scientific reports

OPEN

Learning optimal image representations through noise injection for fine-grained search

Vidit Kumar[1], Vikas Tripathi[1], Bhaskar Pant[1], Manoj Diwakar[1], Prabhishek Singh[2] & Anchit Bijalwan[3]✉

In recent years, fine-grained image search has been an area of interest within the computer vision community. Many current works follow deep feature learning paradigms, which generally exploit the pre-trained convolutional layer's activations as representations and learn a low-dimensional embedding. This embedding is usually learned by defining loss functions based on local structure like triplet loss. However, triplet loss requires an expensive sampling strategy. In addition, softmax-based loss (when the problem is treated as a classification task) performs faster than triplet loss but suffers from early saturation. To this end, a novel approach is proposed to enhance fine-grained representation learning by incorporating noise injection in both input and features. At the input, input image is made noised and the goal is set to reduce the distance between the L2 normalized features of input image and its noisy version in the embedding space, relative to other instances. Concurrently, noise injection in the features acts as regularization, facilitating the acquisition of generalized features and mitigating model overfitting. The proposed approach is tested on three public datasets: Oxford flower-17, Cub-200-2011 and Cars-196, and achieves better retrieval results than other existing methods. In addition, we also tested our approach in the Zero-Shot setting and got favorable results compared to the prior methods on Cars-196 and Cub-200-2011.

Keywords Fine-grained image retrieval, Image representation, Feature learning, Noise injection, Zero-shot learning

Image retrieval has been studied for decades, yielded significant results, and is still a challenging topic. A challenge is to obtaining visually related images to the query sample by analyzing its visual characteristics either by low-level semantics (like shape, texture, color) or by higher semantics (like bag of visual words, neural codes)[1]. Prior (Content based image retrieval) CBIR's methods work well for databases of large inter-class variance as compared to databases of less inter-class variance (see Fig. 1). However, real-life scenarios require fine-grained search, that is, to locate images that correspond to the exact query's sub-category. For instance, when a user queries an image (say bike or flower image), the user needs to access/retrieve images in the same fine-level category as a query (i.e., images correspond to the same model of bike or same flower species)[2]. In such a setting, retrieval becomes a complex and challenging task because it is arduous to distinguish between various models of cars or bikes, or various species of flowers, or different breeds of dogs. The reason for this is that they share visual appearances at the global level, which can only be distinguished by focusing on the critical parts of the object, such as the bird's feature texture, the dog's body color, and the shape of the bike's headlight, etc. Therefore, the major challenge of this problem is to produce strong representations that can capture these subtle details and reduce differences between nearly identical categories. Fine-grained search can be used for various purposes, including but not limited to surveillance, evaluation of climate change, intelligent retail, monitoring of biodiversity and ecosystems, intelligent transportation, etc.

Learning effective descriptors plays an important role in the fine-grained image retrieval (FGIR) domain. When good features are exploited, a retrieval algorithm allows similar images to be placed in beginning of a ranked list and dissimilar ones at the end. Since[2], FGIR has drawn a growing research focus in computer vision society. Despite recent progress, FGIR is still an open problem for commercial and cataloging applications. With the recent developments in deep learning[3–5], the deep learning methods built upon (convolutional neural network) CNN features have become the mainstream of fine-grained search. However, these features are learned

[1]Department of CSE, Graphic Era Deemed to be University, Dehradun, India. [2]School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India. [3]Faculty of Electrical and Computer Engineering, Arba Minch University, Arba Minch, Ethiopia. ✉email: anchit.bijalwan@amu.edu.et

Fig. 1. Comparison of image database. {Dataset Source: corel_images [https://www.kaggle.com/datasets/elkamel/corel-images], Oxford Flowers-17[14]; https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html and Cars-196[15]; https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset?datasetId=30084&sortBy=dateCreated&select=cars_test}.

from the coarse domain; direct exploitation is not feasible since they cannot capture the fine details of the object. Instead, low dimensional features are learned on top of CNN features using the so-called deep metric learning (DML) approach, which aims to learn the low dimensional metric space (or embedding space) of embeddings where similar things are close and dissimilar are distant. Lots of work has been done in this area using contrastive loss[6], triplet loss[7,8], and quadruplet loss[9,10]. Most of them follow triplet loss. However, triplet loss is based on mining strategies[7,8,11–13] to make it fast convergence, which requires extra computations. On the other hand, softmax is generally faster to converge compared to triplet loss but suffers in early saturation, which converges to some worse local minima. Furthermore, learning embeddings from larger networks poses overfitting to small datasets. In this paper, we tend to overcome these issues by proposing a noise-invariant feature learning approach. In this approach, the model is trained using auxiliary induced noise injected at two positions: at input layer and final layer of the deep network. By introducing noise at the input layer, the model learns noise-invariant features by maximizing the similarity between an image instance and its corresponding noisy version. Meanwhile, the noise added at the final layer, in conjunction with the softmax cross-entropy loss function, serves as a form of regularization by generating augmented features within the embedding space. In the former case, we employ a contrastive learning approach, where positives are formed by injecting noise into images, while other samples serve as negatives. In the latter case, the induced noise prevents softmax from suffering early saturation and allows for the continued propagation of gradients computed on noise-augmented features, thereby helping to reduce overfitting on small datasets.

The following are our key contributions:

1) We propose a Noise-invariant feature embedding learning method by optimizing it using softmax. This minimizes the costly sampling process in training DML, which is the main limitation of triplet loss. This also alleviates the problem of early saturation of softmax-based learning.
2) This is done by adding noise into both the input layer and the last layer of the deep network during the training process. The primary objective, grounded in contrastive learning, aims to maximize the similarity between an image instance and its corresponding noisy version. The secondary objective, relying on softmax cross-entropy, addresses augmented features generated within the embedding space, serving as a form of regularization.
3) Analysis on three fine-grained datasets illustrates that our approach achieves better results than state-of-the-art.

The rest of the paper is structured as follows: existing related works are explored in Section "Related Work". The proposed approach is detailed in Section "Methodology". Section "Experiments" discusses the experimental settings and analyzes the outcome results. Section "Conclusion" concludes the paper.

Related Work

Following the success of CNN[3], deep learning techniques also led to research in image retrieval[1]. For instance, Babenko et al.[16] employed a pretrained CNN, fine-tuned it on the target images, and used its responses for image representation and retrieval. In[17], a feature aggregation method was presented that exploits sum pooling on deep features to generate compact descriptors. Further, Mohedano et al.[18] exploit bag-of-Word model with CNN features, whereas in[19], CNN features with VLAD are exploited for image search. Reference[20] employed sum pooling in their aggregated method over weighted convolutional features across channels and spatial locations. In addition, Yang et al.[21] presented an image retrieval technique based on Cross Batch Reference based feature learning strategy. Tolias et al.[22] presented an approach that generates compact features by encoding multiple locations with convolutional layer's activations. Shakarami et al.[23] present a fusion-based descriptor for image retrieval, which includes LBP, HOG, and CNN features. Although these methods work well for coarse levels, fine-grained localization is required as an initial step for fine-grained images. Using the deep learning paradigm some efforts have also been made for fine-grained image tasks. For instance, reference[24] utilized convolutional kernels for both object's parts selection and representation. Watkins et al.[25] suggested a two-stage learning scheme (localization learning followed by classification using detected location) for fine-grain classification by exploring resnet architectures. Zhou et al.[26] explore label hierarchy using rich relationships through bipartite-graph with

VGG-net[4] for fine-grained classification. In[27], authors deployed pre-trained VGG-16[4] for object localization and selected its deep descriptors by removing noise or background. Zheng et al.[28] suggested the centralized ranking loss and trained the CNN with weakly supervised object localization. Then they employed a CNN response map with the contours to precisely extract the features. Kumar et al.[29] explored ResNet18[5] for the FGIR task, where they fine-tuned it on the target dataset and used its activations for retrieval. Yingying et al.[30] proposed relation based convolutional descriptor that encodes local subtle features for FGIR. Further, some efforts are made in the direction of learning embedding. For instance,[6] used the pair-wise loss and[7] used the triplet loss for learning image embedding with CNN as a backbone. Subsequently, Song et al.[31] exploited every pair in the minibatch to obtain hard negatives. Sohn et al.[32] extends the triplet loss[7,8] into N-pairs loss, which uses softmax cross-entropy loss on pair-wise similarity values within the batch. Song et al.[33] presented the clustering loss for embedding learning by considering the embedding space's global structure. Huang et al.[10] exploited quadruplet and mines hard examples in end-to-end network with PDDM block for similarity evaluation. Zheng et al.[34] proposed softmax Loss for FGIR with normalize-scale layer. The Ranked List loss[35] accounts for both positive and negative data within a batch, aiming to clearly differentiate between the positive and negative sets. Reinforcement learning based sampling was proposed in[36]. Koth et al.[13] also explored policy-adapted sampling via reinforcement learning for triplet losses. Further, Zheng et al.[37] explore hard negative mining via generative approach. Duan et al.[38] proposed multilevel similarity based metric loss which explore global, local and channel level similarity. Sanakoyeu et al.[39] explored divide and conquer approach in which they iteratively divide the embedding to learn different features.

