Modelling Diffusion Process by Deep Neural Networks for Image Retrieval

Yan Zhao[1]
yz298@uowmail.edu.au
Lei Wang[1]
leiw@uow.edu.au
Luping Zhou[2]
luping.zhou@sydney.edu.au
Yinghuan Shi[3]
syh@nju.edu.cn
Yang Gao[3]
gaoy@nju.edu.cn

[1] School of Computing and Information Technology
University of Wollongong Australia
Wollongong, Australia

[2] School of Electrical and Information Engineering
The University of Sydney
Sydney, Australia

[3] State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China

Abstract

By considering the underlying neighbourhood structure of images, diffusion process can better evaluate image similarity and has proven highly effective in improving image retrieval. Nevertheless, diffusion process stores a large neighbourhood graph, costs more online retrieval time, and requires special algorithms other than simple Euclidean search. To address these issues, this paper proposes to treat diffusion process as a "black box" and directly model it by training deep neural networks, so as to obtain better image representation that assimilates the effect of diffusion process and works with Euclidean search. We firstly put forward a kernel mapping interpretation to diffusion process, and then formulate the modelling as a deep metric learning problem. The proposed approach is unsupervised in the sense that it needs neither image labels nor external datasets, and completely avoids online diffusion process in retrieval. More interestingly, we find that this approach could even achieve better retrieval than the original diffusion process, instead of merely approximating it. Experiments verify its effectiveness and investigate its appealing characteristics such as the generalisation to new image insertion.

1 Introduction

Content-based image retrieval aims to retrieve from an image database the images that can meet the requirement set by a user, and the typical scenario may be to find the images visually similar to a query of example. As an important topic in computer vision, image retrieval has received intensive research and gained significant progress during the past two decades [8, 25, 32]. In particular, the recent deep learning techniques greatly boost the performance of image retrieval. With the powerful deep feature representations, a simple Euclidean distance based search has been able to achieve excellent retrieval performance.

Diffusion process, by exploiting the underlying neighbourhood structure of data, has been shown as an effective mechanism to improve image retrieval [10, 33]. Through propagating affinity information on this structure, diffusion process can more accurately evaluate the similarity between images, showing robustness to background clutter, partial occlusion, and the variation on scale or illumination. Its effectiveness has been shown not only via traditional SIFT features [10, 30], but also with the recent deep features [13]. The latter has exhibited the state-of-the-art performance on benchmark datasets. A particular attractive property of diffusion process is that it improves image retrieval in an unsupervised manner. A more detailed introduction on diffusion process can be found in Section 2.1.

Nevertheless, for image retrieval, diffusion process is more sophisticated than a Euclidean search. It needs to store a large neighbourhood graph whose size increases linearly (or even quadratically) with the size of image database. For a given query, diffusion needs to be performed in an online manner to evaluate the similarity of the query to the images in a database. These not only consume a large amount of memory, but also delay the response of retrieval. In short, although diffusion process brings better image similarity, to benefit from it has to pay the price on computational cost, real-time performance, and search complexity.

This paper aims to improve the above situation to make diffusion-based image retrieval more efficient and practical. Above all, we interpret diffusion process as performing a kernel-induced implicit mapping on the input feature representation of each image. It produces a more advanced feature representation upon which the simple Euclidean distance becomes effective in evaluating the similarity of images. This interpretation motivates us to treat diffusion process as a "black box," and instead of precisely modelling the underlying physical process of diffusion, we explicitly learn such a mapping from the result of diffusion process. In doing so, we will be able to avoid performing online diffusion but retain its positive effect, and enjoy the nice properties of Euclidean search such as simplicity, low computational cost, and the access to many data structures and algorithms.

The recent deep neural networks, characterised by the well-proven capacity in modelling complex functional mappings, provide us the instrumental tool. To realise the above idea, we propose to formulate the modelling of diffusion process as a deep metric learning problem. Given an image database, we first extract feature representations for all images with a pre-trained deep network. Diffusion process is then performed offline, once only, to evaluate the similarities among these images. According to these similarities, image triplets are generated to fine-tune the above deep network, making it learn the implicit kernel-induced mapping and therefore "assimilate" the effect of diffusion process. This fine-tuned network is then used to re-extract feature representations of all images in the database. Once a query (could be out of the database) is given, its representation will be extracted with the same fine-tuned network, and all retrieval in the sequel will purely be performed with Euclidean distance on this new feature representation. Note that the proposed approach does not require any image label information or external datasets for training, and is therefore unsupervised.

