One-Shot Fine-Grained Instance Retrieval

Hantao Yao[1,2], Shiliang Zhang[3], Yongdong Zhang[1,2], Jintao Li[1], Qi Tian[4]

[1]Key Lab of Intelligent Information Processing of CAS, Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China
[4] Department of Computer Science University of Texas at San Antonio, San Antonio, USA
{yaohantao,zhyd,jtli}@ict.ac.cn,slzhang.jdl@pku.edu.cn,qitian@cs.utsa.edu

ABSTRACT

Fine-Grained Visual Categorization (FGVC) has achieved significant progress recently. However, the number of fine-grained species could be huge and dynamically increasing in real scenarios, making it difficult to recognize unseen objects under the current FGVC framework. This raises an open issue to perform large-scale fine-grained identification without a complete training set. Aiming to conquer this issue, we propose a retrieval task named One-Shot Fine-Grained Instance Retrieval (OSFGIR). "One-Shot" denotes the ability of identifying unseen objects through a fine-grained retrieval task assisted with an incomplete auxiliary training set. This paper first presents the detailed description to OSFGIR task and our collected *OSFGIR-378K* dataset. Next, we propose the Convolutional and Normalization Networks (CN-Nets) learned on the auxiliary dataset to generate a concise and discriminative representation. Finally, we present a coarse-to-fine retrieval framework consisting of three components, *i.e.*, coarse retrieval, fine-grained retrieval, and query expansion, respectively. The framework progressively retrieves images with similar semantics, and performs fine-grained identification. Experiments show our OSFGIR framework achieves significantly better accuracy and efficiency than existing FGVC and image retrieval methods, thus could be a better solution for large-scale fine-grained object identification.

KEYWORDS

One-Shot Fine-Grained Instance Retrieval, Fine-Grained Visual Categorization, CNN, CN-Nets, OSFGIR-378K

1 INTRODUCTION

Different from conventional object categorization, Fine-Grained Visual Categorization (FGVC) aims to identify objects belonging to the same or closely-related species that only experienced experts can recognize, *e.g.*, identify a bird as "Black footed Albatross" or "Sooty Albatross". Due to the ability of providing valuable information
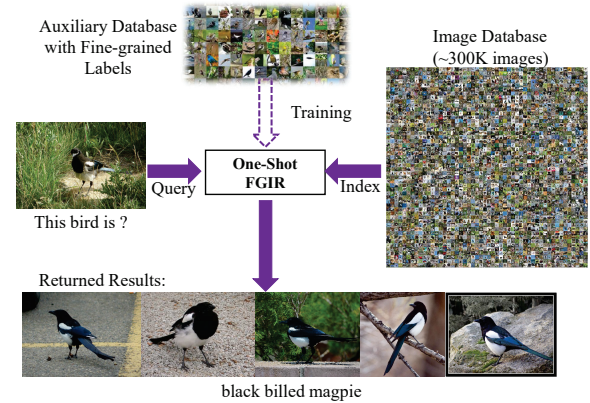
Figure 1: Illustration of One-Shot FGIR, which uses prior knowledge inferred from a small independent auxiliary dataset (dashed arrow) to perform fine-grained query of an unseen instance from a large-scale database (solid arrow).

to users, FGVC has been attracting lots of attentions [4, 7, 8, 14–16, 27, 30, 32–35, 41, 48, 53]. Although FGVC is challenging, its performance has been significantly improved by using powerful Convolutional Neural Networks [22, 27, 32, 53], considering detailed part localization [7, 30, 35, 41, 53], and generating better visual descriptions [4, 27, 35, 41]. For instance, the classification accuracy on CUB-200-2011 [45] has been pushed from 17.31% [45] to 85.5% [35] within five years.

As a special case of visual categorization, FGVC is designed to identify the species existing in the training set. However, the number of fine-grained species in real world could be huge and varying, *e.g.*, new shoes are being designed and produced every week, making it difficult to get a complete training set to recognize unseen objects under the FGVC framework. In other words, FGVC is powerful for scenarios like passenger plane classification, where the number of species is small and the complete training set is easy to acquire. Therefore, it is still an open issue how to perform fine-grained identification without a complete training set.

Motivated to conquer this issue, we present a novel problem named One-Shot Fine-Grained Instance Retrieval (OSFGIR), where "One-Shot" emphasizes the ability of identifying the unseen fine-grained species through a retrieval task. As illustrated in Fig. 1, OSFGIR takes an image as input, then performs fine-grained instance retrieval within a large-scale dataset without category labels, and finally returns images containing the identical object. OSFGIR can be tackled by learning powerful features and retrieval models. It is not restricted by the number of learned classifiers, thus has

Hantao Yao[1,2], Shiliang Zhang[3], Yongdong Zhang[1,2], Jintao Li[1], Qi Tian[4]

potential to show better generalization ability to unseen species than the FGVC framework. To facilitate model and feature learning, we introduce a small independent auxiliary training set. This incomplete auxiliary set is labeled with fine-grained species and is easy to collect. We thus call this task as One-Shot FGIR, because the training set is small and independent with the testing set.

In this paper, we firstly give the detailed description of OSFGIR, and introduce the OSFGIR-378K dataset. To extract a powerful image feature, we then propose a deep model called Convolutional and Normalization Networks (CN-Nets), which learns and combines two complementary features to generate the object description. Finally, we present a coarse-to-fine OSFGIR framework that consists of coarse retrieval, fine-grained retrieval, and query expansion, respectively. Given a query image, the coarse retrieval firstly returns the Top-K similar images based on a compact descriptor. A more powerful descriptor is hence extracted to rank the Top-K images. Finally, query expansion is used to further improve the retrieval performance. Experimental results show that our feature and retrieval approach significantly outperform existing deep features and retrieval methods in the aspects of both accuracy and efficiency.

OSFGIR is different from and substantially more challenging than traditional instance retrieval task. Most of the instance retrieval datasets are designed for partial-duplicate or semantic-similar search tasks, e.g., Oxford5K [37] and Holidays [23]. Such problems could be effectively solved by extracting and matching robust local features, i.e., Scale-invariant feature transform (SIFT) [36], or extracting semantic features by off-the-shelf deep leaning models [40]. OSFGIR aims to return images containing the identical fine-grained specie in the query, e.g., images of "black billed magpie" with different poses, sizes, backgrounds, etc. For OSFGIR, more powerful features are required to identify the fine-grained details among species, because different species may exhibit similar appearances and semantics.