However, most of these methods rely on sampling strategies that make model training more computationally expensive. In contrast to the above analysis, we implemented a simple strategy for learning fine-grained features via a noise-assisted learning approach which strengthens the feature representation potential of the base network without requiring any sampling strategies.

Methodology

The outline of proposed method is depicted in Fig. 2. First a minibatch of images is randomly sampled and noised. Then pairs of noisy images and natural images are fed to Siamese network and a minibatch of natural images is fed to two standalone networks. The Siamese network is responsible for making features noise-invariant, while other two networks are responsible for learning class discriminating features. All networks are jointly trained with common goal of feature representation learning for fine grained image retrieval.

Consider the training images $\{x_1, x_2, \ldots, x_m\} \in X$ with associated labels $y_i \in Y$ in the minibatch. Let f_p and f_n be the L_2 normalized feature embedding of positive instance x_p and negative instance x_n to instance x_i such that $y_i = y_p; y_i \neq y_n$. These positives and negatives are selected from the minibatch during training. Assume $(\cdot, \odot, \cdot)$ the cosine similarity function with $\odot$ as dot product. To enforce the compactness among same class instances and separateness among different class instances in the embedded space, the class discrimination loss inspired by[40] could be given as:

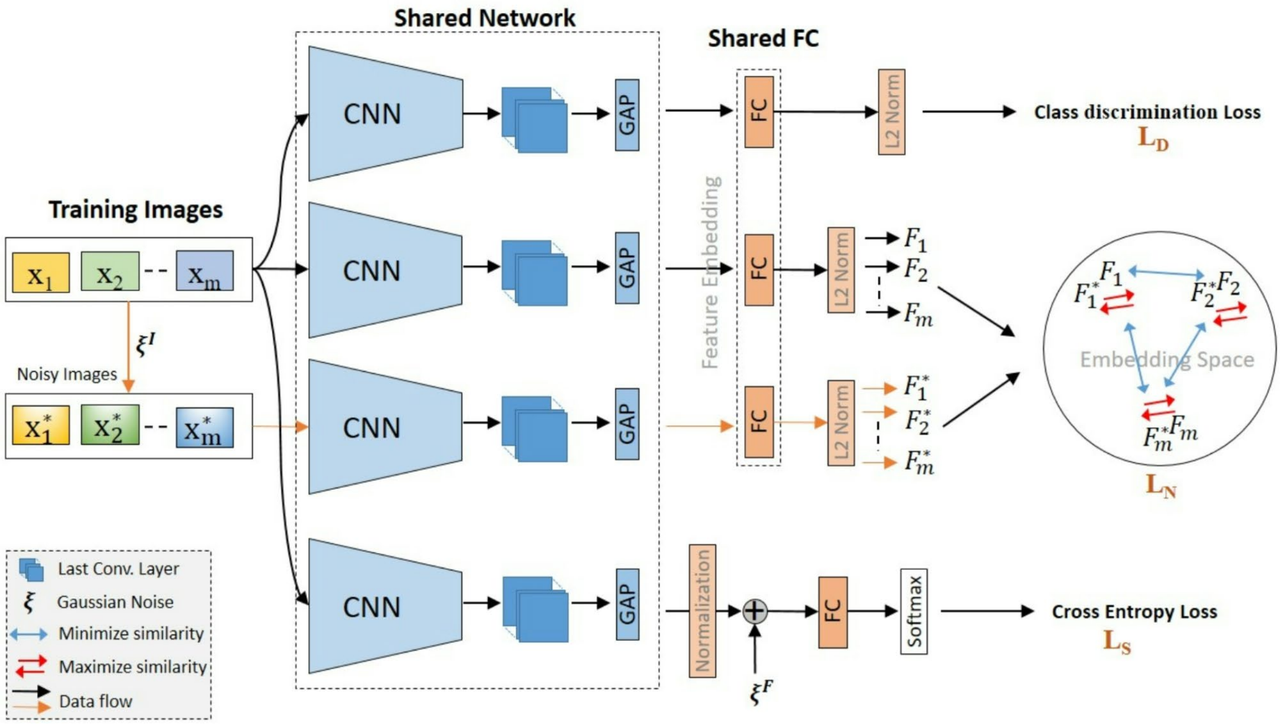

Fig. 2. Proposed Noise-invariant feature learning method.

$$\mathrm{L_D} = \sum_{\mathrm{i}} \frac{1}{|\mathrm{P}(\mathrm{i})|} \sum_{\mathrm{p}\in\mathrm{P}(\mathrm{i})} \log\left(1 + \sum_{\mathrm{n}\in\mathrm{N}(\mathrm{i})} \exp\left(\mathrm{f_i}\odot\mathrm{f_n}/\tau - \mathrm{f_i}\odot\mathrm{f_p}/\tau\right)\right) \tag{1}$$

where, P(i) is set of positive indices to *i*th instance and N(i) is set of negative indices to $\mathrm{i^{th}}$ instance.

Noise-invariant feature learning for FGIR

To improve the feature representation capability for a network, the noise can helps the deep CNN to learn better representations for fine-grained images. The noisy labels used in prior publications[41,42] for feature learning need a large dataset with noisy labels network's training. Instead of using noisy labels, the network is optimized by injecting noise at the input layer and CNN's higher layer. Specifically, for each training iteration, a noise is sampled from zero mean Gaussian distribution, which is injected to the input images as well as in the activations of last layer (output of average pooling layer in our case) of the deep CNN (refer Fig. 2).

Let $\xi_i^I \in N\left(0, \delta_i^2\right)$ be the noise sampled from the zero mean Gaussian distribution. The noise is injected to each sample selected for minibatch as $\tilde{x}_i = x_i + \xi_i^I$. Let $\mathrm{f_i}$ and $\widetilde{\mathrm{f_i}}$ be the $\mathrm{L_2}$ normalized feature embedding of x_i and $\tilde{x}_i$. For all instances $x_i \in X$, the objective is to maximize $\left(\mathrm{f_i}\odot\tilde{\mathrm{f}}_\mathrm{i}\right)$.

Given a Siamese network, we compute the probability of noisy sample $\tilde{x}_i$ being classified as *i*th image as:

$$\mathrm{P}\left(i|\tilde{\mathrm{x}}_i\right) = \frac{\exp\left(\mathrm{f_i}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right)}{\exp\left(\mathrm{f_i}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right) + \sum_{\mathrm{j=1:m,\ j\neq i}}\exp\left(\mathrm{f_j}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right)} \tag{2}$$

The loss[40] associated with (2) is given as:

$$\mathrm{L_N} = -\sum_{\mathrm{i}}\log\frac{\exp\left(\mathrm{f_i}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right)}{\exp\left(\mathrm{f_i}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right) + \sum_{\mathrm{j=1:m,\ j\neq i}}\exp\left(\mathrm{f_j}\odot\tilde{\mathrm{f}}_\mathrm{i}/\tau\right)} \tag{3}$$

The Siamese network in this approach excels at learning embeddings for fine-grained representation by comparing and distinguishing pairs of inputs. In our approach, it is utilized to create a meaningful embedding space that brings similar images closer together. Here, the loss $\mathrm{L_N}$ will take care for compacting the distance between $\left(f_i, \tilde{f}_i\right)$ pairs which means making features noise invariant. It also minimizes $\exp\left(\mathrm{f_j}\odot\widetilde{\mathrm{f_i}}\right)$ for all other instances, making separateness among other instances relative to its clean instance.

We also adopt multi-classification task to further optimize the network, however softmax suffers early saturation due to overfitting to smaller datasets. To overcome this, we inject the gaussian noise to the output of final layer of network (avg. pool in our case), so that each time loss will penalize the noisy feature for predicting low score.

Let $\xi_i^F \in N\left(0, \delta_i^2\right)$ be a noise, Z_i represents the deep CNN's last layer normalized[43] activations for input image i, the noisy response can be deduced as $\tilde{Z}_i = Z_i + \xi_i^F$. Now, with K-way softmax through fully connected layer $FZ = w_z\tilde{Z}_i + b_z$, the probability distribution of a model parameterized by ϕ over m classes is given as:

$$P\left(y_i\,|i\,,\phi\right) = \frac{\exp(FZ_i)}{\sum_K \exp(FZ_j)} \tag{4}$$

With the goal to maximize this probability (4), the loss is to minimize is:

$$L_S = -\frac{1}{|m|}\sum_{n=1}^{|m|}\log P\left(y_i\,|i_n,\phi\right) \tag{5}$$

The total loss is given as:

$$L = L_D + \lambda_1 L_N + \lambda_2 L_S \tag{6}$$

Minimizing L means minimizing all three losses $\mathrm{L_D}$, $\mathrm{L_N}$ and $\mathrm{L_S}$. first Eq. (1) can be reformulated as

$$\mathrm{L_D} = -\sum_{\mathrm{i}}\frac{1}{|\mathrm{P}(\mathrm{i})|}\sum_{\mathrm{p}\in\mathrm{P}(\mathrm{i})}\log\frac{\exp\left(\mathrm{f_i}\odot\mathrm{f_p}/\tau\right)}{\exp\left(\mathrm{f_i}\odot\mathrm{f_p}/\tau\right) + \sum_{\mathrm{n}\in\mathrm{N}(\mathrm{i})}\exp\left(\mathrm{f_i}\odot\mathrm{f_n}/\tau\right)} \tag{7}$$

