



## Article

# Feature Fusion-Based Re-Ranking for Home Textile Image Retrieval

Ziyi Miao <sup>1,†</sup>, Lan Yao <sup>2,\*,†</sup> , Feng Zeng <sup>1,\*,†</sup> , Yi Wang <sup>3</sup> and Zhiguo Hong <sup>3</sup>

<sup>1</sup> School of Computer Science and Engineering, Central South University, Changsha 410083, China; 214712276@csu.edu.cn

<sup>2</sup> School of Mathematics, Hunan University, Changsha 410082, China

<sup>3</sup> Raycloud Technology Company, Hangzhou 310052, China

\* Correspondence: yao@hnu.edu.cn (L.Y.); fengzeng@csu.edu.cn (F.Z.)

† These authors contributed equally to this work.

**Abstract:** In existing image retrieval algorithms, negative samples often appear at the forefront of retrieval results. To this end, in this paper, we propose a feature fusion-based re-ranking method for home textile image retrieval, which utilizes high-level semantic similarity and low-level texture similarity information of an image and strengthens the feature expression via late fusion. Compared with single-feature re-ranking, the proposed method combines the ranking diversity of multiple features to improve the retrieval accuracy. In our re-ranking process, Markov random walk is used to update the similarity metrics, and we propose local constraint diffusion based on contextual similarity. Finally, the fusion–diffusion algorithm is used to optimize the sorted list via combining multiple similarity metrics. We set up a large-scale home textile image dataset, which contains 89k home textile product images from 12k categories, and evaluate the image retrieval performance of the proposed model with the Recall@k and mAP@K metrics. The experimental results show that the proposed re-ranking method can effectively improve the retrieval results and enhance the performance of home textile image retrieval.

**Keywords:** home textile image retrieval; feature fusion; similarity diffusion; fusion diffusion; local constraint diffusion

**MSC:** 68T07



**Citation:** Miao, Z.; Yao, L.; Zeng, F.; Wang, Y.; Hong, Z. Feature Fusion-Based Re-Ranking for Home Textile Image Retrieval. *Mathematics* **2024**, *12*, 2172. <https://doi.org/10.3390/math12142172>

Academic Editors: Juan Gabriel Avina-Cervantes and Konstantin Kozlov

Received: 4 April 2024  
Revised: 20 June 2024  
Accepted: 9 July 2024  
Published: 11 July 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In existing image retrieval algorithms, the retrieval results often have the problem of negative samples listing in the front, which is not expected in real applications. Negative samples are generally images which are of different classes from the query image, while presenting only minor differences. In home textile image, the fine-grained information such as designs and patterns in key areas may be complex, and various classes of images can only be distinguished by small differences. These differences are mainly manifested in the texture and color information of the image, which can be extracted from the lower layers of the network model. Existing methods are mainly based on convolutional neural networks (CNN) to extract image features, which are usually obtained by using the output feature maps of the last convolutional layer in the network structure for aggregation [1,2]. Different levels of features in a convolutional neural network focus on different information of the image, where the higher-level features carry more semantic information, and the lower-level features are relatively concerned with detailed information such as edge texture [3]. Certainly, fusing high-level semantic information and low-level detail information can enhance feature representation.

In general, there are two fusion mechanisms for multiple features, namely early fusion and late fusion. Early fusion usually combines features at the feature level [4,5],

and then uses the merged features as the output features of the model or feeds them into the loss function for training. Late fusion refers to score or decision-level fusion at the image retrieval stage. Early feature fusion can easily lead to excessively high dimensions of feature vectors, reducing retrieval efficiency, and the extraction of traditional image features or lower-level network features can easily affect the image features extracted by neural networks, which may instead lead to a decrease in retrieval accuracy. With the late fusion process, a good balance can be achieved between information content and fusion difficulty [6].

The concept of re-ranking can be utilized to improve retrieval precision. Re-ranking refers to using the nearest neighbor structure to update the similarity between samples after obtaining the image features, thereby improving the ranking of correct matches and reducing the ranking of incorrect matches. The advantage of re-ranking is that it does not require retraining of the network and additional training samples, and it can be directly applied to the initial sorting list. Therefore, after obtaining the initial retrieval ranking list, adding a re-ranking step can effectively improve the accuracy of home textile image retrieval.

Currently, the idea of re-ranking has been applied to fields such as instance retrieval [7], person re-identification [8], and so on. With the ranking metric of Euclidean distance or cosine similarity, the incorrect matches may rank high in the retrieval results, while those of correct matches may be low. To solve this problem and improve the accuracy of home textile image retrieval, we introduce the re-ranking method into the home textile field. Compared with re-ranking algorithms that rely solely on a single feature, re-ranking algorithms based on multiple features can combine similarity information of multiple features, obtaining more accurate ranking lists and thus greatly improving retrieval performance. Different from traditional multi-feature fusion re-ranking methods, we propose a home textile image re-ranking method based on feature similarity fusion, which can simultaneously utilize the semantic similarity at the high level and texture similarity at the shallow level, exploring the ranking diversity of deep features at different levels. Our contributions are mainly summarized as follows.

1. We propose the similarity diffusion process of a single-layer feature, including the construction of a weighted graph, the definition of transition matrices and diffusion methods, and obtain the converged solution after sufficient similarity diffusion.
2. We propose the concept of the  $k$ -context nearest neighbor set. In order to reduce the impact of noise data on the diffusion process, we propose a locality-constrained diffusion process based on context similarity.
3. We propose a fusion diffusion method for multi-layer feature similarity, utilizing the semantic similarity at the higher level and texture similarity at the shallow level, more effectively re-ranking the original home textile image retrieval list.

The remainder of this paper is organized as follows. In Section 2, relevant work in recent years is briefly introduced. We present the details of our image re-ranking method in Section 3. Section 4 shows the experimental results. Finally, we conclude the work in the last section.

## 2. Related Work

To improve the accuracy of retrieval, image re-ranking methods have quickly developed in content-based image retrieval. The advantage of re-ranking is that it does not need to retrain the network or use additional training samples. This process can be directly applied to the initial ranking list without additional data or complex calculations. The core idea of re-ranking is to use the neighbor structure to reevaluate the similarity between samples and thus optimize the ranking results. With the rank of correct matches raising and the rank of wrong matches lowering, re-ranking can significantly improve the accuracy and efficiency of retrieval. The re-ranking algorithms can be divided into two types, namely context re-ranking and manifold re-ranking. Context re-ranking explicitly replaces the similarity of sample pairs with the similarity of the neighbor set of sample pairs [9–11],

and manifold re-ranking implicitly updates the similarity of sample pairs through the Markov chain [12,13].

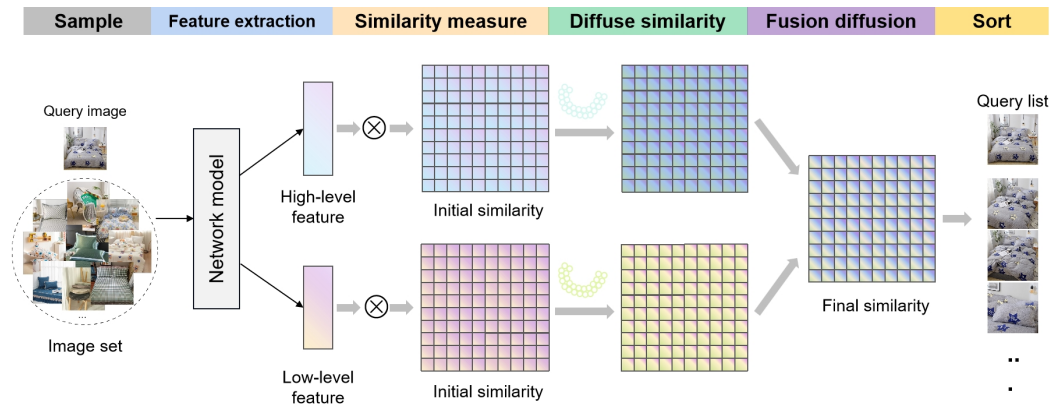
In context re-ranking, some earlier works [14,15] utilized the similarity relationships among images in the initial ranking list, employing methods such as  $k$ -nearest neighbors. SCA [11] calculates the context similarity between two images by comparing  $k$ -nearest neighbor sets, converts the nearest neighbor list into a vector, and introduces an inverted index, which greatly speeds up the speed of re-ranking. However, the  $k$ -nearest neighbor set of the query image is likely to contain wrong matches, and  $k$ -reciprocal nearest neighbor [16,17] is an effective solution to this issue. Compared with  $k$ -nearest neighbor,  $k$ -reciprocal nearest neighbor takes into account the local density of samples around sample  $x_q$ , thus  $k$ -reciprocal nearest neighbor is a stronger similarity measure than  $k$ -nearest neighbor. Lv et al. [9] proposed an extended neighborhood distance (END) for re-ranking, the END distance is calculated according to the top  $t$  images in the query ranking list and the top  $m$  images in the ranking list of  $t$ , and the Jaccard distance and END distance are aggregated for re-ranking. Jayavarthini et al. [10] proposed a context-based extended neighborhood distance re-ranking model, combining Mahalanobis distance and END distance as the final distance for re-ranking.