In the recent literature, several pieces of work have been aware of the aforementioned issue and made efforts to resolve it. The following two are particularly relevant to our work. The authors in [14] perform an offline low-rank spectral decomposition of the affinity matrix in diffusion process, which helps to realise online diffusion with Euclidean and dot product search. Another work [15] shares an even similar spirit as ours[1] but has a different focus. It utilises the change of k-nearest neighbourhoods before and after diffusion process to mine hard training examples for deep metric learning. In contrast, we focus more explicitly on

[1] We would like to clarify that the work in our paper has been independently developed since 2017.

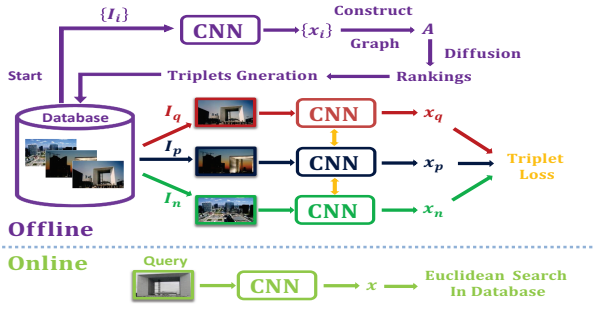

Figure 1: The proposed framework consists of four components for offline training: 1) original features extracted for the images in a database with a convolutional neural network (CNN); 2) constructing the neighbourhood graph with the extracted features and performing diffusion with the graph to obtain image similarities; 3) image triplet generation based on the rankings obtained with the image similarities; and 4) training a deep metric CNN network with the generated triplets. Specifically, the three coloured CNNs at the centre of the figure will be trained with stochastic gradient descent. The yellow arrows connecting them mean that the weights are shared across the three CNNs. After the training process, the features for all images in the database are re-extracted with the newly trained CNN network. For online retrieval, when a query is submitted, extract the features of this query with the same trained CNN network and simply perform a Euclidean search over the database.

modelling the diffusion process by interpreting it as a kernel mapping. Thanks to this different perspective, we have several interesting findings on the effectiveness of this direct modelling (e.g., it could even achieve better performance than the original diffusion-based retrieval), its database-specific characteristic, and its generalisation and robustness with respect to the insertion of new images to a given database.

The contributions of this work are summarised as follows:

1) By taking advantage of the powerful modelling capability of deep neural networks, we propose to model the highly nonlinear diffusion process to generate explicit, better feature representation for image retrieval. It retains the positive effect of diffusion process but avoids online diffusion, significantly reducing computational cost and search complexity.

2) This work indicates an interesting unsupervised learning framework to bootstrap image retrieval, which exploits the underlying structure information of images in a database and converts it to better feature representations for Euclidean search. Moreover, better retrieval could be attained when this bootstrapping process is conducted with more iterations.

3) Experimental study shows multiple appealing properties of the proposed approach. In particular, it could even outperform diffusion process for image retrieval, although our original goal is merely to simulate the effect of diffusion process. This is significant and inspiring, and better justifies the value of the proposed approach.

2 The proposed approach

As pointed out in Section 1, although diffusion process can effectively improve image retrieval, it also brings a number of issues. In detail, this work identifies the following ones: i) large memory cost to store the neighbourhood graph; ii) prolonged retrieval time; iii) special treatment to handle a query out of a given image database; iv) having to perform query-

specific diffusion; and v) having to update the neighbourhood graph when new images are inserted into a database. All of these issues, more or less, significantly affect image retrieval in practice. Our idea is to view diffusion process as performing an unknown, implicit, highly nonlinear mapping from the input feature space to another feature space in which a Euclidean-based measure can align well with the image similarities obtained by diffusion process. The framework of our method is shown in Fig. 1.

