, we considered both short-term and long-term retrieval. For short-term retrieval, we divided the datasets into the same conditions (being in view, wearing clothes, carrying a bag) since gait is captured from the camera in a short amount of time. In real-world gait applications, people are captured at different times and views from different cameras. Therefore, we considered the prepared dataset for both same and mixed conditions to further evaluate our proposed gait retrieval framework.

**CASIA-B dataset**

The CASIA-B gait dataset contains 124 subjects with 11 views and three walking conditions (normal walking, wearing a coat and carrying a bag). We evaluated only the same view ($90^\circ$) with the same walking condition for this setting. For the normal walking condition, four walking sequences (nm1- nm4) are used for the training set, and the remaining sequences (nm5 and nm6) are used for the testing set. There are only two walking sequences for both wearing a coat (cl1, cl2) and carrying the bag (bg1, bg). Therefore, we prepared cl1 as the training set and cl2 as the testing set. Likewise, bg1 was used for training and bg2 was used for testing in the condition of carrying a bag. To evaluate the long-term retrieval problem, we mixed three conditions (NM, CL, BG) and prepared the datasets. The training sets (nm1, nm2, nm3, nm4, cl1, bg1) are included, and for the testing set, we used nm5, nm6, cl2, and bg2.

**OUISIR Large Population (LP) dataset**

To evaluate the proposed framework, we used a subset of the OULP datasets with 1912 subjects and 4 different views ($55^\circ, 65^\circ, 75^\circ, 85^\circ$) following the protocol in. The training and testing dataset is divided equally with 956 subjects
each. Therefore, each subject also has eight GEIs with 4 views and 2 sequences. For the evaluation under the same condition, we prepared the datasets with only the same views (55° − 55°, 65° − 65°, 75° − 75°, and 85° − 85°). Under mixed conditions, each subject with 4 different views is combined for evaluation. Therefore, each subject has eight GEIs (2 walking sequences * 4 views) to perform long-term gait retrieval.

**OUISIR-Multiview Large Population (MVLP) dataset**

OUISIR-MVLP includes 10307 subjects; 5114 subjects are males, and 5193 are females. For the experiments in the OU-MVLP dataset, we followed the protocol setting of [24] which divided the dataset into nearly equal groups with 5153 subjects for the training set and 5154 subjects for the testing set. As we pre-processed the gait sequences into gait energy images with normalized dimensions (128 × 88), we obtained 28 GEIs with 14 different views and 2 walking sequences for each. To evaluate our proposed method, we used only four views (0°, 30°, 60°, and 90°) for both the same-condition and mixed-condition settings since the GEIs with 180° view differences are flipped versions of the images and are considered same-view pairs based on the perspective of the projection. Only the same view was used to perform the short-term gait retrieval. Under mixed conditions, we created datasets with all four views for each subject. Therefore, the number of subjects was still the same, and each subject had eight sequences (2 walking sequences * 4 views).

**Evaluation criteria**

We adopted the Hamming space retrieval approach to evaluate the performance of our proposed model. In hashing, similarity-preserving hash codes are represented instead of data points to increase the speed of retrieval and reduce the storage space. The common methods for searching hash-based codes are the Hash lookup table and the Hamming ranking. The Hamming ranking uses the Hamming distance between the query image and the images from the database. The Hamming distance is computed using a bitwise operation, and the ranked list is generated according to the distance. The returned ranked list is in ascending order with the nearest neighbour in the database with the query. For the hash lookup table approach, lookup tables are constructed with the data points in the database within the Hamming radius (r) of the query. Hash lookup, also known as bucket searching, tries to retrieve all r-neighbours for each query.

The k-bit hash code lengths for the proposed model are 16 bits, 32 bits, 48 bits and 64 bits. The retrieval results are evaluated in three different matrices: precision curves with respect to the Hamming radius, precision curves based on different returned images and the mean average precision (MAP). The precision of r (P@r) is calculated, and the accuracy of the returned images based on the Hamming distance of the query and database images is less than or equal to r (≤r). Here, we set r = 2 for the hash lookup and constructed the precision curves within the Hamming radius (P@r = 2) with respect to different bit lengths. In Hamming ranking, the precision is calculated by the given top returned image. Therefore, we analysed the precision curves with different numbers of top returned images (P@N).