Manifold re-ranking updates similarities through the diffusion process [13], which can capture the intrinsic manifold geometric structure of the data more deeply, thus more accurately reflecting the relationship between samples. Manifold re-ranking includes three parts which are building a weighted graph, defining a transition matrix, and a diffusion process. Zheng et al. [18] defined the transition matrix as a row stochastic matrix derived from the weighted graph. The value of the transition probability is related to the weight of the edges in the weighted graph and the degree of the vertices. Similarly, Iscen et al. [19] adopted a symmetrical transition matrix, which is also derived from the degree of the vertices and the pairwise similarity. However, this matrix is a symmetrical matrix, so that the similarity information is symmetrically diffused. Chen et al. [20] explored the idea of confining the diffusion of the weighted graph to the local neighborhood. By defining locality through KNN, the random walk is restricted within the range of the  $k$ -nearest neighbors of the data point and the proposed method can adapt to the local probability density and the geometric structure of the underlying manifold.

Different from  $k$ -nearest neighbors and  $k$ -reciprocal nearest neighbors, in this paper, we consider the list similarity between images and use contextual similarity to construct  $k$ -nearest neighbors in manifold re-ranking. Generally speaking, the combination of multiple measurement methods for re-ranking can inevitably result in inconsistent measurement and discrepancies in measurement units. To avoid this issue, we focus on the model itself, combining the extracted low-layer texture similarity and high-layer semantic similarity, and generate a final similarity distance between images for re-ranking.

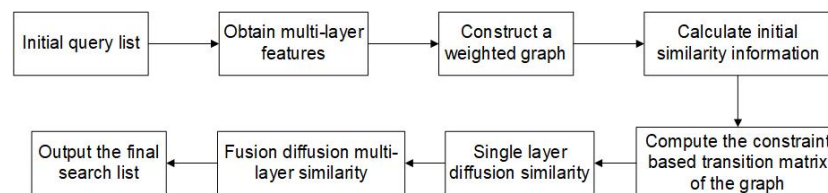
### 3. Materials and Methods

In image retrieval, convolutional neural networks are widely regarded as one of the important tools for achieving efficient and accurate queries. However, training deep convolutional neural networks requires a lot of computational resources, and the training process may be very time-consuming. In addition, different levels of features focus on different image information, and the information of all levels should be all taken into account in image recognition. In order to fully extract information from the depth features at various levels, as well as the sorting diversity of the depth features at different levels, we propose an image re-ranking method based on feature fusion for home textile image retrieval, as shown in Figure 1.



**Figure 1.** The overall framework of feature fusion-based re-ranking method for home textile image retrieval. First, get the high-level and low-level features corresponding to the query image and the retrieval list, and establish different similarity measures. Next, establish Markov random walks to update each similarity measure. Then, use the fusion diffusion algorithm to combine multiple similarity measures, and finally return the final retrieval list based on the size of the distance.

For the query image, we first obtain the retrieval list, then extract the high-level and low-level features of the query image and each image in the retrieval list, and establish different similarity metrics by constructing a weighted graph. Then, a Markov random walk is established to update each similarity measure, realizing local constraint diffusion based on context similarity to capture the geometric structure of the data manifold. Next, the fusion diffusion algorithm is used to combine multiple similarity measures to obtain the final similarity measure. Finally, the re-ranked retrieval list is returned based on the similarity distance between the query image and the searched image. The main flow of the home textile image re-ranking algorithm based on feature fusion is shown in Figure 2.



**Figure 2.** The main flow of feature fusion-based re-ranking algorithm for home textile image retrieval.

In the following, we will introduce each part of the proposed re-ranking algorithm, including the methods of constructing weighted graphs, calculating the initial similarity information and single-layer diffusion similarity in Section 3.1, the method of calculating the constraint-based transfer matrix of graphs in Section 3.2, and the method of fusion and diffusion of multi-layer similarity information in Section 3.3.

### 3.1. Similarity Diffusion for Single-Layer Features

Given a query image  $x_1$ , retrieval list  $Q = \{x_i | i = 2, 3, \dots, n + 1\}$ ,  $x_i$  represents the image that is the  $(i - 1)$ th most similar to the query image in the image set, and  $n$  represents the number of images in the retrieval list that need to be re-ranked. Suppose the list corresponding to all images is  $X = \{x_1, x_2, \dots, x_{n+1}\}$ , and the image vector is  $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n+1}\}$ . In the diffusion process, a weighted graph  $G = \{X, E\}$  is first built,  $X$  is the vertex set, each vertex corresponds to an image,  $E$  represents the edge set, and the weight of the edges is proportional to the similarity  $S_c$  between the data points. It is worth noting that the similarity between two images is represented by the cosine distance between them, and each element in  $E$  is defined as:

$$E(i, j) = S_c(x_i, x_j) = \cos(\mathbf{v}_i, \mathbf{v}_j), i, j = 1, \dots, n + 1, \tag{1}$$

where we have  $|E(i, j)| \leq 1$ , and  $E(i, j) = 1$  when  $i = j$ . This results in a symmetric weighted graph  $\mathbf{G}$ :

$$\mathbf{G} = \begin{bmatrix} E(1,1) & E(1,2) & \cdots & E(1, n+1) \\ E(2,1) & E(2,2) & \cdots & E(2, n+1) \\ \vdots & \vdots & \ddots & \vdots \\ E(n+1,1) & E(n+1,2) & \cdots & E(n, n+1) \end{bmatrix}. \tag{2}$$

To realize the diffusion of similarity information, as done in previous works [21,22], we first obtain the row random matrix  $\mathbf{P}$  from  $\mathbf{G}$ :

$$p_{ij} = \frac{E(i, j)}{\sum_{j=1}^{n+1} E(i, j)}, i, j = 1, 2, \dots, n + 1. \tag{3}$$

$\mathbf{P}$  no longer satisfies the symmetry, while satisfying the property in (4):

$$\sum_{j=1}^{n+1} p_{ij} = 1, i = 1, 2, \dots, n + 1. \tag{4}$$

This operation defines a Markov random walk on the graph  $\mathbf{G}$ .  $\mathbf{P}$  is the transfer matrix and  $p_{ij}$  represents the probability of transferring from vertex  $x_i$  to  $x_j$ . The diffusion process diffuses the similarity information based on the weights of the edges, and this process can be seen as a random walk on the graph.

From a data analysis point of view, the reason for studying the diffusion process is that the transfer matrix  $\mathbf{P}$  contains geometric information about the dataset [23], and the transfer probability defined by the matrix  $\mathbf{P}$  directly reflects the local geometry defined by the nearest neighbors of each vertex in the graph.  $\mathbf{P}(x_i, x_j)$  denotes the probability of transferring from a vertex  $x_i$  to  $x_j$  in one time step, which is proportional to the weight of the edges  $E(x_i, x_j)$ . For  $t \geq 0$ , the probability of transferring from  $x_i$  to  $x_j$  in  $t$  time steps is  $\mathbf{P}^{(t)}(x_i, x_j)$ . In the diffusion process, the similarity information is chained forward over time, and all the localized geometries are gradually captured, and ideally the diffusion process can reveal the underlying geometric structure of the data manifold.

The implementation of diffusion is not unique, and in this paper, we use a simple but effective method [12]:

$$\mathbf{P}^{(t)} = (\alpha\mathbf{P})^1 + (\alpha\mathbf{P})^2 + \cdots + (\alpha\mathbf{P})^t, \tag{5}$$

where  $t$  is a positive integer and  $0 < \alpha < 1$  is a decay coefficient that makes  $\mathbf{P}^{(t)}$  converge as  $t \rightarrow \infty$ . Intuitively,  $\alpha$  controls the diffusivity of  $\mathbf{P}^{(t)}$  at fixed  $t$ : the larger the value of  $\alpha$ , the greater the influence of each vertex on the others. Typically, the literature [24] sets  $\alpha$  between 0.8 and 0.95.

**Theorem 1.** It follows from the literature [22] that  $\mathbf{P}^{(t)}$  converges to a fixed nontrivial solution for arbitrary:

$$\lim_{t \rightarrow \infty} \mathbf{P}^{(t)} = (\mathbf{I}_{n+1} - \alpha\mathbf{P})^{-1} - \mathbf{I}_{n+1}, \tag{6}$$

where  $\mathbf{I}_{n+1}$  is a unit matrix of size  $(n + 1) \times (n + 1)$ .