OSFGIR is also a novel retrieval task different from most of existing FGVC works. The most related FGVC work is [49], where Xie et al. present a fine-grained image search algorithm. Similar to FGVC pipeline, Xie et al.[49] learns a series of classifiers based on the training set, and then identifies the fine-grained species with the learned classifiers. Note that, the training and testing datasets share the same species in [49]. OSFGIR differs from [49] in that, it uses independent training and testing datasets, i.e., uses a small incomplete auxiliary dataset for training, but a large-scale dataset for retrieval, which corresponds to more realistic settings.

For the past several years, lots of FGVC works [20, 30, 35, 41, 52, 53] have been proposed, and they focus on generating image representations from object parts. However, these representations are either complex or require expensive part annotations. CN-Nets is proposed with the motivation of designing a concise representation easy to implement and repeat. It reveals the shortcomings of CNN in feature learning and significantly outperforms the latest deep models in the aspects of efficiency, training complexity, and classification accuracy. FGVC framework is difficult to recognize unseen fine-grained species. This paper defines the OSFGIR problem and presents the OSFGIR-378K dataset. Compared with FGVC, OSFGIR is shown as a better solution for large-scale fine-grained object identification, e.g., our method significantly outperforms recent FGVC works on OSFGIR-378K by more than 11% on Mean Average Precision. We will release the OSFGIR-378K, and continually enlarge this dataset by adding more species to benefit OSFGIR and large-scale fine-grained object recognition research.

2 RELATED WORK

OSFGIR is related to works on fine-grained visual categorization [4, 7, 8, 14–16, 20, 27, 30, 33–35, 41, 47, 48, 52, 53] and deep learning-based visual retrieval [2, 17, 18, 26, 39, 46, 51]. In the following, we summarize these two categories of works respectively.

Fine-Grained Visual Categorization: In the past five years, researchers have significantly boosted the classification accuracy of FGVC. Existing methods could be summarized into four categories according to the type of image representation they use, *i.e.,* 1) part-based methods, 2) attribute-based methods, 3) object-based methods, and 4) global-description based methods. 1) As the CUB-200-2011 dataset provides 15 part annotations, the authors of [4] employ the labeled part annotations for training and testing to generate the part description. Based on the labeled part annotations for training images, the other works [7, 20, 30, 33, 34, 52, 53] firstly infer the part annotations for testing images, then generate the part descriptions. As most fine-grained datasets lack manually labeled part annotations, some works infer part labels with unsupervised methods [15, 27, 41]. 2) Recently, Liu *et al.* [35] employ the given attributes of each part to infer the part annotations, which are then used to generate the object description. 3) Besides the descriptions from local parts, the description from the object bounding box is also commonly used to identify the fine-grained species. [53] and [27] infer the bounding boxes for testing images based on those of training images. [48] and [41] generate bounding boxes for training and testing images only with image-level labels. 4) Different from the methods mentioned above, the Bilinear CNN [32] and Spatial Transformer Networks (STN) [22] generate a robust global description for FGVC with a forward pass of the CNN.

CNN for Visual Retrieval: CNN has exhibited promising performance for various vision tasks. Several works have attempted to apply CNN in image and instance retrieval [3, 17, 18, 26, 39, 44, 51]. NeuralCode [3] is an early work that applies CNN for image retrieval, *e.g.,* Babenko *et al.* employ the output of fully-connected layer as image feature for retrieval. Since Vector Locally Aggregated Descriptors (VLAD) [24] shows good retrieval performance by encoding SIFT descriptors. Ng *et al.* [51] replace the SIFT with CNN feature and encode the convolutional feature maps into a global feature with VLAD. In [44], Tolias *et al.* demonstrate that simply applying a spatial max-pooling over all locations on convolutional feature maps produces an effective visual descriptor. Instance retrieval differs slightly from image retrieval, because it focuses on image regions containing the target object, rather than the entire image. Given the object bounding boxes of query images, Tolias *et al.* [44] propose approximate integral max-pooling to select the best matching bounding box from hundreds of candidates. Different from [44], Salvador *et al.* [39] and Gordo *et al.* [18] apply Faster R-CNN [38] to reduce the number of candidate proposals.

OSFGIR differs from FGVC because it is a retrieval task, thus is able to query and identify the unseen query object. OSFGIR is also different from most of the visual retrieval tasks because it needs to further identify and capture the subtle differences among visually

and semantically similar objects. Among recent visual retrieval methods, Gordo *et al.* [18] have achieve promising performance. However, the method in [18] is not suitable for OSFGIR because: 1) it works on partial-duplicate image retrieval and is evaluated on the widely-used Oxford5K [37] and Holidays [23]. OSFGIR aims to return images containing the identical fine-grained specie in the query. Those two problems are quite different. 2) The deep regional feature training in [18] involves keypoint matching to generate the bounding boxes for each candidate object, thus is more suited to partial-duplicate image search.

In the next section, we proceed to give the formulation of OSFGIR, then introduce the OSFGIR-378K dataset.

3 PROBLEM FORMULATION

3.1 One-Shot Fine-Grained Instance Retrieval

As illustrated in Fig. 1, OSFGIR defines a fine-grained instance retrieval task assisted with a small independent training set. We denote the set of query images as $Q = \{q_1, ..., q_n\}$, where n is the number of query images. Each query image has a ground truth label $p_c (0 \leq p_c \leq \mathcal{P}_c, 0 \leq c \leq C)$, which denotes the p-th fine-grained specie in the c-th object category. Note that, we use "object" to denote the coarse category and "specie" to denote the fine-grained specie within a coarse object category. C thus is the total number of objects in the query set, $\mathcal{P}_c$ is the number of species in the c-th object category. We denote the image database as $\mathcal{D} = \{d_1, , ..., d_m\}$, where each image either contains a specie in one of the C categories, or could be a distracter for the retrieval.