2.1 A mapping view of diffusion process

Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ denote a set of data points (e.g., images) in a vector space $\mathcal{X}$. Diffusion process usually starts from computing an $n \times n$ pairwise affinity matrix $\mathbf{A}$. A weighted undirected graph is then constructed, with each node corresponding to a data point and each edge corresponding to the pairwise affinity of two linked points. We now show that diffusion process can be interpreted as performing an implicit, nonlinear kernel mapping.

As surveyed in [10], although many variants of diffusion process have been developed in the literature, they can be well categorised and summarised according to three factors, i.e., the initialisation matrix $\mathbf{W}_0$, the transition matrix $\mathbf{T}$, and the update scheme. The update scheme in [30] gives the best image retrieval performance, and it is expressed as

$$\mathbf{W}_{t+1} = \mathbf{T}\mathbf{W}_t\mathbf{T}^\top, \tag{1}$$

where $\top$ denotes matrix transpose. Note that most of the methods surveyed in [10] set the initialisation $\mathbf{W}_0$ as the affinity matrix $\mathbf{A}$. In this case, it is not difficult to obtain that

$$\mathbf{W}_{t+1} = \mathbf{T}^{t+1}\mathbf{W}_0(\mathbf{T}^{t+1})^\top = \mathbf{T}^{t+1}\mathbf{A}(\mathbf{T}^{t+1})^\top, \tag{2}$$

where the superscript of $\mathbf{T}$ denotes the order of power. A common way to compute the entries of $\mathbf{A}$ uses a Gaussian RBF kernel, making $\mathbf{A}$ positive definite (PD). Immediately, this makes $\mathbf{W}_{t+1}$ a PD matrix and satisfy the Mercer's condition [6]. So, we can interpret $\mathbf{W}_{t+1}$ as a kernel matrix. In particular, the kernel function between points $\mathbf{x}_i$ and $\mathbf{x}_j$ can be written as

$$\kappa(\mathbf{x}_i, \mathbf{x}_j|\mathbf{A}) \triangleq \mathbf{W}_{t+1}(i,j) = \mathbf{T}^{t+1}(i,:)\mathbf{A}(\mathbf{T}^{t+1}(j,:))^\top, \tag{3}$$

where $\mathbf{T}^{t+1}(i,:)$ denotes the ith row of the matrix $\mathbf{T}^{t+1}$. As seen, this is a "context-aware" kernel and its value depends on the whole matrix $\mathbf{A}$ due to the diffusion process. This well shows the characteristic of diffusion process. Therefore, since i) the output of diffusion process, $\mathbf{W}_{t+1}$, can be interpreted as a kernel matrix obtained via the kernel κ and ii) each kernel induces an implicit nonlinear mapping from an input space to another feature space, we can indeed interpret diffusion process as performing an implicit, nonlinear kernel mapping.

Meanwhile, it is worth noting that we use this mapping view to primarily illustrate the idea behind our method, that is, showing what the deep neural network is essentially modelling. In practice, our method requires neither the positive definiteness of $\mathbf{W}_{t+1}$ nor the existence of a kernel function like $\kappa(\mathbf{x}_i, \mathbf{x}_j|\mathbf{A})$. What we need will just be the ranking information of images, from which we can generate image triplets to train the network. This requirement allows our method to readily work with any diffusion process employed in the literature of image retrieval.

2.2 A deep metric learning approach

The challenge of learning the aforementioned implicit nonlinear mapping via deep neural networks lies at how to train the deep neural networks. We can certainly train the network to

produce feature representations such that their inner products best approximate the obtained affinity values in $\mathbf{W}_{t+1}$. Nevertheless, considering that we are dealing with image retrieval and many diffusion processes used in the literature output ranking scores instead of affinity values, we formulate our idea as a common deep metric learning problem, that is, a deep neural network is trained with a set of triplets of the images chosen from a database. Each triplet is composed of one anchor image, one closer image and one farther image.[2] Being closer or farther from an anchor image is defined according to the ranking scores produced by diffusion process. During training, we enforce that for a given query, its distance to the closer image should be smaller than the distance to the farther one by a margin. In the following part, three key issues on training the deep triplet network are elaborated, including 1) network structure, 2) triplet generation, and 3) triplet loss function.