**Proof.** The proof procedure for the convergence of  $\mathbf{P}^{(t)}$  as  $t \rightarrow \infty$  is as follows:

1. For (5), we can write  $\mathbf{P}^{(t)}$  in the following format:

$$\lim_{t \rightarrow \infty} \mathbf{P}^{(t)} = (\alpha\mathbf{P})^0 + (\alpha\mathbf{P})^1 + (\alpha\mathbf{P})^2 + \cdots + (\alpha\mathbf{P})^t - (\alpha\mathbf{P})^0 = \sum_{t=0}^{\infty} (\alpha\mathbf{P})^t - 1, \tag{7}$$

where  $\sum_{t=0}^{\infty} (\alpha\mathbf{P})^t$  is the form of an infinite geometric progression with the general form  $\sum_{n=0}^{\infty} ar^n$ . If  $a = 1$  and  $r$  is replaced by  $\alpha\mathbf{P}$ , the geometric progression  $\sum_{t=0}^{\infty} (\alpha\mathbf{P})^t$  in the text is obtained.

2. For an infinite geometric series  $\sum_{n=0}^{\infty} ar^n$ , the sum of its first  $n$  terms: when  $r \neq 1$ ,

$$s_n = \sum_{k=0}^n ar^k = a + ar + ar^2 + \dots + ar^n = \frac{a(1 - r^n)}{1 - r}, \tag{8}$$

when  $|r| < 1$ ,  $\lim_{n \rightarrow \infty} r^n = 0$

$$\lim_{n \rightarrow \infty} s_n = \frac{a}{1 - r}. \tag{9}$$

3. For the infinite geometric series  $\sum_{t=0}^{\infty} (\alpha \mathbf{P})^t$  in the text, since  $\mathbf{P}$  is a row random matrix, all its eigenvalues whose absolute values are less than or equal to 1, therefore  $|\mathbf{P}| \leq 1$ ; and  $0 < \alpha < 1$ , we can obtain  $|\alpha \mathbf{P}| < 1$ , which satisfies the condition in step 2.
4. For an infinite geometric series  $\sum_{t=0}^{\infty} (\alpha \mathbf{P})^t$  in the text, find the sum of the first  $t$  terms: when  $|\alpha \mathbf{P}| \neq 1$ ,

$$s_t = \sum_{k=0}^t (\alpha \mathbf{P})^k = (\alpha \mathbf{P})^0 + (\alpha \mathbf{P})^1 + (\alpha \mathbf{P})^2 + \dots + (\alpha \mathbf{P})^t = \frac{1 - (\alpha \mathbf{P})^{t+1}}{1 - \alpha \mathbf{P}}. \tag{10}$$

when  $|\alpha \mathbf{P}| < 1$ ,  $\lim_{t \rightarrow \infty} \alpha \mathbf{P}^t = 0$

$$\lim_{t \rightarrow \infty} s_t = \frac{1}{1 - \alpha \mathbf{P}}. \tag{11}$$

5. From steps 1 and 4 we can conclude:

$$\lim_{t \rightarrow \infty} \mathbf{P}^{(t)} = \sum_{t=0}^{\infty} (\alpha \mathbf{P})^t - 1 = \frac{1}{1 - \alpha \mathbf{P}} - 1 = (\mathbf{I}_{n+1} - \alpha \mathbf{P})^{-1} - \mathbf{I}_{n+1}. \tag{12}$$

The above Equation (6) is proved.  $\square$

Ideally, the diffusion process can reveal the underlying geometric structure of the data manifold. However, the diffusion process is sensitive to noise. If the actual topological structure of the data manifold changes due to noise or outliers, the diffusion process may not be able to capture the correct topological structure. As noise and outliers will affect the distribution of data points, this will cause a certain error in the transition matrix during the diffusion process. At that time, the introduction of local constraints can reduce the impact of noisy data points on the diffusion process.

### 3.2. Local Constraint Diffusion Based on Contextual Similarity

Since the diffusion process is affected by noise and outliers, in order to minimize the effect of these data points, we introduce a locally constrained diffusion process based on contextual similarity.

In the classical diffusion process, all paths between vertices  $x_i$  and  $x_j$  are considered when calculating the probability of walking from vertex  $x_i$  to  $x_j$ . If there are several noisy points in the retrieval list, such as a negative sample image, then the paths through these noisy points affect the calculation of the transfer probability.

In order to solve the above problem, we introduce a method [25] to limit the random walking on the weighted graph to the  $k$ -contextual nearest neighbors of the current data point, which can mitigate the effect of noise on the transfer probability calculation [23]. First, a  $k$ -contextual nearest neighbor graph  $\mathbf{G}_K$  is constructed from the original weighted graph  $\mathbf{G}$ . The vertices of  $\mathbf{G}_K$  are the same as those in  $\mathbf{G}$ , while the weights of the edges are different, and the edge weights in the  $k$ -contextual nearest neighbor graph  $\mathbf{G}_K$  are defined as (13):

$$E_k(i, j) = \begin{cases} E(i, j), & x_j \in Q_k(x_i) \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

where  $Q_k(x_i)$  denotes the set of  $k$ -contextual nearest neighbors of  $x_i$ . When  $x_j$  belongs to the set of  $k$ -contextual nearest neighbors of  $x_i$ , the weight is set to the original similarity  $E(i, j)$ , and the weight between non-contextual nearest neighbor points is set to 0. At this point, the probability of the transfer from vertex  $x_i$  walking to  $x_j$  is:

$$\hat{p}_{ij} = \frac{E_k(i, j)}{\sum_{j=1}^{n+1} E_k(i, j)}, i, j = 1, 2, \dots, n + 1. \tag{14}$$

In this case, the convergent solution of (6) is still satisfied.

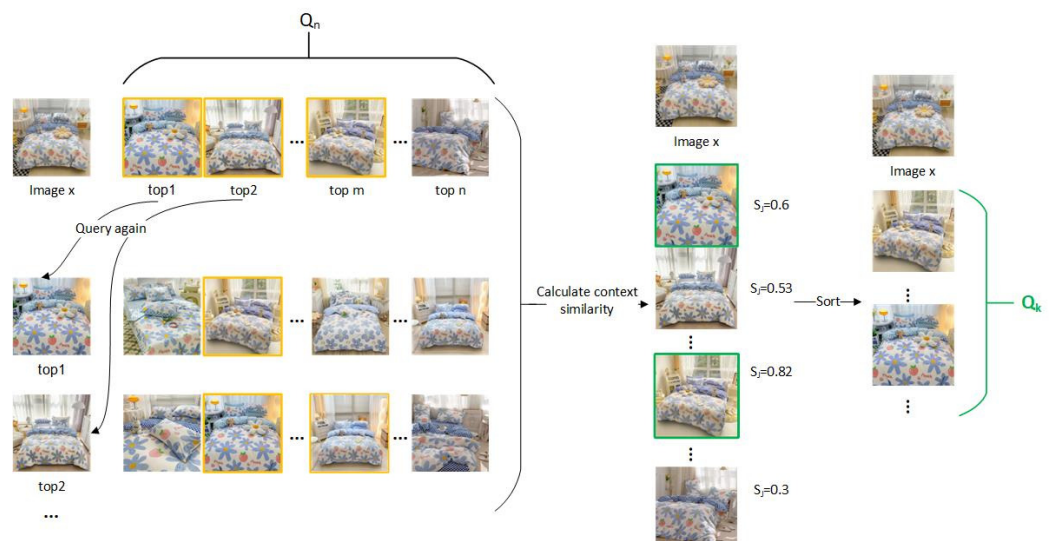
Replacing the original transfer probability matrix  $\mathbf{P}$  with  $\hat{\mathbf{P}}$  reduces the effect of noise. The complete information about the similarity of each data point to all other data points is maintained in the matrix  $\mathbf{P}$ , whereas  $\hat{\mathbf{P}}$  only retains information about the similarity of each data point to its contextual nearest neighbors. Essentially, this assumption is that local similarity (high values) is more reliable than distant similarity, and is an assumption widely adopted by other flow learning algorithms [26,27].

For how to determine the set  $Q_k(x_j)$  and how to define the contextual proximity of two points, contextual similarity is introduced here. Let  $Q_n(x_i)$  denote the list of retrieval results for image  $x_i$  and  $Q_n(x_j)$  denote the list of retrieval results for image  $x_j$ , then the contextual similarity between  $x_i$  and  $x_j$  can be measured by the Jaccard similarity coefficient [9,17]:

$$s_j(x_i, x_j) = \frac{|Q_n(x_i) \cap Q_n(x_j)|}{|Q_n(x_i) \cup Q_n(x_j)|}, \tag{15}$$

where  $|\cdot|$  denotes the size of the bases in the set after intersection or union operations.

