Mélange fabric image retrieval based on soft similarity learning

Jun Xiang, Ruru Pan and Weidong Gao

Abstract
Fabric image retrieval, a special case in Content Based Image Retrieval, has high potential application value in many fields. Compared with common image retrieval, fabric image retrieval has high requirements for results. To address the actual needs of the industry for Mélange fabric retrieval, we propose a novel framework for efficient and accurate fabric retrieval. We first introduce a quantified similarity definition, soft similarity, to measure the fine-grained pairwise similarity and design a CNN for fabric image representation. An objective function, which consists of three losses: soft similarity loss for preserving the similarity, cross-entropy loss for image representation, and quantization loss for controlling the quality of hash code, is used to drive the learning of the model. Experimental results demonstrate that the proposed method can not only achieve effective feature learning and hashing learning, but also effectively work on smaller datasets. Comparative experiments illustrate that the proposed method outperforms the compared methods.

Keywords
Fabric representation, fabric retrieval, image retrieval, similarity learning

Introduction
Mélange fabric, also known as color-spun fabric, is a fabric directly woven from Mélange yarns, in which Mélange yarns are made by fully mixing two or more fibers of different colors to have a unique color mixing effect. However, its proofing process is extremely complicated. When the textile company receives a fabric sample from the customer, the company first needs to repeat the proofing and comparison, which is a very time-consuming and laborious task. In this study, we propose to use image retrieval technology to reduce the times of proofing. The specific steps are: (1) First establish the database of images of historical production products; (2) When receiving samples from customers, use the images of the samples to retrieval similar historical production products; (3) Analyze the production process of similar products to guide the proofing process. This is the main motivation for this research. Besides, with the improvement of people’s living standards, consumer demands for goods is no longer limited to its practical performance, but tends to be beautiful and diversified. Therefore, the “small-batch and multi-variety” has increasingly become a new production mode for the textile industry. Under this mode of production, companies have accumulated a large amount of historical production data, which make the search task more difficult. Image retrieval plays a very important role in product search work, the key to this idea is how to establish an accurate retrieval system. The Keywords Based Image Retrieval (KBIR) system is often used for product management in the textile

Original Article

Jiangnan University, Wuxi, Jiangsu, China

Corresponding author:
Weidong Gao, Jiangnan University, No. 1800, Lihu Avenue, Binhu District, Wuxi, Jiangsu 214122, China.

Email: gaowd3@163.com
industry. However, the KBIR system used in the textile industry builds indexes for fabric images through manually annotated keywords and users search for relevant fabrics through the annotated keywords. Although this system has the characteristics of high retrieval speed and easy construction, the search method can provide limited functions. Moreover, manual labeling of fabric images is also highly subjective, leading to instability of retrieval results, and further leading to the inefficiency and inaccuracy of KBIR. As the characteristics of Mélange fabrics are difficult to describe, the use of KBIR technology cannot solve our problem.

On the contrary, the Content-Based Image Retrieval (CBIR) system is more objective. Many researchers in this field have also shifted their research focus to CBIR. Compared with KBIR, CBIR uses image content to index images, which can avoid the influence of human subjectivity on the result. Image retrieval methods have gone through significant development in the last decade, starting with descriptors based on hand-crafted features, first organized in Bag-of-Words (BoWs),17 and further expanded by spatial verification,18 hamming embedding,19 and query expansion.20 Generally, a CBIR system is expected to output the list of images similar to the query. Technically speaking, the CBIR system consists of two key issues, namely image representation, and similarity measurement. The first phase converts the image into a vector and the second phase measures the similarity between the query and the images from the dataset. The traditional methods usually employ hand-crafted descriptors, such as SIFT,21 GIST,22 etc., to represent the feature of the image. Although having achieved certain success, these methods depend heavily on feature engineering, which leads to their limitation.

Recently, a significant breakthrough has been achieved on image analysis by moving from the early hand-crafted feature based methods to deep learning based end-to-end framework. Even though, the most challenging issue is still associating the pixel-level feature to high-level semantic feature from human perception, which is called “semantic gap.” Nowadays, the Convolutional Neural Network (CNN) based methods have been proved to be the most effective. Krizhevsky et al.23 first used a CNN model driven by classification task to extract image features for retrieval and achieved good performance on ImageNet. Recently, many methods of jointly learning image representations and similarity metrics or distances have been proposed, which can be seamlessly used for retrieval. Specifically, these methods use metric learning embedding to train CNN. There are two simple but effective methods for metric learning: pair embedding and triplet embedding. Both methods learn the similarity by bringing samples with same labels closer to each other, and pulling samples with different labels apart. These discriminative model can learning semantically meaningful metrics while learning the image representation. Therefore, these methods are more robust inter-class confusion and intra-class changes.

Fabric image retrieval, as a special case of generic image retrieval, is a meaningful issue, due to its potential values in many areas, such as textile product design, e-commerce, and inventory management. However, due to the particularity of the fabric itself, the commonly used retrieval methods are difficult to directly apply to fabric retrieval. The main reason is that textile fabric images do not contain 3-D shape features, but only contain texture and color features. As shown in Figure 1, we present a natural image and two Mélange fabric images. In Figure 1(a), we can clearly observe a black swan with obvious outline and shape. While Figure 1(b) and (c) present two fabric samples with no obvious 3D shape or contour information. Comparing Figure 1(d) to (f), we can find a huge difference in periodicity and texture complexity between the fabric and natural images. Because of the fundamental differences, retrieval methods suitable for general images may perform poorly for fabric images.