Now, examining L, minimizing Eq. (7) necessitates maximizing $\exp\left(\mathrm{f_i}\odot\mathrm{f_p}/\tau\right)$ and minimizing $\exp\left(\mathrm{f_i}\odot\mathrm{f_n}/\tau\right)$. Given that features are L2 normalized, maximizing $\exp\left(\mathrm{f_i}\odot\mathrm{f_p}/\tau\right)$ involves maximizing the cosine similarity between $\mathrm{f_i}$ and $\mathrm{f_p}$, forcibly aligning the features of the original sample and its positive counterpart. Similarly, minimizing $\exp\left(\mathrm{f_i}\odot\mathrm{f_n}/\tau\right)$ involves decreasing the cosine similarity between $\mathrm{f_i}$ and $\mathrm{f_n}$, forcibly separating the features of the original sample from its negative counterparts. This results in compactness of similar samples and separateness of dissimilar samples in embedding space. Now looking into $\mathrm{L_N}$, minimizing it necessitates maximizing $\exp\left(f_i \odot \widetilde{f}_i/\tau\right)$ and minimizing $\exp\left(f_j \odot \widetilde{f}_i/\tau\right)$. Maximizing $\exp\left(f_i \odot \widetilde{f}_i/\tau\right)$ compels forcibly aligning the features of the original sample f_i and its noisy counterpart $\widetilde{f}_i$. The outcome is a noise-invariant feature embedding. Similarly, minimizing $\exp\left(f_j \odot \widetilde{f}_i/\tau\right)$ forcibly separating $\widetilde{f}_i$ from the features of other instances f_j. This further ensures separateness of dissimilar samples in embedding space. Last minimizing L_S will further enhance the noise invariant property and class separability.

Overall steps of our approach is summarized in Algorithm 1.

Input: training images set X, initialized $f(\cdot, \theta)$, parameters $\ell, \lambda_1, \lambda_2$

// ℓ is a learning rate, $\lambda_1 = 1$, $\lambda_2 = 1$, δ= 0.1

Output: Optimized model $f(\cdot, \theta)$

for i =1 to max_Iteration **do**

    Sample K classes randomly

        for each of K classes **do**

            Sample K1 images randomly

            Inject noise $\xi_i^I \in \mathcal{N}\left(0, \delta_i^2\right)$ to create K1 noisy images

        end for

    Fed noisy clean image pairs to Siamese network $f(\cdot, \theta)$ and extract L2 normalized acivations

    Fed clean images to other shared network $f(\cdot, \theta)$ and extract acivations

    Compute the loss using (1) and (3) by computing pair-wise cosine similarities

    Compute the loss using (5) over noisy features created by injecting noise $\xi_i^F \in \mathcal{N}\left(0, \delta_i^2\right)$ to normalized final layer's activations

    Compute the total loss L using (6)

    Update the network parameter $\theta = \theta - \ell \dfrac{\partial L}{\partial \theta}$

end for

return optimized $f(\cdot, \theta)$

Algorithm 1. Noise-invariant Feature Learning for FGIRTraining details

We used resnet18 (R18)[5] as a backbone. To make a good start, we initialize the R18's parameters with weights trained on imagenet[48]. The dense layers' weights are initialized as in[5]. The size of embedding is set to 256 and adam with weight decay of 10e-4 is used for network training. The learning rate and mini-batch's size is set to 10e-4 and 64 respectively. We first sample 8 class randomly and then sample 8 instances per class. For each sample, noisy sample is created for siamese network. We exploit the data augmentation operations as follows: after randomly sampling a mini-batch of training images, first it is resized with its shorter side to 256 by preserving the aspect ratio, which maintains the original shape of the object. Then it is crop with size 224 × 224 from random location within the image. Next, it is rotated with degree within the range of (-15, 15) (followed by a center crop to maintain same spatial size). At last, with a 0.5 probability, color augmentation takes place followed by horizontal flipping with 0.5 probability. For color augmentation, we employ the proposed method of[44] that generates realistic like synthetic images. Using[44], we randomly select one image out of 10 generated images for each image of the minibatch. For L_S (Eq. 5), we utilize label smoothing for the target probabilities within the cross-entropy to better tackle overfitting. This entails setting the probability of the correct class to 1 – φ with φ = 0.1, while assigning φ/(cl-1) as the probability for all other classes. Also L2 normalization is done to sampled noise before adding to feature. For inference, we first rescaled the image to shorter side with 224 and samples 3 network input's sized crops (a center crop and a crop from each of the two shorter sides) from the image before feeding to the network. All crops' feature vectors are then averaged to produce the feature representation of image. For matching we employ cosine similarity using L2 normalized features of gallery set to query.

Experiments

This section first discuss the dataset setting and evaluation measures. Then report the FGIR results and analyze the effect of noise-injection in retrieval performance. Finally, we also test our approach in context with Zero-shot learning.

	Training Set	Gallery set	Query Set
Category wise	40	25	15
Total images	680	425	255

Table 1. Five splits setting of Oxford Flowers-17.

Method	Splits					Mean
	1	2	3	4	5	
LBP[59]	0.098	0.099	0.101	0.103	0.102	0.101
HOG[58]	0.111	0.113	0.112	0.111	0.115	0.112
ResNet18 (Pretrained)	0.512	0.509	0.513	0.515	0.518	0.513
Yang et al.[21] (Vgg-16)	-	-	-	-	-	0.877
Kumar et al.[29]	0.901	0.923	0.946	0.931	0.940	0.928
Our Method	**0.947**	**0.934**	**0.959**	**0.939**	**0.947**	**0.946**

Table 2. Comparisons of mAPs on Oxford Flowers-17 under FGIR. Significant values are in [bold].

Datasets and evaluation setting

The experiments are conducted on two datasets, the Oxford Flowers-17[14] and the Cars-196[15]. Oxford Flowers-17 consists of 17 fine-grained categories with 1360 flower images. Cars-196 consists of 196 fine-grained classes of cars models with 16,185 images. Since Oxford Flowers-17 is a small dataset that contains 80 images per category, we conduct the experiment on randomly selected five splits of the dataset, and each split consist of three sets: training, gallery and query as depicted in Table 1. As a result, there are 680, 425 and 255 images for training, gallery and query sets, respectively. In the case of Cars-196 dataset, we conduct the experiment on the standard training testing split i.e. 8,144/8,041 images for training/testing. Note, the retrieval process is performed in the testing set by treating all images as queries, and the retrieved images are then evaluated by excluding the query image. MATLAB and NVIDIA Tesla K40c GPU are used to perform the experiments. To assess retrieval performance, we use Mean Average Precision (mAP) as described in[27].

Results and analysis

Results on Oxford Flowers-17 under FGIR setting

In this comparative analysis of proposed method with state-of-arts is done and results (mAPs) are reported in Table 2 for. It can be seen that handcrafted features perform poorly with mAPs of 0.101 (LPB[59]) and 0.112 (HOG[58]), as they are unable to distinguish subtle differences in fine-grained images because these methods are not designed by keeping subtle details into consideration. However, Deep CNN descriptors shows great improvement over handcrafted ones. For instance, pre-trained ResNet18 descriptors shows 0.513 mAP, which is around + 0.4 (mAP) improvement over handcrafted features. Further, with fine tuning on target dataset, performance is further enhanced with mAPs of 0.877 (Yang et al.[21]) and 0.928 (Kumar et al.[29]). With 0.946 mAP, the suggested approach is able to achieve better results than others, which confirm the importance of noise insertion while training the network on small datasets. Further, mAP@K is also depicted in Fig. 3, where we can see that our method gradually improves over fine-tuned R18[29] with the increase of K.

Moreover, Tables 3 and 4 depicts the categorical wise performance of Flowers-17 with comparative analysis with state-of-arts. From the results, we can observe the methods of[45,46] and[47] performs much better compared to HOG and LBP, and further[29] able to improves over these methods in 13 classes. Our method is able to outperform[29] in thirteen classes.

Results on Cars-196 under FGIR setting

Further, we compare our method with the SOTA on cars-196, which is reported in Tables 5 and 6 respectively. On comparing with baselines in Table 5, our method is able to achieve 80.2% mAP which is 3.7% higher than 76.5% of Kumar et al.[29] and far ahead of LBP and HOG. That mainly owes to the effectively learning of image representation through intensive augmentation in the form of noise. Along with LBP (0.007 mAP) and HOG (0.010 mAP), pretrained ResNet18's responses performs poorly with mAP of 0.041. This implies that for a larger number of fine-grained classes (compared to classes of flowers-17), the pretrained ResNet18 is unable to distinguish them. The reason is that through imagenet dataset[48] it is learned to focus on the global relationships of the object rather than object's subtle description. Furthermore, in the context of top-1 and top-5 mAP, we can see in Table 6 that our method consistently outperforms the SPOC[17], CroW[20], RMAC[22], Wei et al.[27] and Kumar et al.[29] with an 86.14% top1 mAP and 81.62% top5 mAP.