Therefore, given an image  $x$ , we first perform a secondary query for each image in its retrieval list  $Q_n(x)$ , obtaining additional  $n$  retrieval lists. Then, we calculate the intersection and union between the two retrieval lists, and calculate the context similarity between  $x$  and each image in the corresponding retrieval list  $Q_n(x)$  through (15). Finally, we select the top  $k$  images ranked by context similarity, denoted as the  $k$  context neighbors of image  $x$ ,  $Q_k$ . That is, the image  $x$  and the images in the set  $Q_k$  are context neighbors. The specific process of obtaining the context neighbors is shown in Figure 3.



**Figure 3.** The specific procedure to get the set of contextual nearest neighbors  $Q_k$  of image  $x$ . The yellow boxes represent images that are duplicated between retrieval lists, i.e., intersections in Jaccard similarity, and the green boxes represent images that are in the top  $k$  in terms of contextual similarity to image  $x$ .

Equation (15) uses additional contextual information but has some limitations. First, computing the intersection and concatenation of two sets of nearest neighbors is very

time-consuming, especially when the Jaccard distance needs to be computed for all image pairs. Second, when computing the set of nearest neighbors, all the weights of the nearest neighbors are the same, and each valid nearest neighbor point is noted as 1, while in reality, the image closer to the query image  $x$  should be more similar to  $x$ .

For the first problem, the set of nearest neighbors can be encoded as simpler but equivalent vectors in two quantities, which is more convenient to compute and can greatly reduce the computational complexity. This is solved here by sparse context encoding [11], which encodes the nearest neighbor sets into nearest neighbor vectors and thus converts the computation of sets into the computation of vectors. Specifically, sparse context encoding converts the set of nearest neighbors  $Q_n(x_i)$  into an  $N_g$ -dimensional nearest neighbor vector  $\Lambda_i$  via an indicator function:

$$\Lambda_i = [v_{i,1}, v_{i,2}, \dots, v_{i,N_g}], \tag{16}$$

where  $N_g$  is much larger than  $n$ , the total number of image libraries. The indicator function  $v_{i,j}$  in the formula is defined as follows:

$$v_{i,j} = \begin{cases} 1, & x_j \in Q_n(x_i) \\ 0, & otherwise \end{cases}. \tag{17}$$

In (17), each term in the binary vector  $\Lambda_i$  indicates whether its corresponding image belongs to the  $n$ -nearest neighbor of  $x_i$ , if it is one of the  $n$ -nearest neighbors of  $x_i$ , then the corresponding term is 1, otherwise it is 0.

For the second Problem, (15) assumes that each nearest neighbor is equal and each element in the set of nearest neighbors has the same weight. To solve this problem, we redistribute the weights based on the original distance between the query image and the images in the retrieval list, and get the new indicator function as follows:

$$v_{i,j} = \begin{cases} S_c(x_i, x_j), & x_j \in Q_n(x_i) \\ 0, & otherwise \end{cases}, \tag{18}$$

where  $S_c$  represents the cosine similarity, as in (1). In this case, the weight of near neighbors is larger and the weight of distant neighbors is smaller.

Based on the definition of the indicator function, the computation of the intersection and concatenation of  $Q_n(x_i)$  and  $Q_n(x_j)$  can be rewritten as a vectorial computation:

$$Q_n(x_i) \cap Q_n(x_j) \Leftrightarrow MIN(\Lambda_i, \Lambda_j) \tag{19a}$$

$$Q_n(x_i) \cup Q_n(x_j) \Leftrightarrow MAX(\Lambda_i, \Lambda_j), \tag{19b}$$

where a *MIN* operation is used to compute the minimum value of the corresponding element in the two input vectors and a *MAX* operation is used to compute the maximum value of the corresponding element in the two input vectors. Next, the base size of the intersection and concatenation can be obtained by calculating the  $L1$  paradigm:

$$|Q_n(x_i) \cap Q_n(x_j)| \Leftrightarrow \|MIN(\Lambda_i, \Lambda_j)\|_1 \tag{20a}$$

$$|Q_n(x_i) \cup Q_n(x_j)| \Leftrightarrow \|MAX(\Lambda_i, \Lambda_j)\|_1, \tag{20b}$$

therefore, we can rewrite the Jaccard similarity in (15) as:

$$s_j(x_i, x_j) = \frac{\|MIN(\Lambda_i, \Lambda_j)\|_1}{\|MAX(\Lambda_i, \Lambda_j)\|_1}. \tag{21}$$

Thus, the problems of slow operation and the same weight of data points in the original Jaccard similarity are solved. First, the context neighbor set  $Q_k$  is obtained based on the new Jaccard similarity formula. Then, the transfer probability matrix  $\hat{P}$ , which is the diffusion



matrix based on local constraints, is obtained through (13) and (14), followed by local constraint diffusion. The converged solution of the diffusion result  $\tilde{\mathbf{P}}$  is finally obtained:

$$\tilde{\mathbf{P}} = \lim_{t \rightarrow \infty} \widehat{\mathbf{P}}^{(t)} = (\mathbf{I}_{n+1} - \alpha \widehat{\mathbf{P}})^{-1} - \mathbf{I}_{n+1}. \tag{22}$$

The converged solution contains the similarity information between the query image as well as the retrieved list.

### 3.3. Convergent Diffusion of Multilayer Features

Different from the diffusion process in Section 3.1, which performs similarity degree diffusion for only one weighted graph, fusion diffusion can solve the diffusion problem for  $m \geq 2$  weighted graphs  $\mathbf{G}_m = (X, E_m)$  simultaneously. From the previous sections of this paper, we can obtain the weighted maps and transfer probability matrices corresponding to the high-level features and the low-level features, denoted as  $\mathbf{G}_{high}$ ,  $\mathbf{G}_{low}$ , and  $\tilde{\mathbf{P}}_{high}$ ,  $\tilde{\mathbf{P}}_{low}$ , respectively. The goal of fusion diffusion is to learn a new similarity metric,  $M$ , which is able to utilize the complementarities between multiple visual features to obtain an enhanced similarity metric.

One way to compute  $M$  is to perform a weighted linear combination of multiple similarity measures:

$$M = \frac{1}{m} \sum_{i=1}^m P_i, \tag{23}$$

where  $P_i$  denotes the transfer matrix of the  $i$ th weighted graph.

This approach is simple and easy to implement, but ignores the correlation between different similarity measures. There are two other fusion strategies including Tensor Product Fusion [28,29] and Cross Diffusion Process [30,31]. The general process of Tensor Product Fusion is that, given two different similarity measures, the tensor product graph (TPG) is first constructed, and then two similarities are jointly diffused on the TPG using a diffusion process. Cross Diffusion algorithms are similar to the idea of co-training, where the two state matrices exchange similarity information with each other during iteration, and two parallel diffusion processes are generated. These methods can effectively utilize the correlation between similarities.

Compared with Tensor Product Fusion, Cross-Diffusion has a smaller computational load. Here, we adopt the alternating diffusion algorithm proposed by Lederman et al. [31] to enact Cross-Diffusion, merging similarity information on two graphs. In the literature, the algorithm was utilized to extract common sources of variation from the measurement results of multiple sensors and defined the alternating diffusion operator  $\mathbf{O}$  and diffusion distance  $d$ . The alternating diffusion operator can capture the structure of common variables in multiple sensors while disregarding any distinct variables present in a single sensor. The diffusion distance has the ability to seize the structure of a chart by measuring the “connectivity” between two samples across the entire sample set, as opposed to comparing the distances between solitary samples, such as cosine distance or Euclidean distance.

To capture the common similarity information in the two transfer probability maps, we can construct the alternating diffusion operator based on the obtained transfer matrix  $\tilde{\mathbf{P}}$ . Then, we compute the diffusion distance between the samples, i.e., the distance between the pictures after fusing the similarities. According to the diffusion distance sorting, that is, the final retrieval list is obtained. The specific process is as follows.

1. Construct the alternating diffusion operator  $\mathbf{O}$ :

$$\mathbf{O} = \tilde{\mathbf{P}}_{high} \tilde{\mathbf{P}}_{low}. \tag{24}$$

2. Calculate the diffusion distance between two samples:

$$d_{ij} = \sum_{l=1}^n (\mathbf{O}_{il} - \mathbf{O}_{jl})^2. \tag{25}$$

Intuitively, alternating diffusion operates on the same set of vertices  $X$ . However, the diffusion process is divided into two steps, with the first step having a transition probability matrix of  $\tilde{\mathbf{P}}_{high}$  and the second step having a transition probability matrix of  $\tilde{\mathbf{P}}_{low}$ . The combination of the two consecutive steps is thus a Markov chain on a new weighted graph  $\mathbf{G} = (X, E)$ , where the transition probabilities are determined by the matrix  $\mathbf{O}$ . The diffusion process can be performed in two steps.