The MAP (mean average precision) is the most important metric to evaluate the hashing algorithms. The ranked list achieved by the Hamming distance between the database and each query is evaluated against the given top returned images. First, we compute the average precision for each query with

$$AP@N = \frac{\sum_{n=1}^{N} P(n) \delta(n)}{\sum_{n'=1}^{N} \delta(n')}$$

(10)

where N represents the top returned images in Hamming ranking and P(n) is the precision of the top-N retrieved results. Then, \(\delta(n)\) is 1 if the n-th retrieved result is in the list and \(\delta(n) = 0\) otherwise. Then, we can calculate the mean AP for all the testing queries to obtain the mean average precision (MAP). The larger the number of MAPs is, the better the quality of retrieval performance.

**Results and discussion**

The mean average precision (MAP) of different datasets with the code length (16,32,48,64) for the same condition are described in Table 2, Table 3 and Table 4. In these tables, the MAP is calculated from the top-100 returned images from the given queries. The precision curves within the Hamming radius (P@r = 2) are also illustrated for different code lengths in Figures 3 to 5. Additionally, based on the top-N returned images, precision curves are also shown for further evaluation of the proposed model.

According to the results, the CASIA-B dataset has the lowest MAP values for the carrying condition (bg-bg) since the motion features of the gait are affected by human walking motion. In addition, normal walking (nm-nm) achieves the highest MAP because of its uncorrupted gait features, and the clothing condition (cl-cl) achieves the second-best result. The MAP for the OUISIR-LP dataset is quite similar for all of the views since the motion features of the different views...
are observable in the side view. The highest MAP values are achieved at (85°) and a 64-bit code length. Compared to the other datasets, the MAP of the OUISIR-MVLP datasets is the lowest overall for the training and testing pairs since this dataset has the largest number of subjects. The highest MAP is in the lateral view pair (90°–90°), and the lowest is in the frontal view pair (0°/C0°). For the bitwise comparison, the hashing performance is better for 48 bits and 64 bits. As the bit length increases, the MAP also increases for the proposed model. Therefore, the precision curves @ top-N are constructed based on the precision values at 64 bits with different returned images (10, 50, 100), as shown in Figure 5.

The evaluation result for the mixed condition is shown in Table 5. The purpose of the mixed condition is to address long-term gait retrieval. The CASIA-B dataset explores different walking conditions, such as carrying bags and wearing...
conditions, in the same view. According to the results, the MAP of the mixed condition in the CASIA-B dataset is able to achieve a higher performance than the same condition. Hamming distance is able to overcome the covariate factors in the retrieval task. To analyse the retrieval performance in view changes, the OUISIR-LP dataset and OUISIR-MVLP datasets are subjected to experiments with four different views. The angular difference for the OUISIR-LP is 10°, and the largest difference is only 30°. Therefore, the MAP for the OUISIR dataset is the highest among other datasets, and the results of the precision curves for a Hamming radius of 2 and top-returned images are also desirable. The MAP for the OUISIR-MVLP dataset is the lowest among the datasets, which is probably due to the large angular difference in the view changes.

The gait is captured from both frontal and lateral views ranging from 0° to 90°. The motion and gait features might not be observed well in the frontal view compared to the lateral view. However, the retrieval performances are quite desirable given the large population dataset with larger view changes. In terms of the bit comparison, the 64-bit scheme also achieves better precision results than the other bit lengths. Therefore, the precision curves for the top-returned images are illustrated based on the precision of 64 bits, and the results are shown in Figure 6. According to the result, the highest precision value based on the different number of bits is from the OUISIR-LP dataset with 4 different views and small

![Figure 4. Comparison of the (a) precision curves at a Hamming radius of 2 (P@r = 2) with different bit lengths and the (b) precision curves of the top-N returned images on the OUISIR-LP dataset.](image)

![Figure 5. Comparison of the (a) precision curves at a Hamming radius of 2 (P@r = 2) with different bit lengths and the (b) precision curves of the top-N returned images on the OUISIR-MVLP dataset.](image)

### Table 5: MAP for different datasets with the mixed condition.

| Dataset | Mean average precision (MAP) |
|---------|-----------------------------|
|         | 16 bits | 32 bits | 48 bits | 64 bits |
| 55-55   | 0.59    | 0.57    | 0.62    | 0.64    |
| 65-65   | 0.64    | 0.63    | 0.68    | 0.72    |
| 75-75   | 0.61    | 0.66    | 0.68    | 0.74    |
| 85-85   | 0.69    | 0.76    | 0.74    | 0.78    |
angular differences. The CASIA-B dataset is also able to achieve the desired result, and the lowest value is for the OUISIR-MVLP dataset. In Figure 6, we compare the precision values based on the number of top-returned images in 64 bits. The OUISIR-LP dataset is also able to achieve a high precision value, while the OUISIR-MVLP dataset has the lowest precision with a large number of angular views.