Ablation study

Effect of noise induced on retrieval performance

We conduct experiments on cars-196 to assess the impact of injected noise on retrieval performance. The findings, presented in the form of mAP at Top-k, are shown in Table 7, where our proposed work is performs well compared to other settings, e.g., 86.14% (with all loss) vs. 84.98 (with L_N and L_D) vs. 84.12%

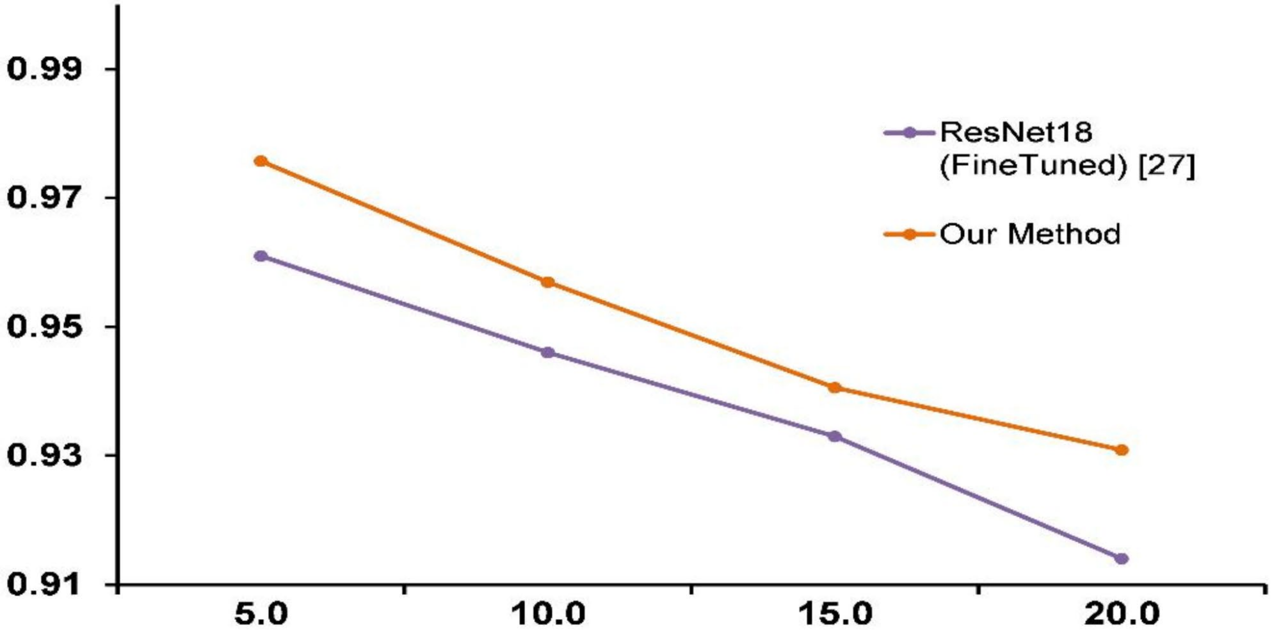

Fig. 3. Top k mAP comparison between[29] and our approach.

(with L_N and L_S) and 82.61% (with L_D) for Top-1 mAP. This also indicates inclusion of noises at both end benefits to learning generalizable features. Figure 4 further visualize the performance under different settings.

Fine-grained recognition
In this ablation study, we analyze the effect of our approach on recognition accuracy. For this we use the cars-196 dataset and the standard protocol for training and testing. We set the minibatch size to 64, learning rate to 0.0001 and data augmentation setting as discussed in Section "Training details". The results in the term of recognition accuracy are reported in Table 8, where we can see the boost in accuracy with our approach.

Zero shot learning
Next, we test the generalization of our method in the context of zero-shot setting, namely to test whether the proposed method helps to find discriminative features even for the unseen images. In this regard, following the settings in[34], we conduct the experiment on the Cars-196 and Cub-200-2011[49] datasets, where the first half classes are employed to train the network and the remaining half classes for testing purpose. We conduct the zero shot learning experiments using pytorch with max 40 epochs. We implement our method on both base networks: resnet18 (R18) and resnet50 (R50). First, we analyze the effectiveness of the proposed method on Cub-200-2011 and Cars-196 using experimentation setting (R18, embedding size = 512, learning rate = 0.002, gamma = 0.1 for every 15 epochs, batch_size = 240 with 12 samples per class) and the results are reported in Table 9, where we can see that by including L_N and L_S the retrieval performance tends to increase, which confirms using noise in L_N can help to incorporate intra-class variance and noise in L_S serves as a form of regularization.

Further, we analyze the effect of embedding size on retrieval performance (recall@k) which is depicted in Fig. 5, and effect of noise in L_S on Cub-200-2011 with our approach is shown in Fig. 6. In Figs. 7 and 8, we additionally depict the retrieval results for a randomly picked query from each dataset.

In Table 10, we can also see that our method is able to achieve better results compare to baseline methods such as EPSHN[50] and NormSoftmax[51] (where, EPSHN[50] is based on contrastive learning approach and NormSoftmax[51] is based on classification approach). For Resnet50 and Resnet101, we set the batch size to 144 and 24 samples per class. As per Table 10, our method consistently achieves better results for Cars-196 and Cub-200-2011 datasets in terms of recall@k than SOTA. However, few methods performs better than proposed method, which can be seen our method's limitation in context of Cub-200-2011 dataset due to small dataset. For SOP[31] dataset our model consistently achieves better results compare to others in Table 11. We can also see that compared to the baseline methods[50,51], the proposed method is able to improve its performance for all three datasets. This study confirms that our approach is able to generalize over unseen classes. We also show, with resnet101 model the proposed method is able to improve even more.

Conclusion
In this paper, a noise-assisted feature learning approach for FGIR is proposed which alleviates the expensive sampling process in triplet learning, and early saturation problem in softmax based learning. The deep CNN is jointly trained with multi loss objective dealing with class discriminative learning as well as noise invariant learning. Oxford flower 17 and cars-196 datasets are consider to validate our approach, where it achieves significant gains over existing schemes. Under the zero-shot setting, we achieved competitive results on cars-

Method			Flower Category								
			Bluebell	Buttercup	ColtsFoot	Cowslip	Crocus	Daffodil	Daisy	Dandelion	Fritillary
LBP[59]	Split	1	0.07	0.089	0.098	0.093	0.094	0.089	0.104	0.114	0.105
		2	0.073	0.094	0.094	0.088	0.137	0.091	0.085	0.12	0.096
		3	0.069	0.088	0.099	0.089	0.096	0.096	0.092	0.145	0.095
		4	0.067	0.124	0.093	0.09	0.113	0.074	0.089	0.138	0.103
		5	0.076	0.095	0.086	0.088	0.107	0.111	0.091	0.148	0.099
	Mean		0.071	0.098	0.094	0.0896	0.1094	0.0922	0.0922	0.133	0.0996
HOG[58]	Split	1	0.071	0.059	0.095	0.18	0.088	0.059	0.096	0.227	0.054
		2	0.091	0.077	0.087	0.182	0.093	0.056	0.093	0.144	0.076
		3	0.094	0.093	0.093	0.126	0.121	0.065	0.101	0.203	0.071
		4	0.081	0.085	0.15	0.196	0.139	0.057	0.09	0.189	0.063
		5	0.077	0.068	0.121	0.137	0.138	0.069	0.11	0.213	0.074
	Mean		0.0828	0.0764	0.1092	0.1642	0.1158	0.0612	0.098	0.1952	0.0676
Yang et al.[45]			0.58	0.43	0.5	0.7	0.7	0.53	0.58	0.38	0.63
Gao et al.[46]			0.46	0.71	0.68	0.5	0.68	0.73	0.83	0.8	0.73
Ahmed et al.[47]			0.89	0.92	0.92	0.89	0.94	0.95	0.95	0.99	0.9
Resnet18 (Pretrained)	Split	1	0.441	0.552	0.581	0.391	0.333	0.491	0.76	0.488	0.761
		2	0.391	0.398	0.55	0.414	0.401	0.49	0.798	0.475	0.833
		3	0.39	0.58	0.57	0.37	0.354	0.431	0.716	0.561	0.845
		4	0.331	0.584	0.502	0.331	0.288	0.421	0.726	0.562	0.814
		5	0.36	0.492	0.472	0.305	0.292	0.442	0.755	0.667	0.619
	Mean		0.3826	0.5212	0.535	0.3622	0.3336	0.455	0.751	0.5506	0.7744
Kumar et al.[29]	Split	1	0.922	0.976	0.936	0.839	0.782	0.911	0.977	0.917	0.899
		2	0.984	0.977	0.92	0.879	0.914	0.921	0.946	0.941	0.919
		3	0.969	0.99	0.944	0.897	0.925	0.937	0.979	0.935	0.953
		4	0.937	0.949	0.942	0.837	0.872	0.957	0.998	0.949	0.926
		5	0.966	0.976	0.934	0.848	0.828	0.965	1	0.955	0.854
	Mean		0.9556	0.9736	0.9352	0.86	0.8642	0.9382	0.98	0.9394	0.9102
Our Method	Split	1	0.95	0.945	0.936	0.933	0.906	0.924	1	0.938	0.937
		2	0.946	0.987	0.889	0.934	0.909	0.868	0.983	0.96	0.967
		3	0.908	0.994	0.92	0.958	0.927	0.958	0.945	0.989	0.988
		4	0.968	0.957	0.971	0.84	0.882	0.92	0.84	0.927	0.955
		5	0.921	0.989	0.969	0.912	0.814	0.885	0.999	0.944	0.915
	Mean		0.9386	0.9744	0.937	0.9154	0.8876	0.911	0.9534	0.9516	0.9524

Table 3. Comparison of mAPs of categories 1–9 on Oxford Flowers-17 under FGIR.