Given a query q, OSFGIR retrieves the specie in q from $\mathcal{D}$, and returns a ranked list of images. If the query specie exists in the database, OSFGIR aims to return images containing the identical specie. For query species do not exist in the database, OSFGIR returns other species with similar appearances and semantics. Because it does not learn a fixed set of classifiers, OSFGIR is potential to show better generalization ability to new species than existing FGVC methods.

OSFGIR is challenging because fine-grained species commonly exhibit subtle inter-class variance and large intra-class variance. To better tackle this task, we introduce an *Auxiliary Database*, which contains a small set of images annotated with fine-grained specie labels. The auxiliary database is defined as $\mathcal{AD} = \{\alpha_1, ..., \alpha_k\}$ and each image α is annotated with a specie label. It allows for feature learning and model fine-tuning, which are potential to significantly improve the OSFGIR performance. Referring to [12], OSFGIR defines an one-shot learning problem, *i.e.*, using prior knowledge in a small independent $\mathcal{AD}$ to identify new objects in the large-scale $\mathcal{D}$. It thus corresponds to more realistic settings than fine-grained image retrieval work [49].

3.2 OSFGIR-378K dataset

Among existing image datasets, ImageNet [9] contains many coarse categories like fish, dog, bird, *etc.*, and many fine-grained species. However, ImageNet is designed for image classification and contains a complete training set. Moreover, most of existing baseline deep models are trained on ImageNet, making ImageNet not suitable to serve as a fair benchmark for OSFGIR features and models. Therefore, we collect a new OSFGIR-378K dataset.

Table 1: The summarization of OSFGIR-378K dataset.

Sub-sets	# Species	# Queries	# Images
$\mathcal{D}^{bird}$	200	1,692	24,119
$\mathcal{D}^{car}$	1,715	13,033	136,725
$\mathcal{D}^{food}$	70	7,000	70,000
$\mathcal{D}^{distr}$			70,194
Total ($\mathcal{D}$)	**1,985**	**21,725**	**301,038**
$\mathcal{AD}^{bird}$	362		30,371
$\mathcal{AD}^{car}$	196		16,185
$\mathcal{AD}^{food}$	31		31,000
Total ($\mathcal{AD}$)	**589**		**77,556**

We aim to build a large-scale OSFGIR dataset labeled with variety types of fine-grained species, *e.g.*, both the man-made and natural objects. To make the dataset collection task feasible, we leverage existing FGVC datasets to construct the new OSFGIR-378K dataset. Specifically, OSFGIR-378K contains three sub-sets of coarse object categories and one sub-set of distractors. We denote the four sub-sets containing birds, food, cars, and distractors as $\mathcal{D}^{bird}$, $\mathcal{D}^{food}$, $\mathcal{D}^{car}$, and $\mathcal{D}^{distr}$, respectively. In the following, we give details about the construction of those sub-sets.

There are two datasets for fine-grained bird categorization, *i.e.*, CUB-200-2011 [45], and BirdSnap [5]. BirdSnap contains a larger number of images and species, *i.e.*, 500 species, and 49,829 images in BirdSnap vs. 200 species and 11,788 images in CUB-200-2011, respectively. Note that, the BirdSnap and CUB-200-2011 share 138 common species. Because the images of CUB-200-2011 generally have better quality, we firstly include CUB-200-2011 in the bird sub-set. Then, we put the 138 common species in BirdSnap into the bird sub-set. Finally, we manually delete the noisy images and construct the clean bird sub-set $\mathcal{D}^{bird}$. Therefore, the final $\mathcal{D}^{bird}$ contains the CUB-200-2011 and a part of BirdSnap.

For the car sub-set, there also exist two datasets, *i.e.*, Car196 [28] consisting of 16,185 images of 196 species, and CompCar [50] consisting of 136,725 images of 1,715 species, respectively. As the CompCar contains more images and species than Car196, and it is a clean dataset, we simply treat the CompCar as $\mathcal{D}^{car}$.

There is only one public dataset for fine-grained food categorization, *i.e.*, the food-101 [6]. We thus randomly select 70 species to generate the food sub-set $\mathcal{D}^{food}$. To test the robustness and efficiency of OSFGIR methods, we further collect 70K images as distractors $\mathcal{D}^{distr}$. The detailed descriptions of OSFGIR-378K are summarized in Table 1.

The Auxiliary Database $\mathcal{AD}$ contains three sub-sets: $\mathcal{AD}^{bird}$, $\mathcal{AD}^{food}$, and $\mathcal{AD}^{car}$, respectively. $\mathcal{AD}^{bird}$ contains the rest species in BirdSnap, thus has no overlap with the species in $\mathcal{D}^{bird}$. For the car dataset, we treat the Car196 as $\mathcal{AD}^{car}$. 70 species of food-101 are selected as $\mathcal{D}^{food}$. We hence use the rest 31 species as $\mathcal{AD}^{food}$. The detailed summarization of auxiliary database is shown in Table 1.

The final OSFGIR-378K dataset contains a dataset $\mathcal{D}$ for retrieval and an Auxiliary Dataset $\mathcal{AD}$ for training. Note that, to test the ability of identifying unseen species and simulate a real experimental setting, $\mathcal{D}$ and $\mathcal{AD}$ do not share common species and $\mathcal{AD}$ is smaller.

Hantao Yao[1,2], Shiliang Zhang[3], Yongdong Zhang[1,2], Jintao Li[1], Qi Tian[4]
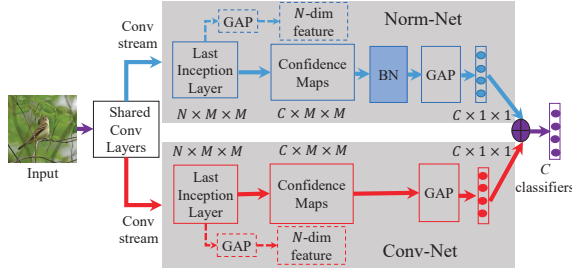

Figure 2: Illustration of Convolutional and Normalization Networks (CN-Nets) fine-tuned on the auxiliary dataset. GAP denotes Global Average Pooling and BN denotes the Batch Normalization.

Figure 3: Images and their confidence maps generated by Norm-Net (second row) and Conv-Net (third row), respectively.