The main features in fabric images are color and texture, which are all global features. The goal of fabric image retrieval is to find fabrics with similar textures, colors, and materials. However, the pairwise and triple embedding methods crudely define images with completely similar labels as positive samples, and all others as negative samples, which seems to be unfair to those samples with partially similar labels. Furthermore, these approaches ignore the relationship between labels, and some information will be lost during model training.

In this paper, we propose a novel embedding method, which can easily be unified into a CNN for joint optimization. In particular, the proposed method aims to learn the similarity from fabric images with multi-dimensional labels. The contributions of this work mainly include:

1. For fabric images rich in color and texture features, we propose to use the output of shallow layers of VGG to represent fabric image content;
2. A more fine-grained similarity is defined to measure the similarity between fabric images with multi-label. To embed fine-grained similarity in the output features, we propose to use covariance to facilitate the model to learn the similarity matrix between the mini-batch input;
3. We formulate an end-to-end deep hashing model which can generate high-quality hash codes for fabric image retrieval.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 introduces the proposed framework in detail. The results and discussion are presented in Section 4. And Section 5 concludes the whole works.
Related works

According to different technical components, there are two topics most relevant to this research, namely fabric image retrieval and similarity learning. The two topics will be reviewed in this section.

Fabric image retrieval

Fabric image retrieval pays more attention to color and texture features, while more shape and contour features in common retrieval algorithms. According to the different feature extraction mechanisms, the previous works of fabric image retrieval can be divided into two categories: hand-crafted based methods and learning based methods.

Hand-crafted based methods commonly adopt pixel-level descriptors, such as image color histogram (CM), Scale-Invariant Feature Transform (SIFT) keypoint descriptor, Histogram of Oriented Gradient (HoG) descriptor, Color Moment (CM), and Local Binary Pattern (LBP) to fabric images. However, most traditional methods use a combination of two or more feature descriptors to represent fabric images. Jing et al. proposed two methods for fabric image retrieval, both of which combine color and texture features to describe the image. Recently, Zhang et al. tried to combine LBP and Fourier Transform to extract features of fabric images for retrieval. In follow-up research, they also used to describe the fabric completely using color feature descriptors, namely dominant color and color moment. Tadi Bani and Fekri-Ershad proposed a image retrieval method based on a combination of texture and color feature. Although achieving certain success, these methods are limited to small datasets. Moreover, the small jitters in scale or details have a great impact on the retrieval results, which demonstrates the less robustness of these methods.

Learning-based methods have been demonstrated to have stronger generalization performance when labeled data is available. Some previous works have used classification tasks to driven CNN models to learn fabric image representation. Xiang et al. proposed a two-strategy with the idea that deep CNN can learn binary codes and features from the labeled data. And Xiang et al. presented a multi-task learning model to learn fabric image representation from four views: coarse texture, fine texture, fabric style, and pattern forming method respectively. In addition, Cai et al. used a Triplet-CNN to learning fabric image representation under similarity metrics in a supervised manner. Deng et al. proposed an embedding method called focus ranking for fabric image retrieval. Although the previous works can achieve good performance on relatively large datasets. However, when there is
not enough labeled data to train the CNN model, these methods may not work well.

**Similarity learning**

Similarity learning, also called distance metric learning, plays an important role in many visual tasks, such as face recognition and image retrieval. Xing et al. proposed a novel similarity learning method based on a discriminative deep CNN for face verification in the wild. Bell and Bala proposed a contrast embedding method to learn the depth metric of visual search in interior design. Chen et al. proposed a novel similarity learning approach, which is combined with space constraint, for person re-identification. Pinheiro proposed a neural network model for unsupervised domain adaption using a similarity-based classifier. Kordopatis-Zilos et al. presented a network that learns to compute the similarity between pairs of videos. Recently, deep hashing methods have shown great retrieval accuracy and efficiency in image retrieval. The key of hashing methods is to preserve similarity into the generated hash codes. Wu et al. presented a novel unsupervised deep video hashing for fast large-scale similarity search, which integrates feature learning and hash function learning in a self-taught manner. To reduce the information loss in hash code generation, many researchers introduce quantization loss. Miao et al. proposed a bottom-up learning strategy, termed Adversarial Constraint Module, where low-coupling binary codes are introduced to guide the learning of binary local descriptors.

As mentioned in Section 1, there are two embedding methods related to our study, namely pair and triplet embedding. The two methods push the samples with the same label closer to each other and pull the samples with different labels apart from each other. However, their optimization and training are quite different. The pair embedding constructs its energy function on a training set composed of image pairs and optimizes it to assign small distances to similar pairs and large distances to different pairs. While triplet embedding mainly uses the Triplet-Loss function for training. However, these two methods cannot solve the challenging problem of fine-grained fabric image retrieval when the number of fabric images is limited. Instead of simply dividing the sample into positive and negative samples, in this paper, we make full use of the multi-dimensional labels of Mélange fabric to define different degrees of similarity for learning the representation of fabric images.