196, Cub-200-2011 and SOP datasets. The proposed approach exhibits great potential and can be explored in various industrial applications such as clothing retrieval, face retrieval, biomedical image retrieval, landmark retrieval, etc. The main limitation of this task may be the training time compared to normal CNN training which needs to be explore in larger networks. A second limitation might be that the loss of the proposed method primarily emphasizes a global perspective. This could be addressed by incorporating local attention mechanisms to capture subtle features more effectively. In subsequent work, we plan to leverage various deep variations of CNN and vision transformers to expand our approach to larger datasets. The applicability of these techniques can be evaluated in the medical field, utilizing both supervised and unsupervised learning techniques for potential advancements.

Method			Flower Category							
			Iris	LilyValley	Pansy	Snowdrop	SunFlower	TigerLily	Tulip	WindFlower
LBP[59]	Split	1	0.255	0.091	0.094	0.073	0.189	0.066	0.084	0.108
		2	0.141	0.073	0.096	0.076	0.154	0.074	0.073	0.116
		3	0.155	0.082	0.089	0.084	0.155	0.075	0.082	0.099
		4	0.189	0.089	0.12	0.063	0.113	0.086	0.063	0.115
		5	0.187	0.083	0.093	0.073	0.154	0.063	0.085	0.119
	Mean		0.1854	0.0836	0.0984	0.0738	0.153	0.0728	0.0774	0.1114
HOG[58]	Split	1	0.467	0.053	0.059	0.061	0.095	0.083	0.131	0.051
		2	0.384	0.055	0.061	0.05	0.108	0.088	0.126	0.044
		3	0.399	0.051	0.078	0.051	0.11	0.096	0.094	0.049
		4	0.417	0.05	0.065	0.048	0.05	0.111	0.063	0.055
		5	0.34	0.049	0.05	0.052	0.101	0.118	0.117	0.049
	Mean		0.4014	0.0516	0.0626	0.0524	0.0928	0.0992	0.1062	0.0496
Yang et al.[45]			0.18	0.68	0.58	0.65	0.58	0.45	0.51	0.20
Gao et al.[46]			0.90	0.75	0.83	0.75	0.88	0.80	0.40	0.88
Ahmed et al.[47]			1.00	0.70	0.93	0.70	0.95	0.85	0.91	0.95
Resnet18 (Pretrained)	Split	1	0.533	0.541	0.687	0.354	0.833	0.572	0.294	0.672
		2	0.441	0.428	0.645	0.398	0.692	0.581	0.277	0.643
		3	0.484	0.388	0.684	0.348	0.701	0.593	0.295	0.558
		4	0.591	0.438	0.677	0.338	0.748	0.672	0.275	0.739
		5	0.634	0.439	0.568	0.401	0.719	0.643	0.264	0.668
	Mean		0.5366	0.4468	0.6522	0.3678	0.7386	0.6122	0.281	0.656
Kumar et al.[29]	Split	1	0.858	0.898	0.990	0.852	0.998	0.933	0.756	0.957
		2	0.799	0.922	0.999	0.904	0.978	0.959	0.814	0.928
		3	0.960	0.940	0.988	0.886	0.981	0.979	0.846	0.976
		4	0.881	0.976	0.994	0.904	0.999	0.967	0.814	0.926
		5	0.995	0.962	0.999	0.935	1.000	0.973	0.830	0.976
	Mean		0.899	0.940	0.994	0.897	0.991	0.962	0.812	0.953
Our Method	Split	1	0.904	0.946	1.000	0.965	0.996	0.998	0.831	0.991
		2	0.792	0.970	0.997	0.903	0.950	0.960	0.869	0.990
		3	0.997	0.941	0.988	0.965	0.961	0.998	0.884	0.974
		4	0.995	0.990	1.000	0.982	0.987	0.951	0.887	0.905
		5	0.999	0.967	1.000	0.993	1.000	0.988	0.806	0.991
	Mean		0.938	0.963	0.997	0.962	0.979	0.979	0.856	0.970

Table 4. Comparison of mAPs of categories 10–17 on Oxford Flowers-17 under FGIR.

Method	LBP	HOG	ResNet18 (Pretrained)	Kumar et al.[29]	Our method
mAP	0.011	0.013	0.045	0.765	**0.802**

Table 5. Comparison of mAPs on Cars-196 under FGIR. Significant values are in [bold].

Method	SPOC[17]	CroW[20]	R-MAC[22]	Wei et al.[27]	Kumar et al.[29]	Our method
Top1 mAP	29.86%	44.92%	46.54%	53.30%	84.11%	**86.14%**
Top5 mAP	36.23%	51.18%	52.98%	59.11%	80.09%	**81.62%**

Table 6. Performance (mAP) Comparison on Cars-196 under FGIR. Significant values are in [bold].

Approach						
Approach	L_N	-	+	-	+	+
	L_D	+	-	+	+	+
	L_S	-	+	+	-	+
mAP	Top1	0.8261	0.8412	0.8487	0.8498	**0.8614**
	Top2	0.8051	0.8229	0.8285	0.8301	**0.8348**
	Top3	0.7987	0.8127	0.8111	0.8192	**0.8285**
	Top4	0.7853	0.8058	0.8023	0.8046	**0.8212**
	Top5	0.7735	0.8009	0.8011	0.8003	**0.8162**

Table 7. Top k mAP when different settings on Cars-196 under FGIR. '+' indicates inclusion of objective '–' otherwise. Significant values are in [bold].

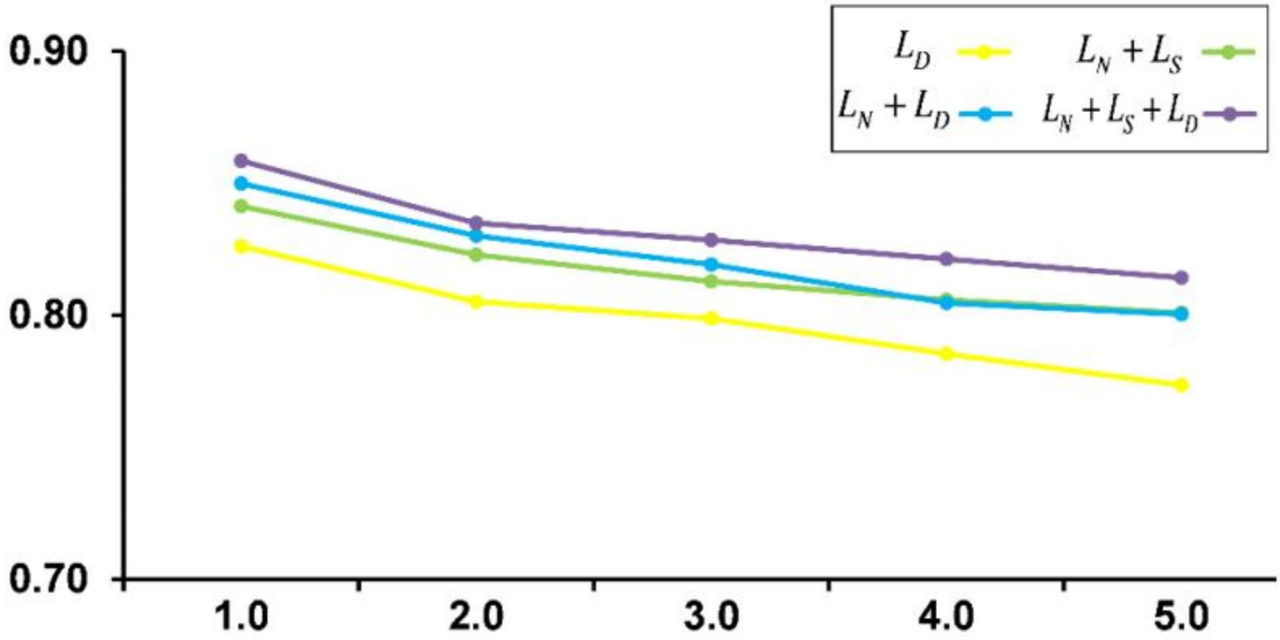

Fig. 4. Top k mAP when different settings on Cars-196.

Base network	Loss	Cars-196
Resnet18	Standard cross entropy	85.23%
	L_S	86.66%
	$L_S + L_N$	86.75%
	$L_S + L_D$	86.97%
	$L_N + L_D$	85.43%
	$L_S + L_D + L_N$	87.38%

Table 8. Recognition performance (accuracy) analysis on Cars-196.