4 PROPOSED APPROACH

One of the key steps in OSFGIR is to learn a discriminative visual descriptor with auxiliary training set. There exist several CNN-based descriptors that have achieved good classification accuracy on FGVC, *e.g.*, Bilinear CNN [32], Spatial Transformer Networks (STN) [22], and CompactBilinear CNN [13]. However, they all have some disadvantages for retrieval task, *e.g.*, time-consuming or hard to extend to unseen data. We propose a novel Convolutional and Normalization Networks (CN-Nets) to generate the image description in Sec. 4.1. With the CN-Nets, we further propose a coarse-to-fine retrieval framework in Sec. 4.2.

4.1 Convolutional and Normalization Networks

CN-Nets is proposed to learn a concise and discriminative representation from image-level labels. It is designed to be more efficient and easy to implement than many FGVC works that generate representations from part labels. As shown in the Fig. 2, CN-Nets takes an image as input and is fine-tuned on the auxiliary dataset in a classification task. It combines outputs from two sub-networks, *i.e.*, Conv-Net and Norm-Net, as the classification result. Conv-Net and Norm-Net share several convolutional and pooling layers, and are designed with different network structures to learn complementary features. We use outputs of their last inception layers to generate features for OSFGIR. In the following, we introduce these two networks and discuss why their features are complementary to each other.

Most of popular networks, such as Alexnet [29], VGG [42], and GoogLeNet [43] feed the extracted feature into the fully-connected layer followed by softmax layer for classification. This setting is proven effective in classification tasks but is hard to interpret and is expensive for training due to the huge number of parameters in fully connected layers. Inspired by [31], we propose Conv-Net, which firstly uses convolutional layers to generate feature maps explicitly corresponding to object categories, then uses Global Average Pooling (GAP) layer to predict the classification score for each category. As shown in Fig. 2, Conv-Net firstly generates C feature maps corresponding to C categories, then computes a C-dim classification score vector with GAP. Because the average response
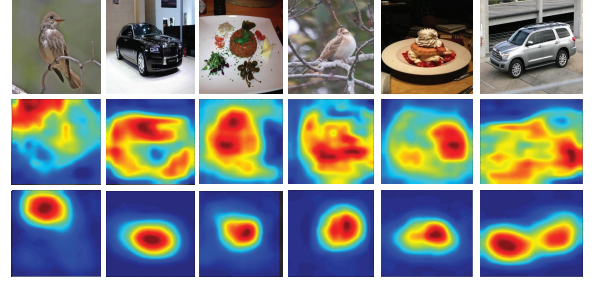
value on each feature map equals to a classification score, we also call the C feature maps as category confidence maps.

Compared with the fully-connected layer, GAP layer also generates the classification score and has the following advantages making it more suited for retrieval task: 1) GAP generates explicit object confidence map, *i.e.*, each feature map denotes the spatial activation for an object category. This makes the network feature easier to interpret. 2) GAP has no parameter to tune, thus avoids overfitting and accelerates the network training and testing. 3) The confidence map reveals the discriminative regions in the input image, thus can be useful for object detection and background elimination. As shown in the third row of Fig. 3, the confidence maps of Conv-Net focus on the foreground objects, *i.e.*, the most discriminative regions in the image.

Based on the Conv-Net, we add *Batch Normalization* (BN) layer between the last convolutional layer and GAP layer to construct the Norm-Net. Given n input images in a mini-batch, the BN layer first collects the activations on each location of a $M \times M$ sized feature map as $\mathcal{B} = \{x_1, x_2, \ldots, x_m\}$, where $m = n \times M \times M$. BN then employs the mini-batch mean $\mu_\mathcal{B}$ and variance $\sigma_\mathcal{B}^2$ to normalize the samples in $\mathcal{B}$, and finally obtains the normalized values $\widehat{x}$. Aiming to make the output of BN represent the identity transform [21] of the input, BN also scales and shifts $\widehat{x}$ by γ and β, respectively. The output of BN y_i in Norm-Net is finally passed to the GAP layers to compute the classifier scores. We summarize the BN algorithm in Algorithm 1. More details of BN can be found in [21].

The second row of Fig. 3 shows that the confidence maps of Norm-Net focus on both the foreground object and the spatial contexts, thus are largely different from the confidence maps of Conv-Net. Our experimental results also validate that the features generated by Conv-Net and Norm-Net are complementary to each other. For example, on CUB-200-2011 dataset [45], the individual classification accuracies of Conv-Net and Norm-Net features are 83.8% and 82.3%, respectively. Combining these two features substantially boosts the accuracy to 85.1%. More extensive experiments about these two features can be found in Sec. 5.2.

Here, we briefly analyze the reason why the Norm-Net features focus on more spatial contexts. With the BN input $\mathcal{B} = \{x_1, x_2, \ldots, x_m\}$ and output $\{y_1, y_2, \ldots, y_m\}$, the backpropagation of the loss l, as well as the computation of gradients with respect to the BN parameters can be summarized as, *i.e.*,

Algorithm 1 Batch Normalizing Transform, applied to activation x over a mini-batch.

1: **Input**: Values of x over a mini-batch: $\mathcal{B} = \{x_1, x_2, \ldots\ldots, x_m\}$; Parameters to be learned: γ, β
2: **Output**: $\{y_i = BN_{\gamma,\beta}(x_i)\}$
3: $\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$
4: $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$
5: $\widehat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$
6: $y_i \leftarrow \gamma \widehat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i)$

$$\frac{\partial l}{\partial \widehat{x_i}} = \frac{\partial l}{\partial y_i} \cdot \gamma, \tag{1}$$

$$\frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^{m} \frac{\partial l}{\partial \widehat{x_i}} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2} (\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}, \tag{2}$$

$$\frac{\partial l}{\partial \mu_{\mathcal{B}}} = \sum_{i=1}^{m} \frac{\partial l}{\partial \widehat{x_i}} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_{\beta})}{m}, \tag{3}$$

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \widehat{x_i}} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial l}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}, \tag{4}$$

where Eq. (4) affects the original activation x_i. The first and third terms in Eq. (4) do not affect x_i individually. For the first term of Eq. (4), because of GAP, $\frac{\partial l}{\partial y_i}$ is identical for each x_i in the same confidence map, so the first term of Eq. (4) has no effect on x_i. Among the second term in Eq. (4), although the sign for $\frac{\partial l}{\partial \sigma_{\mathcal{B}}^2}$ is unknown, it has a same effect on all x_i. Therefore, the second term outputs large values when the gap between x_i and $\mu_{\mathcal{B}}$ is large, vice versa. It is easy to infer that, the back propagation would make larger changes to x_i in this case and would finally make every x_i show similar values close to $\mu_{\mathcal{B}}$. In other words, BN suppresses the highly activated locations on a feature map and encourages the rests.