**Soft similarity learning**

**Problem definition**

Given a training set of $N$ images $I = \{I_1, I_2, \ldots, I_N\}$ and multi-dimensional labels $Y = \{Y_1, Y_2, \ldots, Y_N\}$, the goal of soft similarity learning is to learn a mapping function $f : I \mapsto \{-1, +1\}^q$, so that an input image $I_i$ can be encoded into a $q$-bit binary code $f(I_i)$, with the similarities of images being preserved. For the two images, $I_i$ and $I_j$, with multi-dimensional labels, the similarity is commonly defined as $s_{ij} = 1$ if they completely share the same labels, and $s_{ij} = 0$ if they have at least one label different. The similarity is also called hard similarity. As mentioned in the introduction, this definition does not consider multi-label information, nor can it sort images with multiple class labels for similarity. In our design, the pairwise similarity is defined as follows: (1) Calculate the similarity matrix $M_q = \{m_{ij}\}_{i,j=1}^D$ of the multi-labels, $Y_i$ and $Y_j$, corresponding to the pair of images, $I_i$ and $I_j$, where $D$ is the dimension of the label, and $m_{ij} = 0$ if the $d$th dimension of $Y_i$ and $Y_j$ are the same, otherwise $m_{ij} = 1$; (2) Then the similarity of the two images can be computed by equation (1).

$$s_{ij} = \sum_{d=1}^D m_{ij}$$

As shown in Figure 2, the similarity matrix $M$ has six different situations when $D=3$. According to equation (1), the similarity of pairwise images can be passed into four levels: completely similar ($5=1$), very similar, a little
similar, and dissimilar ($S=0$). For Approximate Neighbor Nearest (ANN) search, the generated hash codes should preserve the similarity of the pairwise fabric images. To be specific, for a pair of generated binary codes $b_i$ and $b_j$, if $S_{ij}=0$, which means the pairwise images do not share any label, the Hamming distance should be large; if $S_{ij}=1$, which means the pairwise images completely share the same labels, we expect the Hamming distance should be small; otherwise, the binary codes $b_i$ and $b_j$ should have a suitable Hamming distance complying with the soft definition of similarity $S_{ij}$.

We present the pipeline of the proposed framework for similarity learning in Figure 3. The proposed model learns the similarity of fabric images in a supervised manner. The model receives a mini-batch of fabric images and then processes them through deep CNN. It consists of several convolution/pooling layers to abstract the feature of fabric images, the fully connected layer to approximate optimal dimension-reduced presentation and the hash layer to generate hash codes. In this study, the soft similarity loss is introduced to the objective function for similarity learning, the quantization loss is used to adjust the quality of generated hash codes, and the Cross-Entropy loss is used to improve the representation learning. Next, the main components of the framework are introduced.

**Network architecture**

It has been proved that low-level features, that is, colors and textures, always appear in natural images and can be extracted by the first several layers of a convolutional neural network which is trained on natural images (ImageNet). As shown in Figure 1, the main features in fabrics are texture and color. So, in this paper, we simply adopt VGG-16 model as the stem network for soft similarity learning. The proposed scheme does not use the entire VGG-16 model, and just uses the first three convolution/pooling layers, whose parameters are initialized with the weights CNN pre-trained on the ImageNet. Such a choice can avoid overfitting on a relatively small dataset. Note that the output of the last pooling layer is flattened after *Local Response Normalization* (LBN) and then input to the fully connected layer, instead of being max-pooled. We connect to the Dropout layer (with a probability of 0.6) to eliminate the redundant information. There are five fully connected layers in the proposed network, where the number of nodes in each layer is $(28 \times 28 \times 256, 4096, 1024, q)$. $q$ is the length of output hash codes. The proposed method can be easily extended to other deep networks (entire or part), such as ResNet50, AlexNet, and GoogLeNet. The detailed discussion will be presented in
the Experiments (Section 5). In order to realize hash coding, we use the \textit{tanh} activation to map the output of the number of relevant imagesthe last fully connected layer to be within \([-1, +1]\).

**Objective function**

The goal of this study is to train a non-linear mapping function \(f\) that encodes each fabric image into \(q\)-bit hash codes with the similarity been preserved. In particular, the output codes of the similar images should be close in Hamming space, while the output codes of dissimilar images should be far apart in Hamming space. For this purpose, we design the objective function of the proposed model, which consists of three parts.

\[
L = \lambda_s L_s + \lambda_e L_e + \lambda_q L_q
\]  
(2)

where \(L\) is the total Loss, \(L_s\) is the similarity loss defined in this study for preserving the similarity, \(L_e\) is the cross-entropy loss for image representation, and \(L_q\) is quantization loss for controlling the quality of hash code. There are three weighting parameters: \(\lambda_s\), \(\lambda_e\), and \(\lambda_q\), which are used to balance the effects of different objective functions.

For a mini-batch \(X\) of input \(n\) fabric images, we first compute the similarity matrix \(M\) using equations (1) and (3).

\[
\begin{pmatrix}
I_{11} & \ldots & I_{1D} \\
\vdots & \ddots & \vdots \\
I_{n1} & \ldots & I_{nD}
\end{pmatrix}
\rightarrow
\begin{pmatrix}
S_{11} & \ldots & S_{1n} \\
\vdots & \ddots & \vdots \\
S_{n1} & \ldots & S_{nn}
\end{pmatrix}
\]  
(3)

The two subscripts of \(I\) respectively represent the index of the corresponding image in the mini-batch and the index of the label dimension. \(S\) denotes the soft similarity of the pairwise images, and its two subscripts are the index of the corresponding image in the mini-batch. The proposed model receives the mini-batch and then outputs a matrix \(O\) of \(n \times q\). The similarity matrix of the output can be computed by:

\[
\begin{pmatrix}
X_1 \\
\vdots \\
X_n
\end{pmatrix}
\rightarrow
\begin{pmatrix}
O_1 \\
\vdots \\
O_n
\end{pmatrix}
\rightarrow
\begin{pmatrix}
E_{11} & \ldots & E_{1n} \\
\vdots & \ddots & \vdots \\
E_{n1} & \ldots & E_{nn}
\end{pmatrix}
\]  
(4)

where \(E_{ij}\) represents the Euclidean distance between \(O_i\) and \(O_j\), which can be calculated by the following equation:

\[
E_{ij} = \sqrt{\sum_{c=1}^{q}(O_{ic} - O_{jc})^2}
\]  
(5)