Comparison with other existing methods
The performance of the DGRH (deep gait retrieval hashing) method is compared against the current gait retrieval works with the same datasets. There are only two papers that address the gait retrieval problem.5,18 Therefore, the first method to be evaluated is the kernel-based semantic hashing method (KSH) proposed by Zhou et al.5 The KSH uses the handcrafted feature learning approach with the Gaussian kernel function to generate the hashing function, and a semantic ranking list is used to retrieve the gait data. Another gait retrieval method that is compared is the deep hashing method presented by Rauf et al.18 who used deep supervised hashing methods with triplet deep learning channels. This method takes the triplet pairs of gait data into the shared triplet channel to calculate the hash function, and the triplet ranking loss is used to retrieve the query from the ranking list.

The proposed framework is compared with the existing methods in Figure 7. The mean average precision (MAP) of the top-100 returned images is used as the evaluation criterion for the retrieval performance.

The mixed condition in different datasets is analysed since gait retrieval is mostly a long-term retrieval problem. According to the results, our proposed method outperforms the other two methods. The retrieval performance of our proposed method is better because of the deep representation of the gait features, and the strength of the CNN can learn better information about the gait motion. The pairwise-based or triplet-based loss in hashing might cause a data imbalance problem because of its complex data preparation for suitable data pairs. In addition, this approach can also suffer from optimization problems. The combination of the classification loss and quantization loss in the proposed method can effectively predict gait labels and control the quality of the hash codes.
Conclusion
This paper proposed the deep gait retrieval hashing (DGRH) model to address gait retrieval. The DGRH uses supervised deep hashing to retrieve the individual gait from the given query. The deep convolutional neural network is used to extract the gait features and generate the hash codes from the last layer of the network. The hash function is learned by optimization of the classification loss and quantization loss, and then gait retrieval is performed in the Hamming Space. The end-end-end hashing model is able to learn discriminative gait features and is efficient in terms of the storage memory and speed. The proposed method is evaluated on three different public datasets and outperforms other methods.

Data availability
Underlying data
CASIB-B Dataset: The dataset is provided by The Institute of Automation, Chinese Academy of Sciences (CASIA) for the research purposes. We used the dataset B from the CASIA Gait Dataset which available on http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp by signing the release agreement.

OU-ISIR LP Dataset: The OU-ISIR Gait Database, Large Population Dataset is provided by The Institute of Scientific and Industrial Research (ISIR), Osaka University (OU). We used that dataset from http://www.am.sanken.osaka-u.ac.jp/BiometricDB/GaitLP.html by signing the release agreement for research purposes.

OU-ISIR MVLP Dataset: The OU-ISIR Gait Database, Multi-View Large Population Dataset (OU-MVLP) is provided by The Institute of Scientific and Industrial Research (ISIR), Osaka University (OU). We used the dataset from http://www.am.sanken.osaka-u.ac.jp/BiometricDB/GaitMVLP.html by signing the release agreement for research purpose.

All the datasets can be obtained by signing the release agreement under research purpose.

Software availability
Source code available from: https://github.com/papamin/Deep-Supervised-Hashing-for-Gait-Retrieval/tree/v1.0.1.

Archived source code at the time of publication: https://doi.org/10.5281/zenodo.5256521.25

License: GPL 3.0.