### 3.4. Algorithmic Process

For easy algorithmic representation, the query image is denoted as  $x_1$ , the original retrieval list of the query image is denoted as  $Q$ , and the corresponding image vector lists  $\mathbf{V}^{high}$  and  $\mathbf{V}^{low}$  of the high level and the low level obtained from Resnet50 [32] are denoted as  $\mathbf{V}^1$  and  $\mathbf{V}^2$ , respectively. Firstly, we obtain the weighted graph  $\mathbf{G}$  corresponding to the single-layer features according to the image vector list  $\mathbf{V}$ , realize the locally-constrained diffusion based on the contextual similarity, obtain the transfer matrix convergence solution, then fuse the two similarity measures to get the diffusion distance between images, and finally sort according to the diffusion distance to get the final retrieval list. In the proposed method, most of the computation costs focus on calculating the contextual similarity between images in the matrix. The time complexity of the MIN and MAX operations is  $O(n)$ , so the overall time complexity of the algorithm is  $O(n^3)$ , where  $n$  is the number of images in the retrieval list. The additional space utilized in the algorithm is primarily for storing neighboring vectors, and the space complexity is  $O(n * N_g)$ , where  $N_g$  is the total number of the test set.

The specific processing flow is shown in Algorithm 1.

---

#### Algorithm 1 Feature fusion-based re-ranking of home textile images.

---

**Input:**

$x_1$ : Query image

$\mathbf{V}^1 = \{v_i | i = 1, 2, \dots, n + 1\}, \mathbf{V}^2 = \{v_i = 1, 2, \dots, n + 1\}$ : Image vector lists

$Q = \{x_i | i = 2, 3, \dots, n + 1\}$ : Initial query list

$k$ : Parameters in locally constrained diffusion

**Output:**

$Q^*$ : Final query list

```

1: for  $i = 1$  to 2 do
2:   Calculate  $E^i$  according to (1)
3:    $\Lambda_1 = [v_1]$ 
4:   for  $x_j$  in  $Q$  do
5:     Calculate  $v_{1,j}$  according to (18) and  $\mathbf{V}^1$ 
6:   end for
7:   for  $x_j (j \neq 1)$  in  $Q$  do
8:     Get the search list  $Q_j$  queried by  $x_j$ 
9:      $\Lambda_j = [v_j]$ 
10:    for  $x_k$  in  $Q_j$  do
11:      Calculate  $v_{j,k}$  according to (18)
12:    end for
13:  end for
14:   $\mathbf{S} = [1]$ 
15:  for  $i = 1$  to  $n + 1$  do
16:    for  $j = i + 1$  to  $n + 1$  do
17:      Calculate  $\|MIN(\Lambda_i, \Lambda_j)\|_1$  according to (20a)
18:      Calculate  $\|MAX(\Lambda_i, \Lambda_j)\|_1$  according to (20b)
19:      Calculate context similarity  $S(x_i, x_j)$  and  $S(x_j, x_i)$  according to (21)
20:    end for
21:  end for
22:  Sort  $\mathbf{S}$  to obtain a set of  $k$ -contextual nearest neighbors
23:  Calculate the weight  $E_k^i$  according to (13)
24:  Calculate the transfer matrix  $\tilde{\mathbf{P}}^i$  according to (14)
25:  Calculate the convergent solution  $\tilde{\mathbf{P}}^i$  according to (22)
26: end for
27: Calculate the alternating diffusion operator  $\mathbf{O}$  according to (24)
28: for  $q = 2$  to  $n + 1$  do
29:   Calculate the diffusion distance  $d_{1,q}$  between the query image  $x_1$  and the retrieved image  $x_q$ 
30: end for
31: The final search list  $Q^*$  is obtained by sorting according to diffusion distance
32: return  $Q^*$ 

```

---

## 4. Experiments

### 4.1. Dataset

The original large-scale home textile image dataset comes from Raycloud Technology Company, Hangzhou, China (<https://www.raycloud.com/> (accessed on 26 May 2022)), which contains 5,511,074 images of 512,231 products. Each product has 3–8 images, and each image contains attributes such as product id, first-level category, second-level category, third-level category, and so on. We selected images of 250 random ids from each third-level category for the experiment, resulting in a final dataset containing 89,399 images from 11,973 categories. Table A1 in Appendix A shows the distribution of the dataset used in the experiment. In the dataset, 43,800 images from 5986 categories are selected for training and validation, 45,599 images from 5987 categories are used for testing, and the ratio of training and validation sets is 4:1. Table 1 shows the details of the used dataset.

**Table 1.** The dataset used in the experiment.

	Id Quantity	Image Quantity
Training set + Validation set	5986	43,800
Test set	5987	45,599
Total	11,973	89,399

### 4.2. Evaluation Indicators

In our experiment, we use the standard Recall@K [33] to evaluate image retrieval performance. If the returned image has the same label as the queried image, it is considered a correct return. Otherwise, it is an incorrect return. For a query image, if there are  $T_k$  correct results in the top  $K$  returned images, and as long as  $T_k$  is greater than 0, the *score* of this query equals 1, otherwise it is 0. For the query set, Recall@K is the average of the *score<sub>q</sub>* of each image in the dataset. The specific calculation formula is as follows:

$$score = \begin{cases} 0, & T_k = 0 \\ 1, & T_k > 0 \end{cases}, \quad (26)$$

$$Recall@K = \frac{1}{r} \sum_{q=1}^r score_q, \quad (27)$$

where  $r$  represents the number of all query images and *score<sub>q</sub>* represents the *score* metric of the  $q$ -th image.

To further evaluate the ranking performance of the model, we also use the Mean Average Precision (mAP) [34] metric. Specifically, mAP first requires the calculation of the Average Precision (AP) score for each query, and then the average of the APs for all query images is taken as the final score. The formula for mAP is:

$$mAP = \frac{1}{r} \sum_{q=1}^r AP(q), \quad (28)$$

$$AP = \frac{1}{R} \sum_{K=1}^n (p(K) \cdot rel(K)), \quad (29)$$

where  $r$  represents the total number of query images and  $q$  represents the current query image.  $R$  represents the total number of images of the same category as  $q$  in the dataset.  $K$  represents the ranking position,  $p(K)$  is the proportion of correct returns in the top  $K$  results, and  $rel(K)$  represents the score of the image at position  $K$ , which is 1 if correct and 0 otherwise.

### 4.3. Experimental Environment

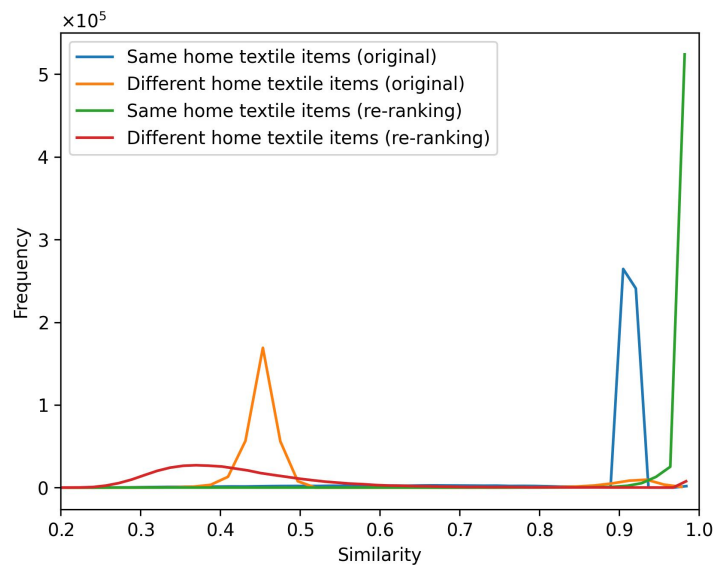
To ensure the fairness of performance comparison, we use Resnet50 as the CNN feature extractor. The image's id is used as the label for metric learning training, with ArcFace [35] as the loss function. The Resnet50 feature extractor was pre-trained on ImageNet [15] using the Pytorch framework. Features were extracted from the last convolutional layer of Resnet50 with adaptive average pooling. We used Adam as the optimizer. The image size was set to  $256 \times 256$ , the batch size was set to 48, the maximum epoch number was 100, and the learning rate was initialized to 0.0001. Data augmentation was used during training, such as horizontal flipping, random rotation, and so on. All the experiments were performed on GEFORCE RTX3080 Ti graphics processing unit. In the testing phase, the dimension of the final global representation extracted from the stage-2 layer was 256 and the dimension of the final global representation extracted from the stage-5 layer was 512.

### 4.4. Experimental Results

#### 4.4.1. Similarity Distribution

Theoretically, in terms of the magnitude of similarity, it is better for the same home textile products to have the bigger similarity values, while the different home textile commodities to have smaller similarity values. In terms of the density of similarity, it is better for the same home textile commodities to have the smaller similarity value range, i.e., the smaller variance between the similarity values of the same commodities is better. On the contrary, it is better for the different home textile commodities to have the larger similarity value range, i.e., the larger variance between the similarity values of the different home textile commodities is the better.