To preserve the similarity, we propose to use covariance to facilitate the model to learn the similarity matrix of the input. In our design, the soft similarity loss \(L_s\) can be expressed as:

\[
L_s = \text{Cov}(S, E)
\]  
(6)

For efficient nearest neighbor search, the similarity of fabric images should be preserved in the Hamming space. The model receives a mini-batch with \(n\) fabric images and converts them into \(n\) \(q\)-bits hash-like codes. The model only generates approximate hash codes that have the values within \([-1, +1]\). To finally get the hash codes, we need to convert the hash-like codes into binary codes by using \textit{sgn} function, as shown in equation (7).

\[
B_k = \text{sgn}(O_k) = \begin{cases} +1 & \text{if } O_k \geq 0 \\ -1 & \text{otherwise.} \end{cases}
\]  
(7)

The \textit{sgn} function is not continuous and is not included in the network to propagate back. Part of the information will be lost in the hashing process. Here we reduce the information loss by adding quantization loss to improve the quality of the hash codes. The quantization loss \(L_q\) can be written as:

\[
L_q = \sum_{i=1}^{n} \sum_{j=1}^{q} (|O_{ij}|-1)^2
\]  
(8)

where \(i\) is the index of the corresponding image in the mini-batch, and \(j\) is the index of this element in the \(q\)-bit output. We use the \textit{tanh} activation to map the output of the last fully connected layer to be within \([-1, +1]\). Quantitative loss is used to encourage numbers less than 0 to be closer to \(-1\), and numbers greater than 0 to be closer to +1. This configuration can fundamentally reduce the loss of information in the hashing process. In addition, the cross-entropy loss is used to enhance the image representation. For specific implementation methods, refer to citation.16

The standard back-propagation algorithm with the mini-batch gradient descent method is used to optimize the pairwise loss function during the training process. The objective function \(L\) is differentiable, so we just use \textit{Stochastic Gradient Descent} (SGD) to optimize the parameters of the model in a standard manner. As shown in equation (2), the proposed method involves three loss function, in this paper, we adopt joint training to optimize the parameters of the model. Note that, after the learning procedure, we have not obtained the corresponding binary codes of input images yet. The binary codes are generated by using equation (7).

In this way, we can train the proposed deep CNN in an end-to-end manner, and then any new input images can be encoded into binary hash codes by the trained model. By ranking the distance of these binary hash codes in the Hamming space, we can obtain effective retrieval.

**Experimental configuration**

**Dataset**

For evaluating the performance of the proposed method, we build a fabric image dataset named MFT-fabric-v1,
In this work, we evaluate the performance of retrieval methods by using widely-used metrics: precision-recall curve and mAP (Mean Average Precision) value. The mAP is a typical evaluation method for CBIR, which is computed by sorting images in descending order of relevance per query and averaging AP of individual queries. The average precision (AP) can be calculated by:

\[
AP(q) = \frac{1}{N_q(q)@n} \sum_{i=1}^{N_q(q)@n} \left( Tr(q,i) - \frac{N_r(q)@i}{i} \right)
\]

where \( Tr(q,i) \in \{0,1\} \) is an indicator function that if the query \( I_q \) and the \( i \)th retrieval result \( I_i \) have the same labels, \( Tr(q,i)=1 \); otherwise, \( Tr(q,i)=0 \). \( N_r(q)@i \) denotes the number of relevant images within the top \( i \) images. Let \( Q \) represents the numbers of query sets. Then the mAP can be computed by:

\[
mAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)
\]

For precision and recall, we first introduce the definition of \( TP, FN, FP, \) and \( TN \) which are clearly shown in Table 2. Then the precision and the recall of retrieval results are defined as:

\[
\text{precision} = \frac{TP}{TP+FP} \times 100\% \\
\text{recall} = \frac{TP}{TP+FN} \times 100\%
\]

In evaluation, two images are considered to be semantically similar if they have the same annotations. For both evaluation metrics used in this work, a larger value indicates better retrieval performance. Generally, the precision will decrease as the recall increases.

**Table 1. Type and amount of fabric by raw materials.**

| Composition | Amount |
|-------------|--------|
| 65T35R      | 4906   |
| 100T        | 4664   |
| 65P35C      | 5091   |
| 60C40T      | 4380   |
| 100R        | 4036   |
| 100CC       | 4179   |
| 100SC       | 3793   |
| 60CC40T     | 5552   |
| 65P35C      | 5636   |
| 100C        | 4631   |

which consists of 46,868 Mélange fabric images as the training-set, and 3672 Mélange fabric images as testing-set. All images in the dataset are annotated from three views: color, raw materials, and texture. The fabrics in the dataset are weft-knitted. The raw materials for woven these fabrics include polyester (T), rayon (R), and cotton (C). There are three types of cotton: common cotton (C), semi-combed cotton (SC), and combed cotton (CC). The Mélange fabrics in the dataset main are formed from 10 different material combinations (one or two raw materials), and the amount of each is shown in Table 1. The color of Mélange fabric is defined as the color of the special colored yarn or fiber (other than the basic color, such as white and black). Figure 4 shows three kinds of blue Mélange fabrics, which have very different percentages of blue fibers. This definition may cause the general handcrafted based methods to be difficult to represent the color of Mélange fabric. So this paper proposes to use the deep learning based method to describe the feature of Mélange fabrics. According to this definition, there are a total of nine colors of Mélange fabric in the dataset, namely gray, red, orange, yellow, brown, green, blue, purple, and colorful gray. With respect to texture, we divide the weft-knitted fabric into six levels based on the thickness and weight of the yarn used. The testing-set is used to evaluate the performance of the image retrieval methods and contains a total of 72 sets (different sets of images belong to different categories), each of which consists of a query and 50 related images corresponding to it. To evaluate the robustness of the retrieval methods, the queries are augmented with some transformations: rotation, flipping, and scaling. To avoid the influence of capture conditions, the images are collected in a stable environment: DigiEye stable light-box, which is equipped with a Nikon D7000 camera, a special pick-up head, and a standard illumination D65. Also, the resolution of the collected images is 96 dpi.