Network structure. As illustrated in Fig. 1, the deep triplet network consists of three CNNs, with all layers shared. They accept the anchor, closer, and farther images as the input, respectively. We adopt the residual network architectures [12] for the CNN due to its outstanding performance demonstrated in the recent literature. The triplet loss function is applied to the features output by the three CNNs. The weights of these three CNNs will be learned with the stochastic gradient descent technique.

Triplets generation. The training of deep triplet network relies on the generation of high quality image triplets. Generating triplets by randomly sampling from the images in a database can hardly provide useful information to benefit the training. In this work, we take a locally constrained triplet generation method. Specifically, given an anchor image I_a, its k-nearest neighbouring images are identified based on the ranking scores obtained by the diffusion process, and they are collectively denoted by a set $\mathcal{N}_k(I_a)$. Two images are then randomly sampled from the set $\mathcal{N}_k(I_q)$. According to their ranking positions with respect to I_a, we regard them as the closer image I_c and the farther image I_f, respectively, to form a triple (I_a, I_c, I_f). We denote all the generated triplets collectively by a set $\mathcal{S}$.

Triplet loss function. We use the following triplet loss which has a soft margin

$$L = \sum_{(I_a,I_c,I_f)\in\mathcal{S}} \left[d(I_a,I_c) - d(I_a,I_f) + \frac{|r_f - r_c|}{k} m_0 \right]_+ , \tag{4}$$

where r_c and r_f denote the ranking positions of I_c and I_f with respect to I_a, $[x]_+$ denotes $\max(x,0)$, and $d(I,J)$ is the Euclidean distance between images I and J based on the features output by the three CNNs. k is the size of neighbourhood used in the triplet generation step and m_0 is a constant as the basic margin. The coefficient $\frac{|r_f - r_c|}{k}$ is our slight modification of the commonly used triplet loss function. In doing so, the magnitude of this soft margin can therefore adapt to the ranking difference between the closer and farther samples, and we find that this is helpful for the network to learn.

2.3 A bootstrapping framework for image retrieval

Built upon the above deep metric learning approach to modelling the diffusion process, we propose an unsupervised bootstrapping framework for image retrieval as follows. It could iterate between performing diffusion process and learning better feature representations to maximise the improvement on retrieval performance.

[2]Note that different from existing deep metric learning methods, we do not access any labelled data. Therefore, we use "closer" and "farther" (instead of positive and negative) to be more precise.

1. Given an image database, extract feature representations of (part or all of) the images, denoted by $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, in the database with a pre-trained deep neural network;

2. Construct the affinity matrix $\mathbf{A}$ with the extracted features. Perform diffusion process with $\mathbf{A}$ (or any of its variants) to obtain the scores on image similarity;

3. With the scores, identify the k-nearest neighbourhood $\mathcal{N}_k(I_a)$ by viewing each of these images as an anchor image I_a. Generate image triplets based on $\mathcal{N}_k(I_a)$;

4. Train the deep triplet network with the triplets. Re-extract features for all the images in the database with the newly trained network; **Go to step** 2 if the maximum number of iterations is yet reached, and go to step 5 otherwise;

5. When a query image (could be out of the database) is given, extract its features with the latest trained network, and perform retrieval with a simple Euclidean distance.

3 Experimental Result

3.1 Experimental setup

Dataset. The proposed approach is tested on the benchmark datasets of Oxford5k [19], Pairs6k [20], Instre [29], Sculpture [1], Oxford105k, and Pairs106k. The last two are obtained by adding 100k distractor images collected from Flicker. For each dataset, there is no overlapping between the images in the database and query images. To ensure an objective evaluation, we only use the images in the database to do diffusion and train the network, and reserve the query images exclusively to test retrieval performance. This is consistent with the protocol commonly adopted in the literature. In addition, we evaluate INSTRE by following the recent work [13] and use standard evaluation protocol for all the other datasets. Mean average precision (mAP) is used to measure retrieval performance in all experiments.