As can be seen in Figure 4, compared with cosine similarity, our method significantly improves the distribution of similarity, as the similarity between the same home textile commodities is large and almost concentrated at 1, while the similarity between different home textile commodities is smaller and has a clearer boundary with the density curve of the same home textile goods. Specifically, without the re-ranking, the similarity value of the same home textile commodities mainly fluctuates between 0.9 and 0.95, with a large variance, and the similarity value of the different home textile commodities mainly fluctuates between 0.4 and 0.45. The difference between the peak similarity value of the same home textile commodities and the peak similarity value of the different home textile commodities is about 0.5. After re-ranking, the similarity of the same home textile commodities mainly focuses on the range between 0.95 and 1 with very small variance, while the similarity value of the different home textile commodities mainly fluctuates between 0.25 and 0.6. The difference between the peak similarity value of the same home textile commodities and the peak similarity value of the different home textile commodities is about 0.6, which is larger than the difference between the peak values in the cosine similarity curve. Therefore, the re-ranking is able to increase the similarity value of the same home textile commodities and decrease the similarity value of the different home textile commodities and the boundaries of the two kinds of similarity curves are clearer.



**Figure 4.** The frequency distribution of similarity in the dataset using the proposed re-ranking and the original cosine distance sorting.

#### 4.4.2. Comparison with Other Approaches

The Recall and mAP accuracy of our method is compared with several other re-ranking algorithms on the same dataset. We select eight re-ranking algorithms as the comparison objects, which are SCA [11], Expanded\_k-Re [17], RLS\_k-Re [16], END\_k-Re [9], M\_END [10], RDP [36], RDPAC [37], and MVD [38]. The first five methods have been introduced in Section 2, and the parameters of the methods are shown in Table 2. RDP, RDPAC, and MVD are manifold re-ranking methods. RDP is a diffusion process on tensor product graph. RDPAC effectively approximates the diffusion process using ranking-based information while ensuring its convergence. MVD is a multi-view graph learning method, in which multiple affinity graphs are fused together via a weight learning scheme based on the unsupervised graph smoothness and utilised as a consensus prior to the diffusion. The parameter  $\mu$  of RDP and MVD is set to 0.01, and the parameter  $\epsilon$  of RDPAC is set to  $5 \times 10^{-5}$ . The baseline stands for the original cosine similarity. In our method, the features extracted from Stage 2 in Resnet50 serve as low-level features, while those from Stage 5 provide high-level features for subsequent similarity fusion.

**Table 2.** The parameters of methods in the comparison experiment.

Method	k	$\lambda$	t	q	K	c
SCA [11]	10	-	-	-	-	-
Expanded_k-Re [17]	15	0.3	-	-	-	-
RLS_k-Re [16]	-	-	3	4	15	-
END_k-Re [9]	15	0.8	3	8	-	2
M_END [10]	-	0.8	3	8	-	2
Ours	8	-	-	-	-	-

As can be seen from Tables 3 and 4, the proposed re-ranking algorithm has achieved the optimal retrieval performance. Our method achieved performance scores of 89.44%, 93.56%, and 94.05% on the performance indicators of Recall@1, Recall@5, and Recall@10, respectively. Compared with the baseline performance, the proposed algorithm had the scores improved by 1.3%, 0.6%, and 0.2%, respectively. On the mAP@5, mAP@10, and mAP@20 metrics, our method achieved scores of 74.79%, 66.75%, and 60.13%, respectively, with improvements of 6.2%, 5.5%, and 1.4% compared to the baseline. The first five methods are context-based re-ranking methods, which scored lower than the baseline on the Recall metric and partially higher on the mAP metric. These methods optimize the original retrieval list by utilizing k-reciprocal nearest neighbors or extended distances, but do not leverage

the underlying manifold structure of the data. While they enhance the ranking of positive samples to a certain extent, they also tend to introduce more negative samples. Compared to RDPAC and MVD, RDP exhibits lower retrieval accuracy due to its reliance solely on high-level semantic similarity information for the diffusion process. RDPAC and MVD calculate all paths between images during their diffusion process, introducing noise into the similarity propagation, resulting in lower retrieval accuracy than our proposed method. Overall, after obtaining the initial 20 retrieval results, our method has a higher accuracy rate at Recall@1 and Recall@5 after re-ranking. The accuracy of each method is relatively close at Recall@10, and our method also has a superior accuracy rate. The improvement in mAP also confirms that our method effectively reduces the ranking of negative samples and enhances the ranking of positive samples.

**Table 3.** Comparison of Recall@K on home textile dataset between the proposed method and other representative methods.

Method	R@1	R@5	R@10	R@20
Baseline	88.18%	93.00%	93.85%	94.45%
SCA [11]	84.59%	91.14%	93.44%	94.45%
Expanded_k-Re [17]	70.60%	91.00%	93.77%	94.45%
RLS_k-Re [16]	78.73%	88.30%	92.99%	94.45%
END_k-Re [9]	78.99%	87.11%	93.68%	94.45%
M_END [10]	76.34%	84.50%	92.51%	94.45%
RDP [36]	88.75%	90.35%	91.45%	94.45%
RDPAC [37]	88.81%	92.90%	93.61%	94.45%
MVD [38]	88.77%	90.37%	91.47%	94.45%
<b>Ours</b>	<b>89.44%</b>	<b>93.56%</b>	<b>94.05%</b>	94.45%

The bold in the results represents the optimal value.

**Table 4.** Comparison of mAP@K on home textile dataset between the proposed method and other representative methods.

Method	mAP@5	mAP@10	mAP@20
Baseline	68.57%	61.29%	58.69%
SCA [11]	72.51%	61.37%	58.98%
Expanded_k-Re [17]	68.77%	60.40%	58.22%
RLS_k-Re [16]	73.24%	61.34%	58.00%
END_k-Re [9]	73.62%	60.86%	58.29%
M_END [10]	70.88%	60.66%	58.09%
RDP [36]	71.29%	63.87%	57.27%
RDPAC [37]	71.84%	65.39%	58.50%
MVD [38]	71.85%	65.69%	59.63%
<b>Ours</b>	<b>74.79%</b>	<b>66.75%</b>	<b>60.13%</b>

The bold in the results represents the optimal value.

Table 5 lists the Recall@19 accuracies on the eight first-level categories in our home textile dataset. Our method performed better than other methods in seven categories. Compared to the baseline, SCA performs slightly worse in most categories. Expanded\_k-Re, RLS\_k-Re, and END\_k-Re, M\_END show a slight decrease in performance in some categories, while maintaining a slight advantage in others. Among them, END\_k-Re and M\_END exhibit lower performance in the “Bathroom supplies” and “Protective gear” categories. This could be attributed to the subtle features and minimal inter-class distinctions of these products, leading to more erroneous samples introduced during extended neighborhood calculation. RDP and RDPAC demonstrate performances close to or slightly better than the baseline across multiple categories. Our approach consistently outperforms others in most categories, particularly in “Bedding sets”, “Quilts”, “Pillows”, etc. These product categories have more noticeable and complex colors and patterns. By utilizing the contextual information of images and low-level texture similarity information, our method can more accurately assess image similarity. Therefore, in these categories, our method achieves superior retrieval lists and better performance.

**Table 5.** The comparison of Recall@19 retrieval accuracy across eight first-level categories.

Method	Bedding Sets	Quilts	Pillows	Mattresses	Bathroom Supplies	Protective Gear	Indoor Decorations	Children's Goods
Baseline	95.41%	93.64%	96.71%	93.51%	89.66%	92.26%	90.90%	92.70%
SCA	95.10%	93.23%	96.43%	92.69%	89.02%	91.91%	90.54%	92.17%
Expanded_k-Re	94.62%	93.23%	96.66%	93.10%	89.92%	91.81%	<b>91.60%</b>	92.93%
RLS_k-Re	94.53%	92.92%	96.49%	92.88%	88.50%	91.13%	89.79%	91.58%
END_k-Re	94.81%	92.96%	96.60%	93.14%	89.27%	91.08%	90.33%	92.46%
M_END	94.70%	92.93%	96.62%	93.25%	89.27%	90.97%	90.21%	92.34%
RDP	95.19%	93.40%	96.64%	93.36%	89.27%	91.94%	90.56%	92.58%
RDPAC	95.41%	93.63%	96.71%	93.58%	89.66%	92.21%	90.90%	92.75%
MVD	95.18%	93.40%	96.64%	93.36%	89.27%	91.91%	90.48%	92.58%
<b>Ours</b>	<b>95.86%</b>	<b>94.09%</b>	<b>97.17%</b>	<b>93.91%</b>	<b>90.14%</b>	<b>92.72%</b>	91.34%	<b>93.29%</b>

The bold in the results represents the optimal value.