**Evaluation metrics**

In this work, we evaluate the performance of retrieval methods by using widely-used metrics: precision-recall curve and mAP (Mean Average Precision) value. The mAP is a typical evaluation method for CBIR, which is computed by sorting images in descending order of relevance per query and averaging AP of individual queries. The average precision (AP) can be calculated by:

\[
AP(q) = \frac{1}{N_r(q)@n} \sum_{i=1}^{N_r(q)@n} \left( Tr(q,i) - \frac{N_r(q)@i}{i} \right)
\]

where \( Tr(q,i) \in \{0,1\} \) is an indicator function that if the query \( I_q \) and the \( i \)th retrieval result \( I_i \) have the same labels, \( Tr(q,i)=1 \); otherwise, \( Tr(q,i)=0 \). \( N_r(q)@i \) denotes the number of relevant images within the top \( i \) images. Let \( Q \) represents the numbers of query sets. Then the mAP can be computed by:

\[
mAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)
\]

For precision and recall, we first introduce the definition of \( TP, FN, FP, \) and \( TN \) which are clearly shown in Table 2. Then the precision and the recall of retrieval results are defined as:

\[
\text{precision} = \frac{TP}{TP+FP} \times 100\% \\
\text{recall} = \frac{TP}{TP+FN} \times 100\%
\]

In evaluation, two images are considered to be semantically similar if they have the same annotations. For both evaluation metrics used in this work, a larger value indicates better retrieval performance. Generally, the precision will decrease as the recall increases.

**Implementation details and compared methods**

In the proposed learning framework, there are three parameters: \( \lambda_1, \lambda_2, \lambda_3 \), for balancing the effect of different loss. In the following section, we will discuss these three parameters through ablation experiments. The previous layers of VGG are pre-trained on ImageNet to make the parameters of the model have a good initialization state. During the training, the hyper-parameter configuration is as follows: batch_size = 32, weight_decay = 5e-5, optimizer = ADAM, and initial learning_rate = 1e-3. Specifically, the learning rate of the parameters of the convolutional layers inherited from VGG-16 is set to one-tenth of the subsequent layers. And the strategy for adjusting the learning rate can be denoted by:

\[
lr_n = \begin{cases} 
    lr_{n-1}, & \text{if } n \leq a \\
    lr_{n-1} \times \left(1 - \frac{n-a}{N-a}\right), & \text{if } n > a 
\end{cases}
\]

During the training, the hyper-parameter configuration is as follows: batch_size = 32, weight_decay = 5e-5, optimizer = ADAM, and initial learning_rate = 1e-3. Specifically, the learning rate of the parameters of the convolutional layers inherited from VGG-16 is set to one-tenth of the subsequent layers. And the strategy for adjusting the learning rate can be denoted by:

\[
lr_n = \begin{cases} 
    lr_{n-1}, & \text{if } n \leq a \\
    lr_{n-1} \times \left(1 - \frac{n-a}{N-a}\right), & \text{if } n > a 
\end{cases}
\]
where $lr_{n}$ is the learning rate of the $n$th epoch and $N$ is the total number of epochs configured. $a$ denotes the starting epoch where the learning rate begins to decay. $p$ is a parameter used to control the intensity of each decay within $(0, 1)$. In this work, we set $a=20$, $N=60$, $p=0.7$, respectively. In this paper, we implement the proposed model by using the Pytorch toolkit. The hardware environment is as follows: CPU=E5 2623V4@2.60GHz, RAM=DDR4 32G, GPU=GeForce RTX 3090(24G) × 2.

Results and discussion

In this section, we do several experiments to demonstrate the superiority of the proposed method, including parameters analysis, ablation experiments, comparison with state-of-the-art methods.

Parameters analysis

The parameters $\lambda_s$, $\lambda_c$, and $\lambda_q$ are used to control the effect of the different objective functions. The role of $L_q$ is to control the quality of the hashing process, we only need to give it a small weight. Therefore, in this study, we fix $\lambda_s=1$ and $\lambda_q=0.1$, and then discuss the effect of $\lambda_c$ on the retrieval results.

Table 3 shows the results of the proposed method by using different values of $\lambda_c$. It is obvious that the model achieves the best results at $\lambda_c=0.1$. It is because that a larger $\lambda_c$ will result in more discrete but less similarity-preserved hash codes and a smaller $\lambda_c$ will make the cross-entropy loss less effective. Note that when $\lambda_c$ goes from 0 to 0.01, the performance of the model is greatly improved, which demonstrates the necessity of introducing Cross-Entropy. The results also show that the longer the hash code length, the better the performance achieved, but the excessively long code length will cause feature redundancy and reduce the performance. Overall, too larger or too small a $\lambda_c$ value will destroy the balance between the cross-entropy loss and the similarity loss.