Network training. The CNN of ResNet101 pre-trained with ImageNet [12] is referred to as "ResNet101-ImageNet" and used in Tasks 1 and 2 defined in the last paragraph of Section 3.1. During training, all images are resized with the longer side having 600 pixels. Stochastic gradient descent technique is used. The learning rate is initialised as 0.01 and gradually attenuates during the training process. The coefficients for weight decay and the momentum are 0.0001 and 0.9. The batch size is 40, and the training process usually takes 1000 epochs to converge. The margin m_0 in Eq.(4) is empirically set as 0.1, and the number of nearest neighbours, k, to generate triplets is set as 300. Note that to clearly show the basic performance of the proposed method, we only train the deep metric network with the diffusion process *once* in all experiments, except the part particularly investigating the case of multiple iterations.

Retrieval setting and baseline. R-MAC [28] feature representation is used to describe each image. The LCDP [30] update scheme (in Eq. (1)) is adopted to perform diffusion. We compare our method with the following two baselines: 1) R-MAC+E and 2) R-MAC+D, where "E" and "D" mean that Euclidean distance search and diffusion process are used for retrieval, respectively. By the comparison, we want to verify whether our approach can effectively achieve or even improve over the retrieval performance obtained via diffusion process. To ensure fair comparison, we implement the baselines and our methods under the same experimental setting.

Method (mAP)	Oxford5k	Paris6k	Oxford105k	Paris106k	INSTRE	Sculpture
R-MAC+E (global)	58.5	73.3	57.3	67.4	37.7	51.0
R-MAC+D (global)	63.1	83.5	62.0	77.0	53.0	61.6
Proposed (global)	**63.2**	**89.6**	**62.4**	**82.6**	**54.5**	**75.5**

Table 1: Comparison with two baseline methods under a global image representation, where "E" denotes the Euclidean distance based search while "D" denotes diffusion process. The "ResNet101-ImageNet" network is used in this experiment.

Experimental tasks. There are five tasks: 1) Compare the proposed method with R-MAC+E and R-MAC+D, where each image is represented by a *global* representation (i.e., R-MAC); 2) Compare it with the state-of-the-art retrieval methods where each image is represented by a set of *regional* representations; 3) Compare it with various recent image retrieval methods to give a whole picture; 4) Compare it with diffusion-based image retrieval in terms of time and memory cost in online retrieval; and 5) Investigating important properties of the proposed method, such as its generalisation and robustness to image insertion and the help of multiple iteration training. The results are reported in order in the next section.

3.2 Result and discussion

Task 1. Table 1 compares our method with R-MAC+E and R-MAC+D under the global image representation. As shown, by conducting diffusion process, R-MAC+D consistently achieves higher retrieval performance (5 to 16 percentage points) than R-MAC+E that uses Euclidean distance based search. By assimilating the effect of diffusion process via deep metric learning, our method, still using Euclidean search, not only achieves comparable performance as R-MAC+D on Oxford5k, Oxford105k, and INSTRE, but also outperforms it on Paris6k, Paris106k, and Sculpture. In particular, our method brings up to 14 percentage points of mAP increase on Sculpture (75.5 (ours) vs 61.6). This result is significant. It shows that our method is indeed effective in assimilating the effect of diffusion process and converting it to enhanced feature representation. Furthermore, it is surprising to observe these large improvements on three datasets. We attribute such improvement to the *wholly* manner of our method in approximating diffusion process. That is, our deep metric learning is performed upon a large number of image triplets generated from the whole database. This provides the network with a "global" view about the similarity of these images, and therefore may help the network to produce overall better feature representations. As for R-MAC+D, a specific diffusion process is performed for a given query, and this process is initialised by or dependent on this query. Such a "local" view may limit its overall retrieval accuracy. This interesting issue will be further explored in our future work.

Task 2. Region-based image retrieval methods have recently shown excellent performance by representing an image as a set of regional features. The image similarity is usually evaluated by summarising the similarity across image regions. In this case, diffusion process is performed on the graph constructed upon these regional deep features. In this task, we focus on Paris6k to compare with two state-of-the-art methods of this kind: the cross-region matching method [23] and the regional diffusion method [13] (re-implemented by us with the network "ResNet101-ImageNet"). They obtain the mAP of 84.4 and 91.8, respectively. The better performance of the regional diffusion method is due to its use of diffusion process to evaluate the similarity of image regions. Our method achieves an mAP of **93.8**, which fur-

ther improves the regional diffusion method by two percentage points. This again indicates the effectiveness and advantage of our method in approximating diffusion process.