As a result, Norm-Net tends to activate larger image regions compared with Conv-Net, which only focuses on the discriminative regions on the object. Because the Norm-Net is trained to minimize the classification error, it has potential to discover more helpful contextual cues in the image. Consequently, Conv-Net focuses on the discriminative region, and Norm-Net "see" more contexts.

By combining features from Conv-Net and Norm-Net, we obtain a more powerful CN-Nets feature. As shown in Fig. 2, given an input image region, Norm-Net and Conv-Net generate the feature maps of the last inception layer, denoted as $\mathcal{X} = \{\mathcal{X}_i\}, i = 1, \ldots\ldots, \mathcal{N}$, where $\mathcal{N}$=1024 in this paper. The $\mathcal{X}_i$ is a 2D tensor denoting the responses of the i-th channel. Conv-Net and Norm-Net then generate two $\mathcal{N}$-dim feature vectors with GAP operation. We denote the features of Conv-Net and Norm-Net as $\mathbf{f}_{Conv}$ and $\mathbf{f}_{Norm}$, respectively. The final CN-Nets feature $\mathbf{f}_{CN}$ is generated by concatenating the Conv-Net feature and Norm-Net feature, *i.e.*,

$$\mathbf{f}_{CN} = [\mathbf{f}_{Conv}, \mathbf{f}_{Norm}], \mathbf{f}_{CN} \in \mathbf{R}^{2048}. \tag{5}$$
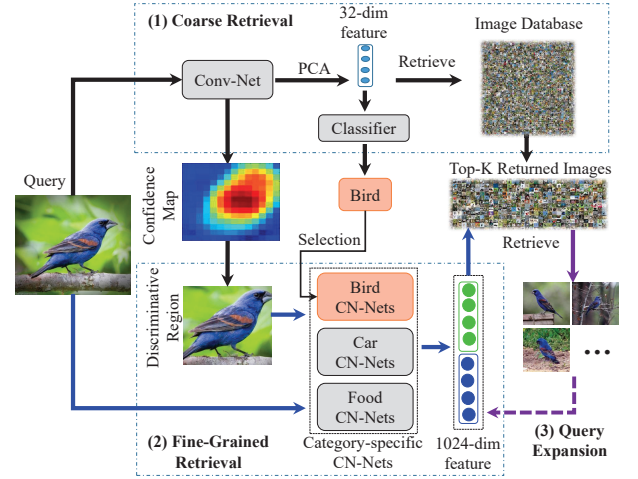

Figure 4: Our coarse-to-fine framework for online OSFGIR.

4.2 Coarse-to-fine retrieval framework

The fine-grained species in image database belong to different coarse object categories. This naturally leads to a coarse-to-fine retrieval framework, which first quickly retrieves images with the same coarse object to narrow-down the search space, then finds the fine-grained species. As shown in Fig. 4, the coarse-to-fine framework consists of three stages, *i.e.*, coarse retrieval, fine-grained retrieval, and query expansion. Both the coarse and fine stages are targeted to build an accurate and efficient retrieval system.

Coarse Retrieval retrieves images containing the same coarse object with the query. Because this is an easier task, we extract a compact efficient feature. Specifically, during off-line indexing, we use Conv-Net with the input size of 224×224 to generate the image feature $\mathbf{f}_{Conv}^d \in \mathbf{R}^{1024}$ for the database image d. To accelerate the similarity comparison, we reduce the dimensionality of $\mathbf{f}_{Conv}^d$ to 32-dim with PCA [11], and apply L_2-normalization to each feature. Moreover, we generate the confidence map for d, which is then used to locate the foreground region. As discussed in Sec. 4.1, Conv-Net outputs $C \times M \times M$ confidence maps for each input image, where C is the number of species in auxiliary dataset, and $M \times M$ is the size of confidence map. Among the C confidence maps, we select the one with the maximum average activation. The selected confidence map is resized to the same size of the input image, and is normalized by dividing the maximum response value on it. We denote the selected confidence map for database image d as I^d.

During online retrieval, we process the query q in the same way to obtain its feature $\mathbf{f}_{Conv}^q$ and confidence map I^q. Euclidean distance is then computed between the query feature $\mathbf{f}_{Conv}^q$ and all database features to obtain the Top-K similar images, *e.g.*, K=10,000. Besides that, we estimate the object category of the query specie. For OSFGIR-378K dataset, a three-way SVM classifier is trained with the 32-dim feature based on the auxiliary database. We assume that the query images do not contain distractors, thus use a three-way classifier and ignore the distractors. As shown in Fig. 4, the coarse retrieval outputs 1) the Top-K similar images, 2) the confidence map, and 3) the coarse category label of the query image.

Hantao Yao[1,2], Shiliang Zhang[3], Yongdong Zhang[1,2], Jintao Li[1], Qi Tian[4]

Fine-Grained Retrieval performs fine-grained identification on the Top-K images. As illustrated in Fig. 4, we employ the complete CN-Nets to extract features $\mathbf{f}_{CN}$ and additionally fuse features from both the discriminative region and the entire image to acquire more discriminative power. The confidence map $\mathcal{I}$ guides the discriminative region selection. $\mathcal{I}$ is first converted to a binary image with threshold t, e.g., $t = 0.5$. Then, we employ the connected region analysis [10] to remove the small regions and select one dominant region. The discriminative region is generated by cropping the dominant region with the minimum enclosing rectangle. As illustrated in Fig. 4, the discriminative region covers most of the foreground object.