	Cub-200-2011				Cars-196			
	Recall@1	Recall@2	Recall@4	Recall@8	Recall@1	Recall@2	Recall@4	Recall@8
L_D	62.09	73.80	82.51	89.13	83.97	89.76	93.80	96.20
$L_D + L_N$	62.39	73.89	82.63	89.46	84.13	90.11	94.13	96.64
$L_D + L_S$	63.28	73.87	82.78	89.72	85.24	91.12	94.81	96.78
$L_D + L_S + L_N$	63.37	74.14	83.24	90.51	85.75	91.64	94.91	96.88

Table 9. Analysis of proposed method on Cub-200-2011 and Cars-196 using R18.

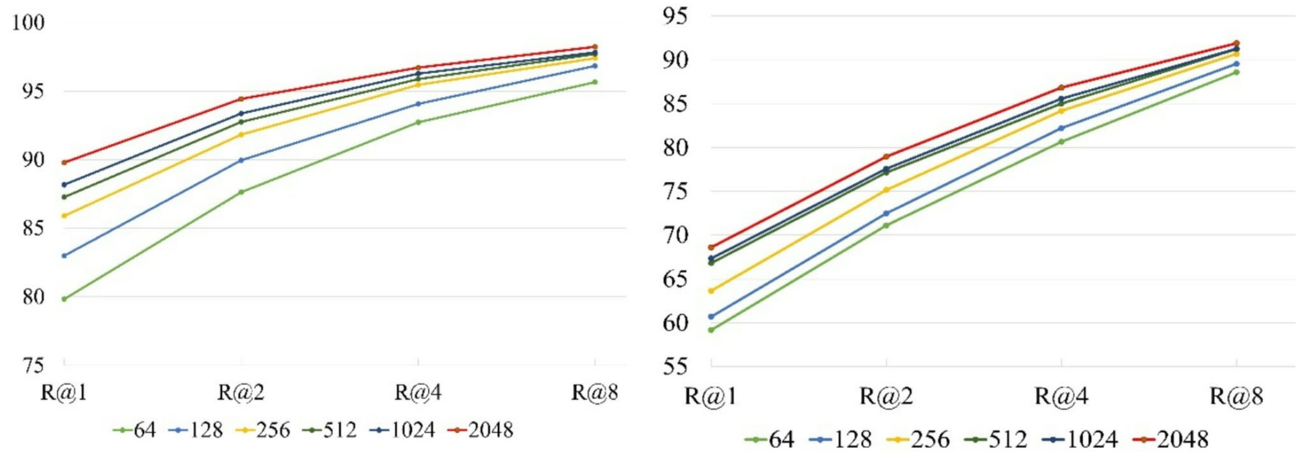

Fig. 5. Effect of embedding size on Cars-196 (Left) and Cub-200-2011 (right) with our approach (R50).

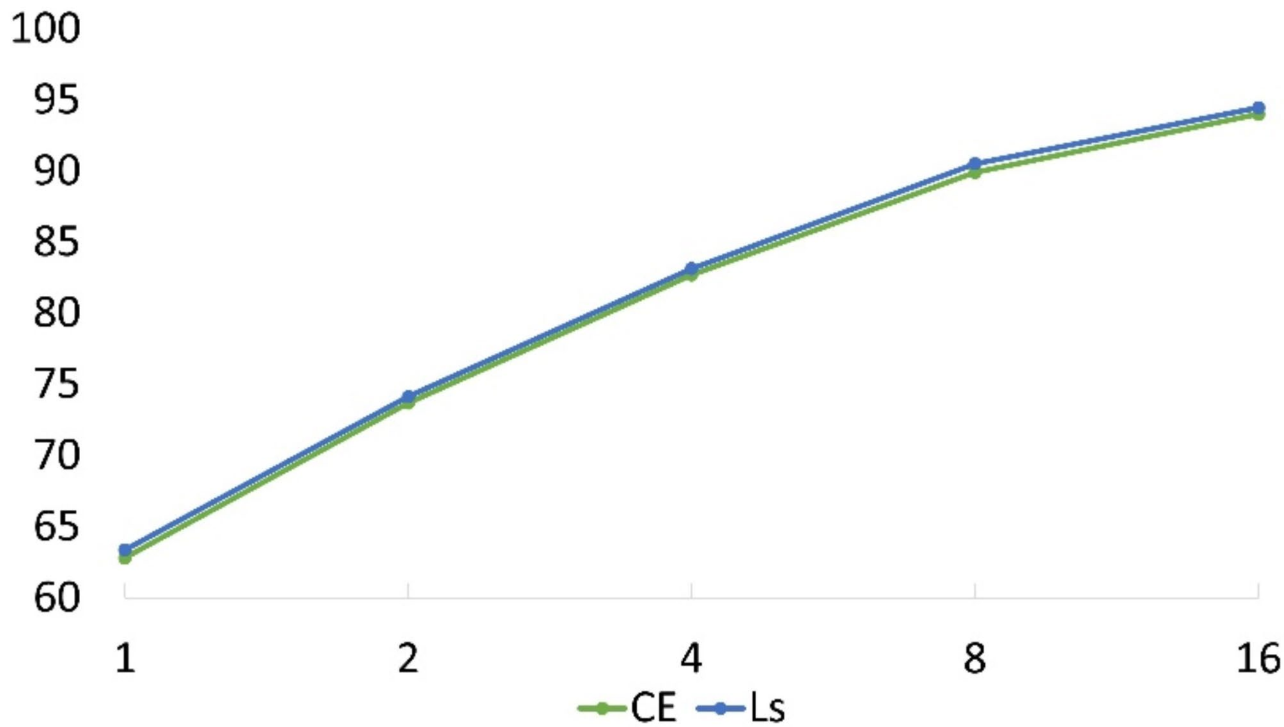

Fig. 6. Effect of noise in L_S on Cub-200-2011 with our approach (R18).

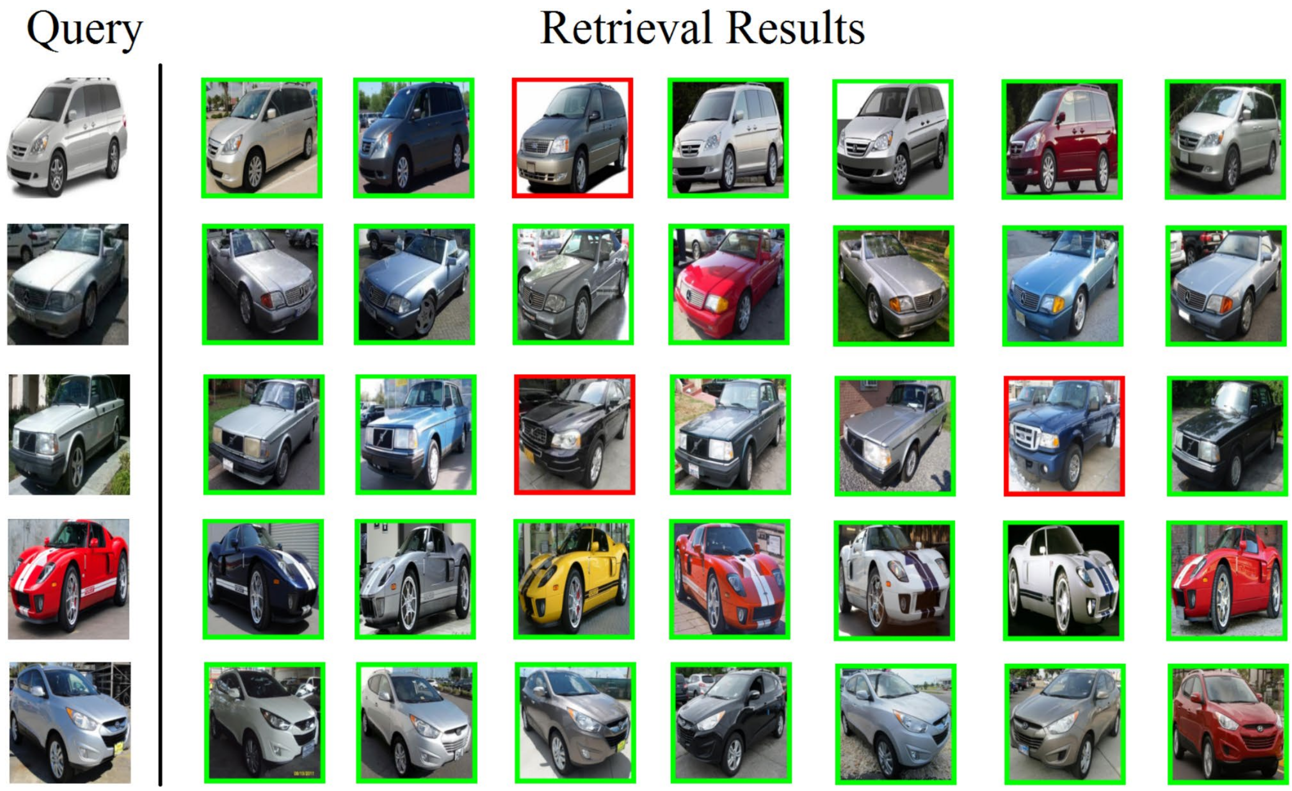

Fig. 7. Findings on Cars-196 dataset. The retrieved instance is indicated correctly by a green boundary box, and incorrectly by a red boundary box. Dataset Source: https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset?datasetId=30084&sortBy=dateCreated&select=cars_test.