Task 3. To give a whole picture about the performance of the proposed method, Table 2 compares it with the image retrieval methods developed in the recent literature. To be consistent with the state-of-the-art methods, we use the CNN structure provided in [11], which fine-tunes ResNet101 with an additional landmark dataset (called "ResNet101-Landmarks" in this work), for our deep metric learning approach. We categorise all the retrieval methods in comparison into two groups: 1) the methods only applying Euclidean distance based search with a global feature representation, as shown in the upper part of the table; and 2) the methods applying diffusion process or post-processing steps (such as query expansion, matching, and verification), which appear in the lower part of the table. The first group of methods enjoys higher computational efficiency in retrieval, while the second group of methods generally achieves higher retrieval performance. For our method, which only conducts Euclidean search to retrieve images, it can well outperform most of the methods in the first group and achieve quite competitive performance to those in the second group that has more sophisticated online retrieval mechanisms. This result shows that by assimilating the effect of diffusion process with new features, our method can enjoy both high computational efficiency and high retrieval accuracy for online retrieval. In addition, it is worth noting that the results in Tables 1 and 2 are not directly comparable. The proposed method in Table 1 is implemented based on the "ResNet101-ImageNet" network, while in Table 2 it is implemented based on the network "ResNet101-Landmarks" to be consistent with the state-of-the-art methods.

Task 4. To show computational efficiency, we compare the time and memory cost of the proposed method with diffusion-based image retrieval in performing online retrieval on three datasets. The experiment is conducted with Matlab2017a on a desktop computer of Intel@core i7-4720 2.60GHz CPU and the result is reported in Table 3 for retrieval with the global and regional representations, respectively. As expected, our method is consistently faster and can shorten online retrieval up to 10 times for a single query. Furthermore, due to the use of Euclidean distance, our method can readily be sped up by utilising off-the-shelf data structure and algorithms. Also, because it does not need to store the neighbourhood graph, it incurs no extra memory usage in this aspect.

Task 5. *1) Image insertion.* One drawback of diffusion-based image retrieval lies at that it needs to update its neighbourhood graph when new images are inserted into a database. This experiment investigates the robustness and the generalisation capability of the proposed method in this situation. Now we only use *part* of the images (n_0) in a database to build the graph, conduct diffusion, and generate triplets for training. However, when a query is submitted, the retrieval will be performed on the *whole* database. This simulates the case that all the remaining images (i.e., other than these n_0 ones) are newly inserted after the proposed method is trained. As previous, the proposed method uses the learned feature representations, respectively, to perform retrieval. The result is plotted in Fig. 2. The horizontal axis is the ratio of images taken from a database used for performing diffusion and training the proposed method, while the vertical axis shows the mAP value. The three dotted lines indicate the baseline performance for Oxford105k, Paris6k, and INSTRE, respectively, when diffusion process is not used. The three solid lines show the corresponding performance of the proposed method. As seen, its performance steadily improves with the increasing ratio and quickly approach the level when all images (i.e., ratio = 1.0) in a database are used for diffusion process and training the proposed method. For Paris6k, improvement over the baseline can be observed even when the ratio is as low as 0.1. As for Oxford105k and INSTRE,