References
1. Chao H, He Y, Zhang J, et al.: Gaitset: Regarding gait as a set for CROSS-VIEW gait recognition. Proc AAAI Conf Artificial Intelligence. 2019; 33: 8126–8133. Publisher Full Text
2. Rida I, Almaadeed N, Almaadeed S: Robust gait recognition: A comprehensive survey. IET Biometrics. 2018; 8(1): 14–28. Publisher Full Text
3. Wang X, Yan WQ: Human gait recognition based on Frame-by-frame Gait Energy images and Convolutional Long short-term memory. Int J Neural Syst. 2019; 30(1): 1950027. PubMed Abstract | Publisher Full Text
4. Xiao T, Li S, Wang B, et al.: Joint Detection and Identification Feature Learning for Person Search. 2017 IEEE Conf Computer Vision Pattern Recognition (CVPR). 2017. Publisher Full Text
5. Zhou Y, Huang Y, Hu Q, et al.: Kernel-Based Semantic Hashing for Gait Retrieval. IEEE Transactions on Circuits and Systems for Video Technology. 2018; 28(10): 2742–2752.
6. Wang J, Shen HT, Song J, et al.: Hashing for similarity search: A survey. CoRR. 2014.
7. Kulis B, Grauman K: Kernelized locality-sensitive hashing for scalable image search. 2009 IEEE 12th Int Conf Computer Vision. 2009. Publisher Full Text
8. Liu W, Wang J, Kumar S, et al.: Hashing with graphs. In Proc. ICML; 2011.
9. Tang J, Li Z, Wang M, et al.: Neighborhood Discriminant Hashing for Large-Scale Image Retrieval. IEEE Transactions Image Processing. 2015; 24(9): 2827–2840. Publisher Full Text
10. Liu W, Wang J, Ji R, et al.: Supervised hashing with kernels. 2012 IEEE Conf Computer Vision Pattern Recognition. 2012. Publisher Full Text
11. Shen F, Shen C, Liu W, et al.: Supervised Discrete Hashing. 2015 IEEE Conf Computer Vision Pattern Recognition (CVPR). 2015. Publisher Full Text
12. Lin K, Yang H-F, Hsiao JH, et al.: Deep learning of binary hash codes for fast image retrieval. 2015 IEEE Conf Computer Vision Pattern Recognition Workshops (CVPRW). 2015. Publisher Full Text
13. Xia R, Pan Y, Lai H, et al.: Supervised hashing for image retrieval via image representation learning. Proc AAAI Conf Artificial Intelligence. 2014; pages. 2156–2162.
14. Lai H, Pan Y, Liu Y, et al.: Simultaneous feature learning and hash coding with deep neural networks. 2015 IEEE Conf Computer Vision Pattern Recognition (CVPR). 2015. Publisher Full Text
15. Zhang R, Lin L, Zhang R, et al.: Bit-Scalable Deep Hashing With Regularized Similarity Learning for Image Retrieval and Person Re-Identification. IEEE Transactions on Image Processing. 2015; 24(12): 4766–4779. Publisher Full Text
16. Zhu H, Long M, Wang J, et al.: Deep hashing network for efficient similarity retrieval. In Thirtieth AAAI Conference on Artificial Intelligence. 2016.
17. Cao Y, Long M, Liu B, et al.: Deep Cauchy Hashing for Hamming Space Retrieval. 2018 IEEE/CVF Conference Computer Vision Pattern Recognition. 2018. Publisher Full Text

18. Rauf M, Huang Y, Wang L: Gait Retrieval: A Deep Hashing Method for People Retrieval in Video. Communications in Computer and Information Science Pattern Recognition. 2016: 383–391. Publisher Full Text

19. Han J, Bhanu B: Individual recognition using gait energy image. IEEE Transactions Pattern Analysis Machine Intelligence. 2006; 28(2): 316–322. Publisher Full Text

20. Min PP, Sayeed S, Ong TS: Gait Recognition Using Deep Convolutional Features. 2019 7th Int Conf Information Communication Technology (ICoICT). 2019. Publisher Full Text

21. Gong Y, Lazebnik S: Iterative quantization: A procrustean approach to learning binary codes. Cvpr. 2011; 2011. Publisher Full Text

22. Yu S, Tan D, Tan T: A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition. 18th Int Conf Pattern Recognition (ICPR’06). 2006. Publisher Full Text

23. Muramatsu D, Makihara Y, Yagi Y: Cross-view gait recognition by fusion of multiple transformation consistency measures. IET Biometrics. June 2015; 4: 62–73(11). Publisher Full Text

24. Takemura N, Makihara Y, Muramatsu D, et al.: Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition. IPSJ transactions on Computer Vision and Applications. 2018; vol. 10, no. 1.

25. Sayeed S, Min PP, Ong TS: Deep Supervised Hashing for Gait Retrieval (v1.0.1). Zenodo. 2021. Publisher Full Text
The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com