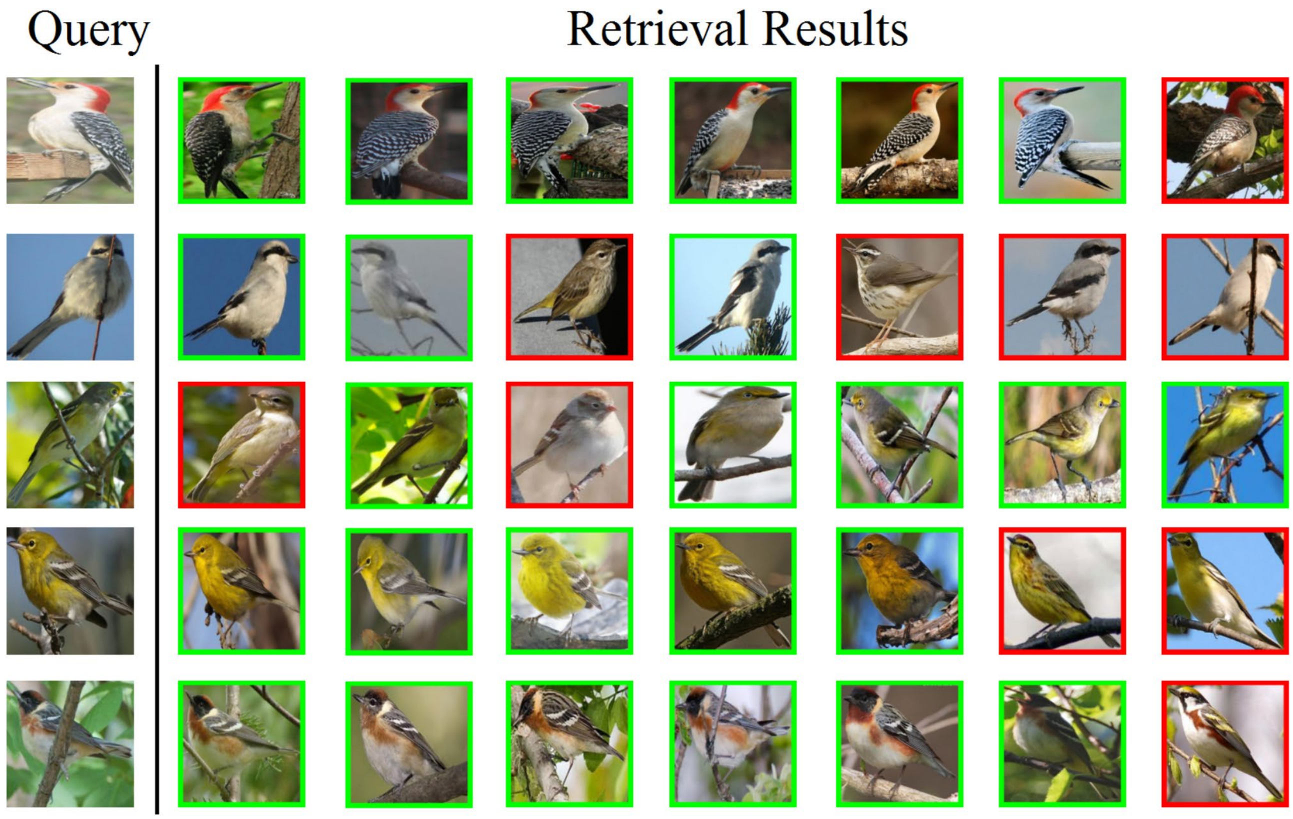

Fig. 8. Findings on Cub-200-2011 dataset. The retrieved instance is indicated correctly by a green boundary box, and incorrectly by a red boundary box. Dataset Source: https://www.vision.caltech.edu/datasets/cub_200_2011/.

Method	CARS-196					Cub-200-2011				
	k = 1	k = 2	k = 4	k = 8	k = 16	k = 1	k = 2	k = 4	k = 8	k = 16
Triplet[7]	39.1	50.4	63.3	74.5	84.1	36.1	48.6	59.3	70.0	80.2
LiftedStruct[31]	49.0	60.3	72.1	81.5	89.2	47.2	58.9	70.2	80.2	89.3
N-pairs[32]	53.9	66.8	77.7	86.3	-	45.4	58.4	69.5	79.4	-
SCDA[27]	58.5	69.8	79.1	86.2	91.8	62.2	74.2	83.2	90.1	94.3
CRL-WSL[28]	63.9	73.7	82.1	89.2	93.7	65.9	76.5	85.3	90.3	94.4
DGCRL[34]	75.9	83.9	89.7	94.0	96.6	67.9	79.1	86.2	91.8	94.8
EPSHN[50]	82.7	89.3	93.0	-	-	64.9	75.3	83.5	-	-
Zheng et al.[37]	81.1	88.8	93.7	96.7	-	55.2	68.7	79.0	89.5	-
Duan et al.[38]	78.2	86.2	92.0	95.5	-	61.2	73.7	83.3	90.3	-
Yingying et al. (VGG16-based)[30]	73.2	82.1	88.6	93.2	95.4	67.5	78.2	86.7	92.0	95.1
D & C[39]	87.76	70.67	65.97	-	-	68.16	69.49	55.35	-	-
Yingying et al. res101-based)[30]	85.4	91.2	94.4	96.5	97.7	73.1	81.5	86.6	92.7	95.4
McSAP[52]	84.6	91.5	95.1	97.4	-	63.5	75.6	84.8	91.3	-
Adaptive hierarchical[53]	82.4	89.5	93.8	95.9	-	65.3	76.1	84.7	90.7	-
HSE-EPSHN[54]	85.4	91.2	96.9	-	-	66.9	77.4	85.5	-	-
HSE-PA[54]	89.6	93.8	96.0	-	-	70.6	80.1	87.1	-	-
Multi-Proxy[55]	90.3	93.7	96.3	-	-	69.6	79.9	87.0	-	-
Anti-Collapse[56]	90.5	94.6	-	-	-	71.7	81.2	-	-	-
NormSoftmax^{512}[51]	84.2	90.4	94.4	96.9	-	61.3	73.9	83.5	90.0	-
NormSoftmax^{2048}[51]	89.3	94.1	96.4	98.0	-	65.3	76.7	85.4	91.8	-
Our Method (R18)^{512}	**85.75**	**91.64**	**94.91**	**96.88**	**98.70**	**63.37**	**74.14**	**83.24**	**90.51**	**94.25**
Our Method (R50)^{512}	**87.27**	**92.74**	**95.87**	**97.70**	**98.83**	**66.81**	**77.14**	**85.01**	**91.24**	**94.55**
Our Method (R50)^{1024}	**88.17**	**93.36**	**96.27**	**97.82**	**99.03**	**67.34**	**77.57**	**85.57**	**91.27**	**95.59**
Our Method (R50)^{2048}	**89.78**	**94.43**	**96.70**	**98.22**	**98.94**	**68.60**	**78.95**	**86.83**	**91.90**	**95.0**
Our Method (R101)^{512}	**90.22**	**94.34**	**96.65**	**98.16**	**98.94**	**69.04**	**79.29**	**86.66**	**92.10**	**95.44**
Our Method (R101)^{2048}	**91.33**	**95.20**	**97.34**	**98.55**	**99.13**	**71.59**	**81.62**	**88.18**	**92.91**	**95.90**

Table 10. Performance (Recall@k) Comparison under Zero-shot setting. Significant values are in [bold].

Method	SOP		
	k = 1	k = 10	k = 100
EPSHN[50]	78.3	90.7	96.3
Zheng et al.[37]	70.7	85.0	93.7
D & C[39]	79.77	90.39	95.20
Adaptive hierarchical[53]	73.6	86.9	94.8
McSAP[52]	79.9	91.5	96.5
SGSL[57]^{512}	81.4	91.8	96.2
SGSL[57]^{2048}	83.19	93.0	97.0
HSE[54]	80.0	91.4	96.3
Multi-Proxy[55]	80.1	91.3	96.6
Anti-Collapse[56]	81.2	92.0	-
NormSoftmax^{512}[51]	78.2	90.6	96.2
NormSoftmax^{2048}[51]	79.5	91.5	96.7
Our Method (R50)^{512}	**80.2**	**91.2**	**95.8**
Our Method (R50)^{2048}	**81.8**	**92.1**	**96.2**
Our Method (R101)^{2048}	**83.21**	**93.2**	**97.08**

Table 11. Performance (Recall@k) Comparison under Zero-shot setting for SOP[31] dataset. Significant values are in [bold].

Data availability

All images used in Figures 1, 7, and 8 are sourced from publicly available datasets intended for research purposes. Therefore, permission for their use is not required. The data that support the findings of this study and publicly available datasets are available at https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html; https:

//www.vision.caltech.edu/datasets/cub_200_2011/; https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset?datasetId=30084&sortBy=dateCreated&select=cars_test.