Method	Dim.	Oxford5k	Paris6k	Oxford105k	Paris106k	INSTRE
Global image representation with Euclidean search						
[16]	128	43.3	-	35.3	-	-
[5]	128	55.7	-	52.3	-	-
[31]	128	59.3	59.0	-	-	-
[9]	256	53.1	-	50.1	-	-
[28]	512	66.9	83.0	61.6	75.7	-
[17]	512	68.2	79.7	63.3	71	-
[13]*	512	77.7	84.1	70.1	76.8	47.7
[15]	512	78.2	85.1	72.6	78.0	57.7
[22]	512	79.7	83.8	73.9	76.4	-
[16]	1024	56.0	-	50.2	-	-
[28]	2048	69.4	85.2	63.7	77.8	-
[11]	2048	86.1	94.5	82.8	90.6	-
[13]⋆	2048	83.9	93.8	80.8	89.9	62.6
[9]	4096	71.6	79.7	-	-	-
Global image representation + diffusion / query expansion / matching / verification						
[17]	-	72.2	85.5	67.8	79.7	-
[24]	-	75.2	74.1	72.9	-	-
[21]	-	81.4	80.3	76.7	-	-
[9]	-	82.7	80.5	76.7	71.0	-
[8]	-	84.3	83.4	80.2	-	-
[18]	-	84.9	82.4	79.5	77.3	-
[27]	-	86.9	85.1	85.3	-	-
[26]	-	89.4	82.8	84.0	-	-
[28]	512	77.3	86.5	73.2	79.8	-
[3]	512	79.0	85.1	-	-	-
[22]	512	84.5	86.4	80.4	79.7	-
[13]*	512	85.4	88.4	79.7	83.5	57.3
[28]	2048	78.9	89.7	75.5	85.3	-
[13]	2048	87.1	**96.5**	87.4	**95.4**	**80.5**
[11]	2048	**90.6**	96.0	**89.4**	93.2	-
[13]⋆	2048	**89.6**	95.3	**88.3**	92.7	70.5
[14]	2048	**87.5**	**96.4**	**87.9**	**95.3**	**80.5**
Our global image representation (by modelling diffusion process) + Euclidean search						
Proposed	2048	85.4	**96.3**	85.1	**94.7**	**71.7**

Table 2: Comparison with the state-of-the-art image retrieval methods. The result shows that the proposed method effectively assimilates the effect of diffusion process to generate better feature representations, upon which it achieves very competitive retrieval performance with simple Euclidean distance. [13]* and [13]⋆ are the results reported by [13] as the re-implementation of [22] and [11] with ResNet101 fine-tuned on an external landmark dataset. Top three values per column are in bold.

clear improvement can be obtained once the ratio exceeds 0.3. These results show that our method generalises well with respect to the insertion of new images. *2) Iterative training by re-applying diffusion.* Our bootstrapping framework for image retrieval supports iterative training. We can alternate between learning new feature representation and performing diffusion process with this learned representation. Tested on INSTRE, our method does obtain better retrieval by extra one or two iterations, and the mAP result is 71.7 at the first iteration,

	Global feature representation			Regional feature representation		
Dataset	Oxford5k	INSTRE	Oxford105k	Oxford5k	INSTRE	Oxford105k
Diff. based	0.020/0.01	0.100/0.03	2.90/0.11	0.6/0.1	2.9/0.6	13.0/2.1
Proposed	0.002/N.A.	0.011/N.A.	0.03/N.A.	0.1/N.A.	0.4/N.A.	1.43/N.A.

Table 3: Comparison of average time / memory usage (Second / GB) in online retrieval. The dimensions of image feature representation are 2048. Time cost is averaged over all of the queries.

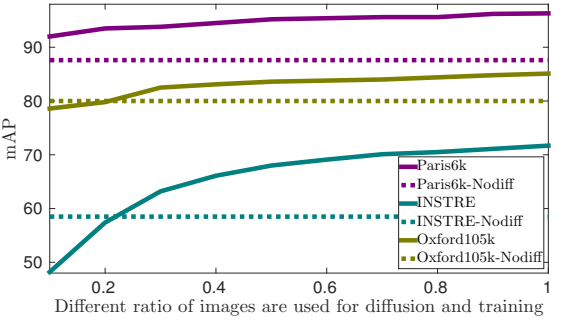

Figure 2: Retrieval performance of the proposed method when different ratio of images are taken from a database for conducting diffusion process and training the proposed method. This experiment investigates its generalisation capability with respect to new image insertions. The three dotted lines indicate the baselines when diffusion process is not used. The solid lines show the corresponding performance of the proposed method. The "ResNet101-Landmarks" network is used in this experiment.

74.2 at the second, and 74.5 at the third, all higher than the baseline of 70.8. This result initially shows the effectiveness of the proposed bootstrapping framework and its potential will be further explored in the future work.

4 Conclusion

Utilising the modelling capability of deep neural networks, this work assimilates the effect of diffusion process into new feature representation, achieving similar or better retrieval with simple Euclidean search. Also, it gives an unsupervised framework to bootstrap image retrieval by exploiting the manifold structure of the images in a database. It effectively improves retrieval without the aid of additional labels or external datasets. Experimental study on benchmark datasets demonstrates its effectiveness and advantages.