Ablation experiments

In the proposed model, we introduce three objective functions. The results of the third and fourth rows in Table 3 have proved that Cross-Entropy has greatly improved the performance of the proposed model. Figure 5 shows the effect of the quantization loss on the fabric image dataset. Notice that the larger the area under the pr curve, the better the retrieval performance. As we can see, there is a significant performance improvement when we add quantization loss to the objective function of the model. The proposed method, the $sgn$ function (as shown in equation (7)) is employed to generate the binary hash codes. Generally, there is a great loss of information in this hashing process. The quantization loss is used to reduce the loss of information, and thus improve the quality of hashing process. The results in Table 3 and Figure 5 together illustrate the effectiveness and necessity of the three proposed objective functions.

Comparison with different settings

To justify the rationality of the proposed configuration, we conduct some comparison experiments. Specifically, we compare the following different network configurations (as the stem network): entire VGG-16, ResNet50, AlexNet, and GoogLeNet.

Figure 4. Three kinds of blue Mélange fabric.
Table 3. Results of mean average precision (mAP) for the different parameter values of $\lambda_c$.

| $\lambda_c$ | mAP  |
|-------------|------|
|             | 32-bit | 64-bit | 128-bit | 256-bit |
| 0           | 0.754  | 0.761  | 0.768   | 0.771   |
| 0.01        | 0.816  | 0.829  | 0.866   | 0.869   |
| 0.1         | 0.877  | 0.893  | 0.915   | 0.913   |
| 1.0         | 0.866  | 0.872  | 0.901   | 0.892   |
| 10.0        | 0.845  | 0.851  | 0.870   | 0.878   |

Figure 5. The precision-recall curves of retrieval performance with and without quantization loss.

Table 4. Results of different network configurations on the fabric dataset.

| Methods      | mAP  |
|--------------|------|
|              | 32-bit | 64-bit | 128-bit | 256-bit |
| ResNet50     | 0.823  | 0.849  | 0.873   | 0.877   |
| AlexNet      | 0.791  | 0.824  | 0.838   | 0.841   |
| GoogLeNet    | 0.769  | 0.775  | 0.792   | 0.804   |
| Entire VGG-16| 0.812  | 0.835  | 0.868   | 0.876   |
| Proposed     | 0.877  | 0.893  | 0.915   | 0.913   |

Table 4 presents the results of mAP metric of the proposed method and its modifications with different code-length, respectively. For fair comparison, all compared methods are trained under the supervision of our proposed objective function with the same weight parameter settings: $\lambda_c = 1, \lambda_r = 0.1, \lambda_q = 0.1$. The experimental results show that the entire VGG-16 obtain a better result than AlexNet, GoogLeNet, but slightly worse than ResNet50, which demonstrates the superior performance of ResNet50. However, the fabric images mainly contain color and texture features, which proved to appear in the first few layers of the CNN. So, the proposed method uses the first three convolution/pooling layers for fabric image representation. The results show that the proposed method achieves better performance than the entire VGG-16 and ResNet-50, which demonstrate the effectiveness of this configuration.

Efficiency analysis

The efficiency issue is addressed in this part because it prevents such methods from being widely deployed in more retrieval applications. As mentioned in the related work, FRHS and FRMT are specifically designed for fabric image retrieval in those baselines (both of them are deep hashing methods). Consequently, the training and query expenses on MFT-fabric-v1 dataset at various code lengths when using those methods are summarized in Table 5. Here, we compare two metrics: network training time (NT) and query time (QT). The experiments are conducted on the same hardware configuration reported previously. As can be seen, the proposed hashing method takes less time, both in training time and query time. In particular, the training time of the proposed model is much less than the others, indicating that the objective function and learning method proposed in this paper have high efficiency. Moreover, it is worth noting that the query time of proposed method only costs 0.25 s (128 bits), which can expand the application scope of fabric image retrieval, such as mobile applications.

Comparison with SOTA methods

In this section, we compare the retrieval performance of the proposed method with three supervised methods, three unsupervised methods, and four methods dedicated to fabric image retrieval. To make a fair comparison, we employ VGG-16 as the stem of all deep learning based methods. With respect to the implementation of three supervised methods, including CNNH,\textsuperscript{52} DPSH,\textsuperscript{53} and CSQ,\textsuperscript{54} and three unsupervised methods, including UHBDNN,\textsuperscript{55} SSDH,\textsuperscript{56} and SADH,\textsuperscript{57} we simply follow the author’s public code and implement it with PyTorch. The four methods, which are dedicated to fabric image retrieval, are FRHS,\textsuperscript{14} FRMT,\textsuperscript{16} CMGF,\textsuperscript{10} and MRI-LBP,\textsuperscript{12} in which FRHS and FRMT are our previous works.

Table 6 presents the mAP comparison results on MFT-fabric-v1 dataset. It can be observed that the retrieval method proposed in this paper achieves the best performance under different code lengths. For example, when we set the length of the code to 128 bits, the proposed
method achieves the mAP value of 0.915, which surpasses other comparison methods. The comparison results also reflect that, from 32 bits to 128 bits, as the length of the hash codes increase, the retrieval performance has improved. To a certain extent, this phenomenon indicates that the longer hash codes can capture more discriminative information from the image in deep hashing models. However, an excessively long code length may result in feature redundancy, resulting in performance degradation. So, in this study, we set the length of hash codes $q$ to 128. Moreover, we clearly find that the supervised methods perform better than the unsupervised methods on MFT-fabric-v1 dataset. The two hand-crafted descriptor based methods cannot achieve good performance due to the limitation of feature engineering. In Figure 6, we also present the precision-recall curves of the compared methods on the testing-set. The area under the curve corresponding to the proposed method is larger than the curve corresponding to other methods, which indicates the retrieval performance of our method is better than other methods.