Received: 10 July 2024; Accepted: 4 April 2025
Published online: 03 May 2025

References

1. Zhou, W., Li, H. & Tian, Q. Recent advance in content-based image retrieval: A literature survey (2017).
2. Xie, L., Wang, J., Zhang, B. & Tian, Q. Fine-grained image search. *IEEE Trans. Multimed.* **17**, 636–647 (2015).
3. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Adv. Neural Inf. Process. Syst.* 1097–1105 (2012).
4. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* (2015).
5. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 770–778 (2016).
6. Bell, S. & Bala, K. Learning visual similarity for product design with convolutional neural networks. In *ACM Trans. Graph.* (2015).
7. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B. & Wu, Y. Learning fine-grained image similarity with deep ranking. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 1386–1393 (2014).
8. Schroff, F., Kalenichenko, D. & Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 815–823 (2015).
9. Chen, W., Chen, X., Zhang, J. & Huang, K. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition* 403–412 (CVPR, 2017).
10. Huang, C., Loy, C. C. & Tang, X. Local similarity-aware deep feature embedding. In *Adv. Neural Inf. Process. Syst.* 1270–1278 (2016).
11. Manmatha, R., Wu, C. Y., Smola, A. J. & Krahenbuhl, P. Sampling matters in deep embedding learning. In *Proc. IEEE Int. Conf. Comput. Vis.* 2840–2848 (2017).
12. Ge, W., Huang, W., Dong, D. & Scott, M. R. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)* 269–285 (2018).
13. Roth, K., Milbich, T., Ommer, B. Pads: Policy-adapted sampling for visual similarity learning. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 6567–6576 (2020).
14. Nilsback, M. E. & Zisserman, A. A visual vocabulary for flower classification. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 1447–1454 (2006).
15. Krause, J., Stark, M., Deng, J., Fei-Fei, L. 3D object representations for fine-grained categorization. In *Proc. IEEE Int. Conf. Comput. Vis.* 554–561 (2013).
16. Babenko, A., Slesarev, A., Chigorin, A. & Lempitsky, V. Neural codes for image retrieval. In *European Conference on Computer Vision* 584–599 (2014).
17. Yandex, A. B. & Lempitsky, V. Aggregating local deep features for image retrieval. In *Proc. IEEE Int. Conf. Comput. Vis.* 1269–1277 (2015).
18. Mohedano, E., Mcguinness, K., O'Connor, N. E., Salvador, A., Marqués, F. & Giró-I-nieto, X. Bags of local convolutional features for scalable instance search. In *ICMR 2016 - Proc. 2016 ACM Int. Conf. Multimed. Retr.* 327–331 (2016).
19. Ng, J. Y. H., Yang, F. & Davis, L. S. Exploiting local features from deep networks for image retrieval. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.* 53–61 (2015).
20. Kalantidis, Y., Mellina, C. & Osindero, S. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision* 685–701 (2016).
21. Yang, H. F., Lin, K. & Chen, C. S. Cross-batch reference learning for deep classification and retrieval. In *MM 2016 - Proc. 2016 ACM Multimed. Conf.* 1237–1246 (2016).
22. Tolias, G., Sicre, R. & Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. In *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.* (2016).
23. Shakarami, A. & Tarrah, H. An efficient image descriptor for image classification and CBIR. *Optik* **214**, 164833 (2020).
24. Zhang, X., Xiong, H., Zhou, W., Lin, W. & Tian, Q. Picking deep filter responses for fine-grained image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 1134–1142 (2016).
25. Watkins, R., Pears, N. & Manandhar, S. Vehicle classification using ResNets, localisation and spatially-weighted pooling (2018).
26. Zhou, F. & Lin, Y. Fine-grained image classification by exploring bipartite-graph labels. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 1124–1133 (2016).
27. Wei, X. S., Luo, J. H., Wu, J. & Zhou, Z. H. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Trans. Image Process.* **26**, 2868–2881 (2017).
28. Zheng, X., Ji, R., Sun, X., Wu, Y., Huang, F. & Yang, Y. Centralized ranking loss with weakly supervised localization for fine-grained object retrieval. In *IJCAI Int. Jt. Conf. Artif. Intell.* 1226–1233 (2018).
29. Kumar, V., Tripathi, V. & Pant, B. Content based fine-grained image retrieval using convolutional neural network. In *2020 7th Int. Conf. Signal Process. Integr. Networks* 1120–1125 (SPIN, 2020).
30. Zhu, Y., Cao, G., Yang, Z. & Xiufan, Lu. Learning relation-based features for fine-grained image retrieval. *Pattern Recogn.* **140**, 109543 (2023).
31. Song, H. O., Xiang, Y., Jegelka, S. & Savarese, S. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 4004–4012 (2016).
32. Sohn, K. Improved deep metric learning with multi-class N-pair loss objective. In *Adv. Neural Inf. Process. Syst.* 1857–1865 (2016).
33. Song, H. O., Jegelka, S., Rathod, V. & Murphy, K. Deep metric learning via facility location. In *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition* 2206–2214 (CVPR, 2017).
34. Zheng, X., Ji, R., Sun, X., Zhang, B., Wu, Y. & Huang, F. Towards optimal fine grained retrieval via decorrelated centralized loss with normalize-scale layer. In *Proc. AAAI Conf. Artif. Intell.* Vol. 33, 9291–9298 (2019).
35. Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R. & Robertson, N. M. Ranked list loss for deep metric learning. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 5207–5216 (2019).
36. Duan, Y., Chen, L., Lu, J. & Zhou, J. Deep embedding learning with discriminative sampling policy. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 4964–4973 (2019).
37. Zheng, W., Lu, J. & Zhou, J. Hardness-aware deep metric learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**, 3214–3228. https://doi.org/10.1109/TPAMI.2020.2980231 (2021).
38. Duan, C. et al. Multilevel similarity-aware deep metric learning for fine-grained image retrieval. *IEEE Trans. Industr. Inf.* **19**(8), 9173–9182. https://doi.org/10.1109/TII.2022.3227721 (2023).
39. Sanakoyeu, A., Ma, P., Tschernezki, V. & Ommer, B. Improving deep metric learning by divide and conquer. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(11), 8306–8320 (2022).

40. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning* 1597–1607 (PMLR, 2020).
41. Rodner, E., Simon, M., Fisher, R. B. & Denzler, J. Fine-grained recognition in the noisy wild: Sensitivity analysis of convolutional neural networks approaches. In: *Br. Mach. Vis. Conf. 2016* 60.1–60.13 (BMVC, 2016).
42. Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J. & Fei-Fei, L. The unreasonable effectiveness of noisy data for fine-grained recognition. In *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 301–320 (2016).
43. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. arXiv preprint arXiv:1607.06450 (2016).
44. Afifi, M. & Brown, M. What else can fool deep learning? Addressing color constancy errors on deep neural network performance. In *Proc. IEEE Int. Conf. Comput. Vis.* 243–252 (2019).
45. Yang, J., Yu, K., Gong, Y. & Huang, T. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. IEEE Int. Conf. Comput. Vis.* 1794–1801 (2009).
46. Gao, S., Tsang, I. W. H. & Ma, Y. Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *IEEE Trans. Image Process.* **23**, 623–634 (2014).
47. Ahmed, K. T., Ummesafi, S. & Iqbal, A. Content based image retrieval using image features information fusion. *Inf. Fusion* **51**, 76–99 (2019).
48. Russakovsky, O. et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
49. Wah, C., Branson, S., Welinder, P., Perona, P. & Belongie, S. The caltech-ucsd birds-200-2011 dataset (2011).
50. Xuan, H., Stylianou, A. & Pless, R. Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA* 2474–2482 (2020).
51. Zhai, A. & Wu, H.-Y. Classification is a strong baseline for deep metric learning. arXiv [cs.CV]. http://arxiv.org/abs/1811.1264 (2018).
52. Zhao, J.-M. & Lian, Q.-S. Multi-centers SoftMax reciprocal average precision loss for deep metric learning. *Neural Comput. Appl.* **35**(16), 11989–11999 (2023).
53. Yan, J., Luo, L., Deng, C. & Huang, H. Adaptive hierarchical similarity metric learning with noisy labels. *IEEE Trans. Image Process.* **32**, 1245–1256 (2023).
54. Yang, B., Sun, H., Li, F. W., Chen, Z., Cai, J. &Song, C. HSE: Hybrid species embedding for deep metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 11047–11057 (2023).
55. Chan, P. P., Li, S., Deng, J. & Yeung, D. S. Multi-proxy based deep metric learning. *Inf. Sci.* **643**, 119120 (2023).
56. Jiang, X., Yao, Y., Dai, X., Shen, F., Nie, L. & Shen, H. T. Anti-collapse loss for deep metric learning. *IEEE Trans. Multimed.* (2024).
57. Yang, L., Wang, P. & Zhang, Y.. Stop-gradient softmax loss for deep metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 37, 3164–3172 (2023).
58. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition* 886–893 (CVPR, 2005).
59. Ojala, T., Pietikäinen, M. & Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 971–987 (2002).

Author contributions

V.K.: write original draft; V.T. and B.P.: Supervision; P.S. and M.D.: writing, review and editing; A.B.: validation and analysis.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.