This work takes a database-specific approach by assuming the access to a database in advance. How to generalise it to unseen databases will be an interesting issue to explore in the future work. It is believed that training it with a sufficiently large and generic database will enhance its generalisation capability to some extent. Also, due to its database-specific characteristic, the feature representation learned by the proposed approach on one database may not be effectively applied to another database of a significantly different nature. This issue will also be addressed in the future work. Integrating the proposed approach with domain adaptation and transfer learning techniques could be a potential solution.

References

[1] Relja Arandjelović and Andrew Zisserman. Smooth object retrieval using a bag of boundaries. In *Proc. ICCV*, pages 375–382, 2011.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, pages 5297–5307, 2016.

[3] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE PAMI*, 38(9):1790–1802, 2016.

[4] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proc. ICCV*, pages 1269–1277, 2015.

[5] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Proc. ECCV*, pages 584–599, 2014.

[6] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, June 1998. ISSN 1384-5810.

[7] Ondrej Chum, Andrej Mikulík, Michal Perdoch, and Jiri Matas. Total recall II: Query expansion revisited. In *Proc. CVPR*, pages 889–896, 2011.

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Ze Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60, 2008.

[9] Cheng Deng, Rongrong Ji, Wei Liu, Dacheng Tao, and Xinbo Gao. Visual reranking through weakly supervised multi-graph learning. In *Proc. ICCV*, pages 2600–2607, 2013.

[10] Michael Donoser and Horst Bischof. Diffusion processes for retrieval revisited. In *Proc. CVPR*, pages 1320–1327, 2013.

[11] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 124(2):237–254, 2017.

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

[13] Ahmet Iscen, Giorgos Tolias, Yannis S Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *Proc. CVPR*, pages 926–935, 2017.

[14] Ahmet Iscen, Yannis S Avrithis, Giorgos Tolias, Teddy Furon, and Ondrej Chum. Fast spectral ranking for similarity search. In *Proc. CVPR*, 2018.

[15] Ahmet Iscen, Giorgos Tolias, Yannis S Avrithis, and Ondrej Chum. Mining on manifolds: Metric learning without labels. In *Proc. CVPR*, 2018.

[16] Hervé Jégou and Andrew Zisserman. Triangulation embedding and democratic aggregation for image search. In *Proc. CVPR*, pages 3310–3317, 2014.

[17] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Proc. ECCV*, pages 685–701, 2016.

[18] Andrej Mikulik, Michal Perdoch, Ondřej Chum, and Jiří Matas. Learning vocabularies over a fine quantization. *IJCV*, 103(1):163–175, 2013.

[19] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, pages 1–8, 2007.

[20] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, pages 1–8, 2008.

[21] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Proc. CVPR*, pages 777–784, 2011.

[22] Filip Radenović, Giorgos Tolias, and Ondřej Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *Proc. ECCV*, pages 3–20, 2016.

[23] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *Proc. CVPR Workshops*, pages 512–519, 2014.

[24] Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Spatially-constrained similarity measurefor large-scale object retrieval. *IEEE PAMI*, 36(6):1229–1241, 2014.

[25] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh C. Jain. Content-based image retrieval at the end of the early years. *IEEE PAMI*, 22(12):1349–1380, 2000.

[26] Giorgos Tolias and Hervé Jégou. Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recognition*, 47(10):3466–3476, 2014.

[27] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: Aggregation across single and multiple images. *IJCV*, 116(3):247–261, 2016.

[28] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *Proc. ICLR*, 2016.

[29] Shuang Wang and Shuqiang Jiang. Instre: A new benchmark for instance-level object retrieval and recognition. *ACM TOMM*, 11(3):37, 2015.

[30] Xingwei Yang, Suzan Koknar-Tezel, and Longin Jan Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *Proc. CVPR*, pages 357–364, 2009.

[31] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. Exploiting local features from deep networks for image retrieval. In *Proc. CVPR Workshops*, pages 53–61, 2015.

[32] Liang Zheng, Yi Yang, and Qi Tian. SIFT meets CNN: A decade survey of instance retrieval. *IEEE PAMI*, 40(5):1224–1244, 2018.

[33] Denny Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *NIPS*, pages 169–176, 2004.