In Figure 5, the top 15 retrieval results of the initial cosine similarity ranking and the proposed re-ranking algorithm are listed for the comparison. In each row, the first image is the query image, the images with green frames are correct returns (same product), and the images with red frames are incorrect returns. After re-ranking, the meaning of the yellow box is that the negative sample is correctly replaced by a positive sample at that position. From the figure, it can be seen that for some negative samples with similar appearance outlines and semantics (such as the top6 in the first row, the top3 in the second row, and the top3 in the third row), their rankings can be lowered after merging texture similarity and re-ranking (corresponding to the top12 in the first row, top6 in the second row, and top5 in the third row after re-ranking). For some positive samples with high texture similarity but dissimilar appearances (such as the top13 in the second row and the top5 in the third row), their ranking can be improved after re-ranking (corresponding to the top8 in the second row and the top2 in the third row after re-ranking). Overall, our re-ranking algorithm can alleviate the situation of negative samples being at the forefront of the original retrieval list, improve the ranking of positive samples, and return an optimized retrieval list.

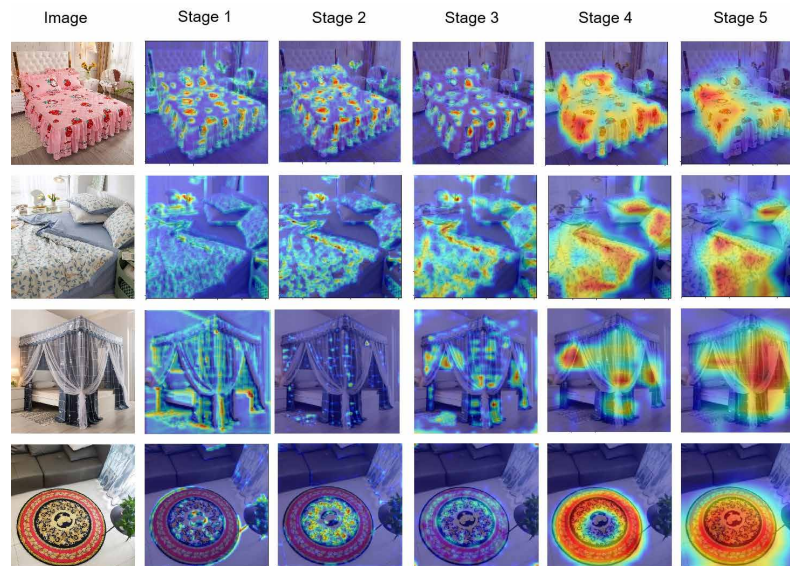


**Figure 5.** Example of retrieval results using original cosine similarity and the proposed re-ranking algorithm. The red box represents incorrect returns and the green and yellow boxes represent correct returns, where the meaning of the yellow box is that the negative sample is correctly replaced by the positive sample at that position.

#### 4.4.3. Multilayer Characterization of Resnet50

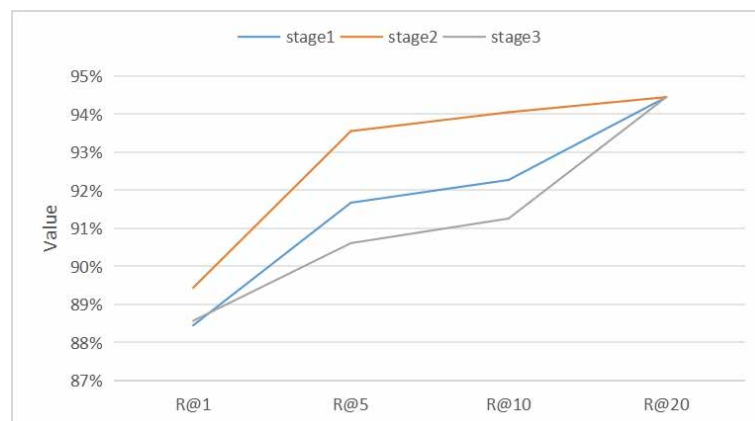
Different layers of the convolutional neural network have different attribute features. In the bottom layer of the network, the extracted features are mainly detailed features such as texture and color. In the middle layer, most of the extracted features are style details, etc., and in the higher layers of the network, most of the extracted features are the overall features of the image. In this paper, Resnet50 is used as the feature extraction network. The Resnet50 model consists of five stages. Stage 1 is the lowest layer of the model and

Stage 5 is the highest layer of the model. For a given image, the Grad-CAM [39] method is used to generate a heat map to visualize the features extracted from each stage as shown in Figure 6. For the query image, the features extracted by Stage 4 and Stage 5 carry more semantic information, while those from Stages 1 to 3 carry more texture information.



**Figure 6.** Heat maps corresponding to different stages in Resnet50. Stage 4 and Stage 5 extract more semantic information; Stage 1 to Stage 3 extract more pattern and detail information.

In general image retrieval frameworks and the output features from the highest layer of the model are used as the final representation, and similarity retrieval is conducted based on these features. The re-ranking method we proposed aims to combine the lower-level texture similarity, therefore we still use the Stage 5 of the model to extract the semantic similarity. In order to accurately integrate high-level semantic similarity and low-level texture similarity, we separately treat Stages 1–3 as the low-level features to be fused, calculate the texture similarity, and combine it with the semantic similarity calculated from the features extracted by Stage 5. The experimental results are shown in Figure 7. The re-ranking method integrated by Stage 2 achieves the optimal retrieval precision, reaching 89.44%, 93.56%, and 94.05% on Recall@1, Recall@5, and Recall@10 performance indicators, respectively, which are 1.0%, 1.9%, and 1.8% higher compared to Stage 1, and 0.9%, 2.9%, and 2.8% higher compared to Stage 3. Therefore, Stage 2 is used as the low-level feature in our re-ranking method.



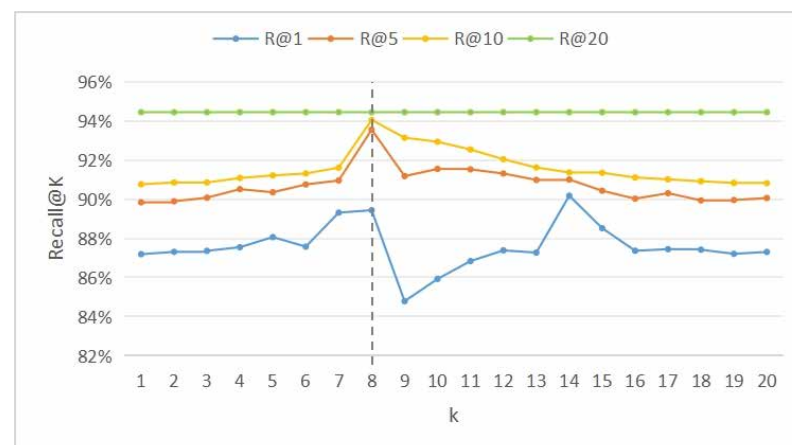
**Figure 7.** The impact of different Stages as low-level features on the precision of re-ranking retrieval. The metric is Recall@K, where K takes 1, 5, 10, and 20.



#### 4.4.4. Parameter Analysis

The proposed re-ranking algorithm has an important parameter, that is the local constraint neighboring parameter  $k$ . In Section 3.2, in order to mitigate the influence of noise on the calculation of transition probabilities, we have the random walk constrained on the weighted graph within the range of  $k$  contextual neighbors of the current data point. Hereinafter, we will discuss the impact of the neighboring parameter  $k$  on the Recall indicator, with the feature baseline being Resnet50.

The experimental results are shown in Figure 8. When  $k < 8$ , the indexes of the Recall@5 and Recall@10 performances steadily increase and then tend to the maximum value, and the precision of the two retrieval indexes shows a downward trend when  $k > 8$ . Recall@1 achieves larger values at  $k = 8$  and  $k = 14$ . Generally, our proposed algorithm is relatively sensitive to the parameter  $k$ . As  $k$  increases, the Recall index shows an upward and then downward trend in general, reaching the optimal precision value when  $k = 8$ . The size of value  $k$  determines the  $k$ -nearest neighbor set of data points, which further affects the process of local constraint diffusion. From the experiments, we can find that too large a  $k$  value will lead to more impact of noise data points on the diffusion process, while too small a  $k$  value will prevent similarity from being better propagated.



**Figure 8.** The impact of parameter  $k$  on the precision of re-ranking retrieval on the home textile dataset. The possible values of  $k$  are integers from 1 to 20.