Table 7 presents the results of the compared methods at 128 bits on MFT-fabric-v1 dataset using a different number of training images. The results demonstrate that the proposed model can be trained on a smaller data set and achieve good results. Four retrieval samples using our method are shown in Figure 7, in which retrieval results and queries are very similar in color and texture. Moreover, the retrieved results have good visual continuity, precisely because we use soft similarity to drive model learning.

**Conclusion**

In this paper, we present an efficient fabric retrieval framework based on our previous work. To drive the model to learn fabric image representation, we introduce a quantified similarity definition, soft similarity, to measure the fine-grained pairwise similarity. In view of the characteristics of
the fabric, we use the first several layers of the pre-trained VGG-16 as the bone of the model. The objective function consists of three parts: soft similarity loss for preserving the similarity, cross-entropy loss for image representation, and quantization loss for controlling the quality of hash code. Extensive experiments on MFT-fabric-v1, a multi-label dataset, demonstrate that the proposed method outperforms the compared methods. And the retrieval results show that the proposed method achieves effective feature learning and hash learning. In our future works, we will try to extend this approach to other fabric types.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0309200, in part by the Fundamental Research Funds for the Central Universities under Grant JUSRP52007A, in part by the National Natural Science Foundation of China under Grant 61976105.

ORCID iD
Jun Xiang https://orcid.org/0000-0001-5177-0812

References
1. Mezaris V, Kompatsiaris I and Strintzis MG. An ontology approach to object-based image retrieval. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), Barcelona, Spain, 14–17 September 2003, pp.II–511. New York: IEEE.
2. Ko BC, Lee J and Nam J-Y. Automatic medical image annotation and keyword-based image retrieval using relevance feedback. J Digit Imaging 2012; 25: 454–465.
3. Liu Y, Zhang D, Lu G, et al. A survey of content-based image retrieval with high-level semantics. Pattern Recognit 2007; 40: 262–282.
4. Müller H, Müller W, Squire DM, et al. Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recognit Lett 2001; 22: 593–601.
5. Wan J, Wang D, Hoi SCH, et al. Deep learning for content-based image retrieval: a comprehensive study. In: Proceedings of the 22nd ACM international conference on multimedia, Orlando, FL, November 2014, pp.157–166. New York: ACM.
6. Smeulders AW, Worring M, Santini S, et al. Content based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 2000; 22: 1349–1380.
7. Gudivada VN and Raghavan VV. Content based image retrieval systems. Computer 1995; 28: 18–22.
8. Li Z, Xiang J, Wang L, et al. Yarn-dyed fabric image retrieval using colour moments and the perceptual hash algorithm. Fibres Text East Eur 2019; 27: 60–69.
9. Jing J. Patterned fabric image retrieval using color and space features. J Fiber Bioeng Informat 2015; 8: 603–614.
10. Jing J, Li Q, Li P, et al. A new method of printed fabric image retrieval based on color moments and gist feature description. Text Res J 2016; 86: 1137–1150.
11. Deng D, Wang R, Wu H, et al. Learning deep similarity models with focus ranking for fabric image retrieval. Image Vis Comput 2018; 70: 11–20.
12. Zhang L, Liu X, Liu Z, et al. Lace fabric image retrieval based on multi-scale and rotation invariant LBP. In: Proceedings of the 7th international conference on internet multimedia computing and service, Zhangjiajie, Hunan, August 2015, pp.1–5. New York: ACM.
13. Zhang N, Xiang J, Wang L, et al. Image retrieval of wool fabric. Part II: based on low-level color features. Text Res J 2020; 90: 797–808.
14. Xiang J, Zhang N, Pan R, et al. Fabric image retrieval system using hierarchical search based on deep convolutional neural network. *IEEE Access* 2019; 7: 35405–35417.

15. Suciati N, Herumurti D and Wijaya AY. Fractal-based texture and HSV color features for fabric image retrieval. In: *2015 IEEE international conference on control system, computing and engineering (ICCSCE)*, Penang, Malaysia, 27–29 November 2015, pp.178–182. New York: IEEE.

16. Xiang J, Zhang N, Pan R, et al. Fabric retrieval based on multi-task learning. *IEEE Trans Image Process* 2021; 30: 1570–1582.

17. Sivic J and Zisserman A. Video Google: a text retrieval approach to and object matching in videos. In: *IEEE international conference on computer vision*, Nice, France, 13–16 October 2003, p.1470. New York: IEEE.

18. Kuo Y-H, Chen K-T, Chiang C-H, et al. Query expansion for hash-based image object retrieval. In: *Proceedings of the 17th ACM international conference on multimedia*, Beijing, China, 19 October 2009, pp.65–74. New York: ACM.

19. Jegou H, Douze M and Schmid C. Hamming embedding and weak geometric consistency for large scale image search. In: *European conference on computer vision*, Marseille, France, 12–18 October 2008, pp.304–317. Berlin, Heidelberg: Springer.

20. Xie H, Zhang Y, Tan J, et al. Contextual query expansion for image retrieval. *IEEE Trans Multimedia* 2014; 16: 1104–1114.

21. Ledwich L and Williams S. Reduced SIFT features for image retrieval and indoor localisation. In: *Australian conference on robotics and automation*, Canberra, Australian, 6–8 December 2004, p.3.

22. Xie B, Qin J, Xiang X, et al. An image retrieval algorithm based on Gist and sift features. *IJ Network Security* 2018; 20: 609–616.

23. Krizhevsky A, Sutskever I and Hinton GE. ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012; 25: 1097-1105.

24. Fekri-Ershad S and Tajeripour F. Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns. *Appl Artif Intell* 2017; 31: 395–410.

25. Tadi Bani N and Fekri-Ershad S. Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains. *Electron Libr* 2019; 37: 650–666.