During online retrieval, we first extract the discriminative regions of query and database images, then extract their CN-Nets features. To ensure the discriminative power of CN-Nets feature, we fine-tune a category-specific CN-Nets for each coarse category on the auxiliary dataset. For instance, if an image is identified to bird category, we employ the CN-Nets fine-tuned on $\mathcal{AD}^{bird}$ for feature extraction. The predicted category label in coarse retrieval hence selects the proper category-specific CN-Nets for feature extraction. This strategy leads to two feature vectors, i.e., $\mathbf{f}_{CN}^{img} \in \mathbf{R}^{2048}$ on the entire image and $\mathbf{f}_{CN}^{reg} \in \mathbf{R}^{2048}$ on the discriminative region, respectively.

We reduce dimensionality of $\mathbf{f}_{CN}^{img}$ and $\mathbf{f}_{CN}^{reg}$ to 512-dim with PCA, respectively. The final feature $\mathbf{f}$ for fine-grained retrieval is generated by concatenating these two 512-dim features, i.e.,

$$\mathbf{f} = [\mathbf{f}_{CN}^{img}, \mathbf{f}_{CN}^{reg}], \mathbf{f} \in \mathbf{R}^{1024}. \tag{6}$$

The Top-K images are finally ranked using $\mathbf{f}$ and Euclidean distance. Note that, in our implementation, the discriminative regions and features of database images can be extracted and stored off-line, thus their computations do not degrade the online efficiency.

Query Expansion (QE) strategy is designed to further improve the retrieval accuracy. The fine-grained retrieval effectively ranks some positive images at the top of returned image list. We simply apply average pooling on the features of Top-$\mathcal{K}$ ($\mathcal{K}$=5) returned images to calculate a new descriptor. A new round of retrieval is performed with the new descriptor to update the original ranking list. In the experimental part, we will evaluate the effectiveness and efficiency of this coarse-to-fine retrieval framework.

5 EXPERIMENTS

5.1 Implementation Details

We use Caffe [25] for CNN model training and fine-tuning. The CN-Nets are firstly initialized with the model introduced in [1], then are modified based on the Batch-Normalized Convolutional Networks described in [21]. Conv-Net in coarse retrieval is fine-tuned on the complete $\mathcal{AD}$ with C=589. The three category-specific CN-Nets in fine-grained retrieval are fine-tuned on $\mathcal{AD}^{bird}$, $\mathcal{AD}^{food}$, and $\mathcal{AD}^{car}$ with $C = 362, 196$, and 31, respectively. All experiments are conducted on a server equipped with Intel Xeon E5-2650 CPU and Tesla K40 GPU. Experiments in Sec. 5.2 are conducted on CUB-200-2011 [45].

We use the widely-used Mean Average Precision (MAP) to evaluate the performance on OSFGIR-378K. Classification accuracy is used when we test on classification tasks.
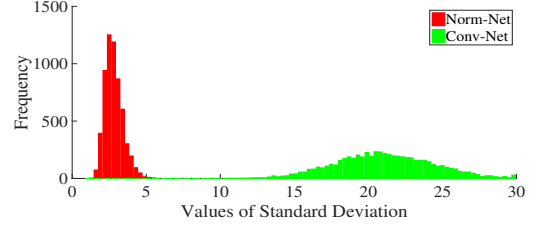

Figure 5: Standard deviation histograms of response values in confidence maps produced by Norm-Net and Conv-Net on all training images, respectively.

Table 2: The classification accuracy of features extracted by CN-Nets with different number of shared layers. "Time" is the feature extraction time. "C_n" and "I_m" denote the n-th convolutional layer and m-th inception layer, respectively. "C_n-C_m" means C_n to C_m layers are shared. "-" denotes no layer is shared.

Shared Layers	-	C_1-C_2	C_1-I_{3b}	C_1-I_{4c}	C_1-I_{4e}
Acc.(%)	83.9	84.5	85.1	83.9	84.4
Time(ms)	50	41	34	29	26

5.2 Analysis and Discussions on CN-Nets

Examples in Fig. 3 demonstrate that Conv-Net focuses on the discriminative object regions, while Norm-Net covers more spatial contexts. To verify this observation, we calculate the standard deviation (std) histogram of the response values in confidence maps on all training images, and compare the results of Conv-Net and Norm-Net in Fig. 5. Higher standard deviation (std) denotes more unbalanced activations, i.e., the size of highly activated regions would be smaller, and lower standard deviation (std) denotes that the confidence map contains similar activation values. As shown in the figure, the confidence maps of Conv-Net show substantially larger standard deviations than those of Norm-Net. This means that Conv-Net tends to activate smaller discriminative regions, while Norm-Net covers larger regions.

We conduct another classification experiment to show the complementarity of Conv-Net and Norm-Net features. It is easy to observe from the results in Fig. 6 that, Conv-Net features get higher accuracies than Norm-Net features because they are extracted from more discriminative regions. It is also clear that, the combined Conv-Net and Norm-Net feature, i.e., the CN-Nets feature, gets the best accuracy. The figure also shows that, although BN layer is inserted at the end of Norm-Net, it affects the learned features in preceding layers, e.g., inception5a and 5b.

As shown in Fig. 2, Conv-Net and Norm-Net share several layers in CN-Nets. Table 2 shows the effects of shared layers on the discriminative power and extraction time of CN-Nets feature. It can be observed that, sharing more layers between Conv-Net and Norm-Net actually improves both the efficiency and discriminative power of the CN-Nets feature. For example, by sharing the C_1-I_{3b} layers, CN-Nets feature obtains the highest classification accuracy and only needs $34ms$ for feature extraction. This validates that the structure of CN-Nets is efficient and reasonable.
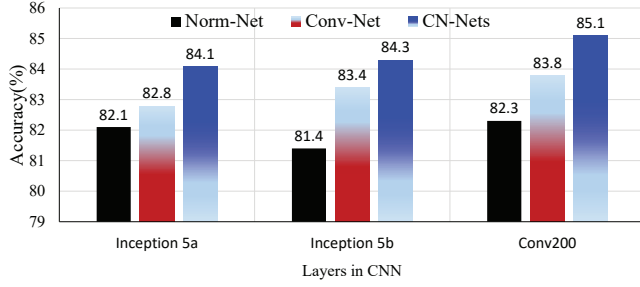
Figure 6: The classification accuracy of features extracted by Conv-Net, Norm-Net, and CN-Nets, respectively. Three groups of features are extracted on "Inception5a", "Inception5b","conv200" layers, respectively. Details of these layers can be found in [43].