## 5. Conclusions

In this paper, we have proposed a home textile image retrieval re-ranking method based on feature fusion to solve the problem of negative samples advancing in home textile image retrieval. This proposed method integrates high-level semantic similarity information and low-level detail similarity information, which can improve the accuracy of retrieval. Firstly, we have established a Markov random walk to update each similarity measure. To alleviate the impact of noise data points on the diffusion process, we have proposed a locality-constrained diffusion based on context similarity, restricting the random walk on the weighted graph within the range of the current data point's  $k$  context neighbors. To utilize the correlation among multiple similarity measures, we have adopted the alternating diffusion algorithm to implement cross-diffusion, blending the similarity information on two graphs. Finally, experimental results have shown that the proposed re-ranking method have better performance than other classic algorithms, and can effectively improve the performance of home textile image retrieval.

**Author Contributions:** Z.M. and F.Z. conceived and designed the experiments; Z.M. performed the experiments; Z.M. and L.Y. analyzed the data; Y.W. and Z.H. contributed reagents/materials/analysis tools; Z.M. and L.Y. wrote the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported in part by the National Science Foundation of China (Grant No. 62172450).









**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

**Acknowledgments:** The authors would like to thank the Raycloud company for the big data and support of data processing.

**Conflicts of Interest:** Authors Yi Wang and Zhiguo Hong were employed by the Raycloud Technology Company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Appendix A

**Table A1.** Distribution of the dataset used in the experiment.

First-Level Category ID	Main Products	Second-Level Category Count	Third-Level Category Count	Number of Images	Image Examples
11	Bedding sets	3	43	26,832	
12	Quilts	3	21	13,305	
13	Pillows	2	11	23,137	
14	Mattresses	2	11	5404	
15	Bathroom supplies	2	7	1636	
16	Protective gear	2	10	6971	
17	Indoor decorations	5	29	9013	
18	Children's goods	3	13	3101	

## References

1. Ayyachamy, S.; Alex, V.; Khened, M.; Krishnamurthi, G. Medical image retrieval using Resnet-18. In Proceedings of the Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications, San Diego, CA, USA, 17–18 February 2019; pp. 233–241.
2. Luo, Y.; Li, W.; Ma, X.; Zhang, K. Image retrieval algorithm based on locality-sensitive hash using convolutional neural network and attention mechanism. *Information* **2022**, *13*, 446. [\[CrossRef\]](#)
3. Ren, J.; Wang, Z.; Ren, J. PS-Net: Progressive Selection Network for Salient Object Detection. *Cogn. Comput.* **2022**, *14*, 794–804. [\[CrossRef\]](#)
4. Khan, F.S.; Van de Weijer, J.; Vanrell, M. Modulating shape features by color attention for object recognition. *Int. J. Comput. Vis.* **2012**, *98*, 49–64. [\[CrossRef\]](#)

5. Khan, F.S.; Anwer, R.M.; Van De Weijer, J.; Bagdanov, A.D.; Vanrell, M.; Lopez, A.M. Color attributes for object detection. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3306–3313.
6. Zheng, L.; Wang, S.; Tian, L.; He, F.; Liu, Z.; Tian, Q. Query-adaptive late fusion for image search and person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1741–1750.
7. Qian, X.; Lu, D.; Wang, Y.; Zhu, L.; Tang, Y.Y.; Wang, M. Image re-ranking based on topic diversity. *IEEE Trans. Image Process.* **2017**, *26*, 3734–3747. [[CrossRef](#)] [[PubMed](#)]
8. Garcia, J.; Martinel, N.; Gardel, A.; Bravo, I.; Foresti, G.L.; Micheloni, C. Discriminant context information analysis for post-ranking person re-identification. *IEEE Trans. Image Process.* **2017**, *26*, 1650–1665. [[CrossRef](#)] [[PubMed](#)]
9. Lv, J.; Li, Z.; Nai, K.; Chen, Y.; Yuan, J. Person re-identification with expanded neighborhoods distance re-ranking. *Image Vis. Comput.* **2020**, *95*, 103875. [[CrossRef](#)]
10. Jayavarthini, C.; Malathy, C. Deep convolution neural network with context based expanded neighbourhoods distance re-ranking model for person re-identification. *Multimed. Tools Appl.* **2022**, *81*, 5957–5971. [[CrossRef](#)]
11. Bai, S.; Bai, X. Sparse contextual activation for efficient visual re-ranking. *IEEE Trans. Image Process.* **2016**, *25*, 1056–1069. [[CrossRef](#)]
12. Pang, S.; Ma, J.; Zhu, J.; Xue, J.; Tian, Q. Improving object retrieval quality by integration of similarity propagation and query expansion. *IEEE Trans. Multimed.* **2018**, *21*, 760–770. [[CrossRef](#)]
13. Donoser, M.; Bischof, H. Diffusion Processes for Retrieval Revisited. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1320–1327.
14. Shen, X.; Lin, Z.; Brandt, J.; Avidan, S.; Wu, Y. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3013–3020.
15. Ye, M.; Liang, C.; Yu, Y.; Wang, Z.; Leng, Q.; Xiao, C.; Chen, J.; Hu, R. Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *IEEE Trans. Multimed.* **2016**, *18*, 2553–2566. [[CrossRef](#)]
16. Chen, Y.; Yuan, J.; Li, Z.; Wu, Y.; Nouioua, M.; Xie, G. Person re-identification based on re-ranking with expanded k-reciprocal nearest neighbors. *J. Vis. Commun. Image Represent.* **2019**, *58*, 486–494. [[CrossRef](#)]
17. Zhong, Z.; Zheng, L.; Cao, D.; Li, S. Re-ranking Person Re-identification with k-Reciprocal Encoding. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3652–3661.
18. Zheng, D.; Fan, J.; Han, M. Hybrid Regularization of Diffusion Process for Visual Re-Ranking. *IEEE Trans. Image Process.* **2021**, *30*, 3705–3719. [[CrossRef](#)]
19. Iscen, A.; Toliás, G.; Avrithis, Y.; Furon, T.; Chum, O. Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 926–935.
20. Chen, X.; Yang, Y. Diffusion K-means clustering on manifolds: Provable exact recovery via semidefinite relaxations. *Appl. Comput. Harmon. Anal.* **2021**, *52*, 303–347. [[CrossRef](#)]
21. Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; Schölkopf, B. Ranking on data manifolds. *Adv. Neural Inf. Process. Syst.* **2003**, *16*, 169–176.
22. Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; Schölkopf, B. Learning with Local and Global Consistency. In Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS 2003), Vancouver, BC, Canada, 6–12 December 2004; pp. 321–328.
23. Yang, X.; Koknar-Tezel, S.; Latecki, L.J. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 357–364.
24. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The Pagerank Citation Ranking: Bringing Order to the Web*; Stanford Digital Libraries: Stanford, CA, USA, 1999.
25. Szummer, M.; Jaakkola, T. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2002; p. 945.
26. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [[CrossRef](#)] [[PubMed](#)]
27. Tenenbaum, J.B.; Silva, V.d.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323. [[CrossRef](#)]
28. Zhou, Y.; Bai, X.; Liu, W.; Latecki, L. Fusion with diffusion for robust visual tracking. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2978–2986.
29. Zhou, Y.; Bai, X.; Liu, W.; Latecki, L.J. Similarity fusion for visual tracking. *Int. J. Comput. Vis.* **2016**, *118*, 337–363. [[CrossRef](#)]
30. Talmon, R.; Wu, H.T. Latent common manifold learning with alternating diffusion: Analysis and applications. *Appl. Comput. Harmon. Anal.* **2019**, *47*, 848–892. [[CrossRef](#)]
31. Lederman, R.R.; Talmon, R. Learning the geometry of common latent variables using alternating-diffusion. *Appl. Comput. Harmon. Anal.* **2018**, *44*, 509–536. [[CrossRef](#)]

32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Su, H.; Wang, P.; Liu, L.; Li, H.; Li, Z.; Zhang, Y. Where to look and how to describe: Fashion image retrieval with an attentional heterogeneous bilinear network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3254–3265. [[CrossRef](#)]
34. Lyou, E.; Lee, D.; Kim, J.; Lee, J. Modality-Aware Representation Learning for Zero-shot Sketch-based Image Retrieval. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 5646–5655.
35. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4685–4694.
36. Bai, S.; Bai, X.; Tian, Q.; Latecki, L.J. Regularized diffusion process for visual retrieval. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3967–3973.
37. Guimaraes Pedronette, D.C.; Pascotti Valem, L.; Latecki, L.J. Efficient rank-based diffusion process with assured convergence. *J. Imaging* **2021**, *7*, 49. [[CrossRef](#)] [[PubMed](#)]
38. Li, Q.; An, S.; Li, L.; Liu, W.; Shao, Y. Multi-View Diffusion Process for Spectral Clustering and Image Retrieval. *IEEE Trans. Image Process.* **2023**, *32*, 4610–4620. [[CrossRef](#)] [[PubMed](#)]
39. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.