26. Wang X, Wu G and Zhong Y. Fabric identification using convolutional neural network. In: *International conference on artificial intelligence on textile and apparel*, Singapore, 17 May 2018, pp.93–100. Cham: Springer.

27. Cai Z, Gao W, Yu Z, et al. Feature extraction with triplet convolutional neural network for content-based image retrieval. In: *2017 12th IEEE conference on industrial electronics and applications (ICIEA)*, Siem Reap, Cambodia, 18–20 June 2017, pp.337–342. New York: IEEE.

28. Xing EP, Ng AY, Jordan MI, et al. Distance metric learning with application to clustering with side-information. *Adv Neural Inf Process Syst* 2002; 15:12.

29. Mignon A and Jurie F. PCCA: a new approach for distance learning from sparse pairwise constraints. In: *2012 IEEE conference on computer vision and pattern recognition*, Providence, RI, 16–21 June 2012, pp.2666–2672. New York: IEEE.

30. Hu J, Lu J and Tan Y-P. Discriminative deep metric learning for face verification in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp.1875–1882. New York: IEEE.

31. Bell S and Bala K. Learning visual similarity for product design with convolutional neural networks. *ACM Trans Graph* 2015; 98: 1–10.

32. Chen D, Yuan Z, Chen B, et al. Similarity learning with spatial constraints for person re-identification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, 27–30 June 2016, pp.1268–1277. New York: IEEE.

33. Pinheiro PO. Unsupervised domain adaptation with similarity learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, 18–23 June 2018, pp.8004–8013. New York: IEEE.

34. Kordopatis-Zilos G, Papadopoulos S, Patras I, et al. ViSiLe: fine-grained spatio-temporal video similarity learning. In: *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, South Korea, 27 October–2 November 2019, pp.6351–6360. New York: IEEE.

35. Oyen D, Kucer M and Wohlberg B. ViSHash: visual similarity preserving image hashing for diagram retrieval. In: *Applications of machine learning 2021*, San Diego, CA, 1 August 2021, p.118430C. Bellingham, WA: SPIE.

36. Lin Z, Ding G, Han J, et al. Cross-view retrieval via probability-based semantics-preserving hashing. *IEEE Trans Cybern* 2017; 47: 4342–4355.

37. Tang H, Xu L and Zeng X. Binary auto-encoders hashing with manifold similarity-preserving for image retrieval. In: *2021 7th international conference on computing and artificial intelligence*, Tianjin, China, 24 September 2021, pp.76–82. New York: ACM.

38. Wu G, Han J, Guo Y, et al. Unsupervised deep video hashing via balanced code for large-scale video retrieval. *IEEE Trans Image Process* 2018; 28: 1993–2007.

39. Liu H, Xiong J, Zhang N, et al. Quadruplet-based deep cross-modal hashing. *Comput Intell Neurosci* 2021; 2021: 9968716.

40. Wu G, Lin Z, Ding G, et al. On aggregation of unsupervised deep binary descriptor with weak bits. *IEEE Trans Image Process* 2020; PP: 9266–9278.

41. Zhao S, Wu D, Zhou Y, et al. Rescuing deep hashing from dead bits problem. *arXiv preprint* arXiv:2102.00648, 2021.

42. Miao Y, Lin Z, Ma X, et al. Learning transformation-invariant local descriptors with low-coupling binary codes. *IEEE Trans Image Process* 2021; 30: 7554–7566.

43. Chen B and Deng W. Deep embedding learning with adaptive large margin N-pair loss for image retrieval and clustering. *Pattern Recognit* 2019; 93: 353–364.

44. Ko B and Gu G. Embedding expansion: augmentation in embedding space for deep metric learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, WA, 13–19 June 2020, pp.7255–7264. New York: IEEE.
45. Bai Y, Lou Y, Gao F, et al. Group-sensitive triplet embedding for vehicle reidentification. *IEEE Trans Multimedia* 2018; 20: 2385–2399.
46. Vijaya Kumar BG, Carneiro G and Reid I. Learning local image descriptors with deep siamese triplet convolutional networks by minimising global loss functions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, 27–30 June 2016, pp.5385–5394. New York: IEEE.
47. Bui T, Ribeiro L, Ponti M, et al. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Comput Vis Image Underst* 2017; 164: 27–37.
48. Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*, Miami, FL, 20–25 June 2009, pp.248–255. New York: IEEE.
49. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
50. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, 27–30 June 2016, pp.770–778. New York: IEEE.
51. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, 7–12 June 2015, pp.1–9. New York: IEEE.
52. Xia R, Pan Y, Lai H, et al. Supervised hashing for image retrieval via image representation learning. In: *Proceedings of the AAAI conference on artificial intelligence*, Québec, Canada, 27–31 July 2014.
53. Li W-J, Wang S and Kang W-C. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.
54. Yuan L, Wang T, Zhang X, et al. Central similarity quantization for efficient image and video retrieval. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, WA, 13–19 June 2020, pp.3083–3092. New York: IEEE.
55. Do TT, Doan A-D and Cheung N-M. Learning to hash with binary deep neural network. In: *European conference on computer vision*, Amsterdam, The Netherlands, 8–16 October 2016, pp.219–234. Cham: Springer.
56. Yang E, Deng C, Liu T, et al. Semantic structure-based unsupervised deep hashing. In: *Proceedings of the 27th international joint conference on artificial intelligence*, Stockholm, Sweden, 13 July 2018, pp.1064–1070.
57. Shen F, Xu Y, Liu L, et al. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE Trans Pattern Anal Mach Intell* 2018; 40: 3034–3044.