5.3 Comparison with Other Deep Features

To test the discriminative power of CN-Nets feature, we compare it with deep features extracted with recent deep learning models, *i.e.,* Res-152 [19], BilinearCNN [32], and Spatial Transformer Networks (STN) [22] in FGVC tasks on CUB-200-2011 [45] and Car196 [28], respectively. We summarize the results in Table 3.

Among the compared features, CN-Nets feature achieves the best classification accuracy. *E.g.*, single CN-Nets outperforms both the recent BilinearCNN and STN on CUB-200-2011 and Car196. We also compare with an attribute-based method [35] and a part-based method [27], which use extra cues for model training and currently report the highest classification accuracies on CUB-200-2011 and Car196, respectively. As shown in Table 3, by simply concatenating the features from two CN-Nets, fused CN-Nets outperforms both of these works. The above comparisons clearly show CN-Nets is a powerful feature extractor for fine-grained species.

It is also necessary to point out that, CN-Nets is the fastest deep feature extractor in Table 3. CN-Nets only needs $34ms$ to extract the feature. The Fused CN-Nets needs about $60ms$. They are both faster that STN and BilinearCNN, which cost about $80ms$ and $100ms$, respectively. CN-Nets is also faster than attribute-based [35] and part-based [27] methods, which need to firstly localize the parts and then generate the description for each part. Therefore, the CN-Nets is also an efficient deep network.

Finally, CN-Nets is also easier to train on new datasets, because it uses simple network structure and involves fewer parameters by removing the fully connected layers. Consequently, CN-Nets features better scalability than more complicated networks like STN and BilinearCNN. As a consequence, the CN-Nets is better suited for the proposed OSFGIR task.

5.4 OSFGIR Performance

The coarse retrieval extracts a 32-dim feature and trains a three-way category classifier for category-specific CN-Nets selection. We thus first test the validity of this compact feature and the 3-way classifier. Table 4 shows that, the original 1024-dim Conv-Net feature performs well in identifying the coarse object categories. Further applying PCA to reduce the dimensionality does not degrade the accuracy, *e.g.,* 32-dim feature achieves a slightly better accuracy of

Table 3: Comparison of classification accuracy (%) with other deep features. "Time" denotes the feature extraction time. "Complexity" measures the complexity of deep model training on a new dataset. "Fused CN-Nets" concatenates the features of two CN-Nets sharing C_1-I_{3b} and C_1-I_{4e} layers, respectively. "CN-Nets" denotes the network sharing C_1-I_{3b} layers (refer to Table 2).

Methods	Time(ms)	Complexity	CUB200	Car196
Res-152 [19]	120	Easy	79.8	90.45
STN [22]	80	Hard	84.1	-
Bilinear [32]	100	Medium	84.1	91.3
Recent Report	$\geq$ 100		85.5 [35]	92.6 [27]
CN-Nets	34	Easy	85.1	92.39
Fused CN-Nets	60	Easy	**85.65**	**93.06**

Table 4: The accuracy of the 3-way classifier with different feature dimensionality.

Dim	1024	512	128	64	32
Acc.(%)	97.22	97.12	97.25	97.56	97.67

Table 5: The retrieval performance on OSFGIR-378K. "Dim" denotes the feature dimensionality. "Coarse" and "C-to-F" denote the coarse retrieval and coarse-to-fine retrieval, respectively.

Methods	Net. input size	Dim	MAP(%)
Coarse	224×224	1024	11.27
Coarse	224×224	32	9.18
C-to-F_$\mathbf{f}_{CN}^{img}$	448×448	2048	22.19
C-to-F_$\mathbf{f}_{CN}^{img}$	448×448	512	22.22
C-to-F_Unified	448×448	2048	16.90

97.67%. Therefore, the 32-dim feature and the classifier are effective for coarse retrieval and CN-Nets selection.

We then discuss the accuracy of the coarse-to-fine retrieval framework. As shown in Table 5, coarse retrieval achieves the Mean Average Precision (MAP) of 9.18%. The coarse-to-fine retrieval with the 2048-dim feature $\mathbf{f}_{CN}^{img}$ significantly boosts the performance from 9.18% to 22.19%. After reducing the feature dimensionality to 512, the retrieval performance increases to 22.22%. The large improvement over coarse retrieval demonstrates the importance of fine-gained retrieval stage. The retrieval results with complete feature $\mathbf{f}$ concatenating $\mathbf{f}_{CN}^{img}$ and $\mathbf{f}_{CN}^{reg}$ will be presented in Sec. 5.5.

During fine-grained retrieval, we use category-specific CN-Nets for feature extraction. To show the validity of this strategy, we compare with the performance of a Unified-CN-Nets fine-tuned on the complete auxiliary dataset. As shown in Table 5, category-specific CN-Nets plus the accurate 3-way category classifier achieves a significantly better performance, *e.g.,* 22.19% *vs* 16.90% of Unified-CN-Nets. The above experiments clearly show the validity of our coarse-to-fine retrieval framework.

Hantao Yao[1,2], Shiliang Zhang[3], Yongdong Zhang[1,2], Jintao Li[1], Qi Tian[4]

Table 6: The time complexity of our OSFGIR system. ∗ means the running-time is evaluated on GPU K40.

Stages	Operations	Time(ms)
Coarse	Feature Extraction	21*
Retrieval	Retrieval	73.1
	Classifier	8
Fine-Grained	Feature Extraction	68*
Retrieval	Retrieval	100
Total		270.1

Table 7: Comparison with other methods on OSFGIR-378K.

Methods	Net. input size	Dim	MAP(%)
VGG19+VLAD [51]	224×224	512	4.14
VGG19+MAC [44]	224×224	512	9.6
VGG19+NeuralCode [3]	224×224	4096	10.17
VGG19+CROW [26]	224×224	512	10.73
Res-152 [19]	224×224	1024	12.13
L2_$\mathbf{f}_{CN}$	224×224	2048	12.93
GoogLeNet	448×448	1024	14.9
Res-152	448×448	1024	14.7
CompactBilinear CNN [13]	448×448	8192	15.06
L2_$\mathbf{f}_{CN}$	448×448	2048	16.90
C-to-F_**f**	448×448	1024	**23.68**
C-to-F_**f**+QE	448×448	1024	**26.31**

The online querying time consists of five operations, *i.e.*, two feature extractions, two retrievals, and one classification. We finally analyze the time complexity and summarize the details in Table 6. As shown in the table, our total online query time is about $270\,ms$. The coarse retrieval returns results in less than $100\,ms$ to reduce the search space. This allows the online retrieval to finish retrieval in less than $170\,ms$ using more powerful features.

5.5 Comparison with Existing Methods

To further test the performance of our OSFGIR system, we compare with recent image and instance retrieval methods including NeuralCode [3], CROW [26], CNN+VLAD [51], MAC [44], and CompactBilinear CNN [13]. To make the comparison fair, we fine-tune the network of each method on the auxiliary dataset. For NerualCode, we use the output of the $fc6$ layer in VGG19 as the feature. For the method of [44, 51], we employ the output of the last convolutional layer, *i.e.*, $conv5\_4$ to extract feature. The CROW is implemented based on the *pool5* of VGG19. The comparsions with different network input sizes are summarized in Table 7.

L2_$\mathbf{f}_{CN}$ denotes directly using Euclidean distance and the 2048-D original CN-Nets feature extracted from the entire image for retrieval. The comparison shows our work outperforms existing retrieval methods by large margins, *e.g.*, our work achieves the MAP of 23.68% using the final 1024-dim CN-Nets feature **f**. This is significantly better than 14.7% of Res-152, and 15.06% of CompactBilinear CNN [13].

Referring to the results in Table 5, we can infer that the final 1024-dim CN-Nets feature **f** boosts the retrieval performance from 22.22% to 23.68%. Additionally, applying QE further improves the retrieval performance to 26.31%. Therefore, we can conclude that,

Figure 7: Several examples of Top-5 retrieved images of our method (first row) and CompactBilinearCNN [13] (second row). Green solid and red dashed bounding boxes denote true positive and false positive, respectively.

CN-Nets is a powerful feature extractor, and our coarse-to-fine retrieval work is effective and efficient for OSFGIR. Therefore, this work proposes an effective and efficient solution to the challenge of large-scale fine-grained identification of unseen objects. Retrieval examples of our work can be found in Fig. 7, where our method substantially outperforms the CompactBilinerCNN [13].

6 CONCLUSION

This paper presents a novel OSFGIR approach to tackle the challenging large-scale fine-grained identification of unseen objects. OSFGIR aims to return a ranked list of images containing the identical fine-grained specie in the query from a large-scale dataset. We first define the OSFGIR problem and construct a OSFGIR-378K dataset containing about 378K images and 1,985 fine-grained species. To extract discriminative descriptors for the fine-grained species, we propose the Convolutional and Normalization Networks (CN-Nets). CN-Nets conducts one-shot learning on the auxiliary dataset to extract two complementary deep features as the representation of target testing data. A coarse-to-fine retrieval framework is hence proposed to chase a reasonable trade-off between efficiency and accuracy. Experiments on OSFGIR-378K show that our descriptor and retrieval framework achieve significantly better performance than existing FGVC and image retrieval methods. Further works will be conducted to explore more efficient one-shot learning algorithms to optimize both the feature extraction and indexing modules.

REFERENCES

[1] https://github.com/lim0606/caffe-googlenet-bn.
[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*. IEEE, 2016.
[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, pages 584–599. Springer, 2014.
[4] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, pages 955–962. IEEE, 2013.
[5] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, pages 2019–2026. IEEE, 2014.
[6] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014.
[7] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014.
[8] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328. IEEE, 2013.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
[10] L. Di Stefano and A. Bulgarelli. A simple and efficient connected components labeling algorithm. In *ICIAP*, pages 322–327. IEEE, 1999.
[11] G. H. Dunteman. *Principal components analysis*. Number 69. Sage, 1989.
[12] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.
[13] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*. IEEE, 2016.
[14] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, pages 1713–1720. IEEE, 2013.
[15] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Local alignments for fine-grained categorization. *IJCV*, pages 1–22, 2014.
[16] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Subset feature learning for fine-grained category classification. In *CVPR*, number IEEE, 2015.
[17] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407. Springer, 2014.
[18] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*. Springer, 2016.
[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*. IEEE, 2016.
[20] S. Huang, Z. Xu, D. Tao, and Y. Zhang. Part-stacked cnn for fine-grained visual categorization. In *CVPR*. IEEE, 2016.
[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
[22] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2008–2016, 2015.
[23] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search-extended version. 2008.
[24] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.
[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
[26] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*. Springer, 2016.
[27] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *CVPR*, pages 5546–5555. IEEE, 2015.
[28] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561. IEEE, 2013.
[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
[30] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, pages 1666–1674. IEEE, 2015.
[31] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.
[32] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*. IEEE, 2015.
[33] J. Liu and P. N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *ICCV*, pages 2520–2527. IEEE, 2013.
[34] J. Liu, Y. Li, and P. N. Belhumeur. Part-pair representation for part localization. In *ECCV*, pages 456–471. Springer, 2014.
[35] X. Liu, J. Wang, S. Wen, E. Ding, and Y. Lin. Localizing by describing: Attribute-guided attention localization for fine-grained recognition. In *NIPS*, 2016.
[36] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
[37] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. IEEE, 2007.

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
[39] A. Salvador, X. Giró-i Nieto, F. Marqués, and S. Satoh. Faster r-cnn features for instance search. In *CVPRW*. IEEE, 2016.
[40] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR*, pages 806–813. IEEE, 2014.
[41] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*. IEEE, 2015.
[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*. IEEE, 2015.
[44] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016.
[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
[46] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo. Deep joint task learning for generic object extraction. In *NIPS*, pages 523–531, 2014.
[47] X.-S. Wei, C.-W. Xie, and J. Wu. Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition. In *NIPS*, 2016.
[48] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*. IEEE, 2015.
[49] L. Xie, J. Wang, B. Zhang, and Q. Tian. Fine-grained image search. *TMM*, 17(5):636–647, 2015.
[50] L. Yang, P. Luo, C. Change Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981. IEEE, 2015.
[51] J. Yue-Hei Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval. In *CVPR*, pages 53–61. IEEE, 2015.
[52] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *CVPR*. IEEE, 2016.
[53] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, pages 834–849. Springer, 2014.