Mining on Manifolds: Metric Learning without Labels

Ahmet Iscen[1] Giorgos Tolias[1] Yannis Avrithis[2] Ondřej Chum[1]
[1]VRG, FEE, CTU in Prague [2]Inria Rennes

Abstract

In this work we present a novel unsupervised framework for hard training example mining. The only input to the method is a collection of images relevant to the target application and a meaningful initial representation, provided e.g. *by pre-trained CNN. Positive examples are distant points on a single manifold, while negative examples are nearby points on different manifolds. Both types of examples are revealed by disagreements between Euclidean and manifold similarities. The discovered examples can be used in training with any discriminative loss.*

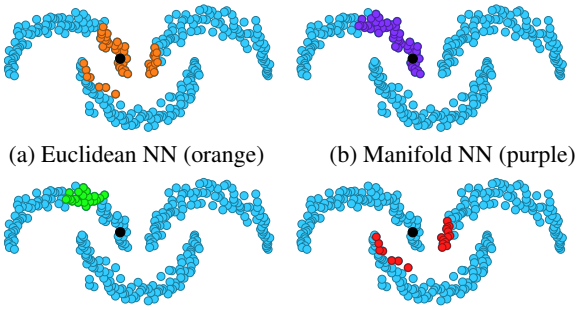
The method is applied to unsupervised fine-tuning of pre-trained networks for fine-grained classification and particular object retrieval. Our models are on par or are outperforming prior models that are fully or partially supervised.

1. Introduction

The success of deep neural networks on large-scale problems has been first demonstrated on the task of supervised classification [26]. It was shown that embeddings, typically provided by the convolutional layers of a network, are applicable beyond classification tasks. These tasks include particular object retrieval [15], local descriptor learning [17], ranking [62], and nearest-neighbor regression [6]. A common practice is to start with a pre-trained network [50, 55, 19] and apply metric learning [9, 62, 18] to fine-tune the network for a particular task.

To improve over the initial network, novel training samples are sought for which the initial network performs poorly. Such training samples are used to re-train the network using loss functions alternative to cross-entropy (*e.g.* contrastive [9], triplet [62] or batch-level [36]). The approaches to obtain relevant training data range from further human supervision [3, 36] to instance clustering [44, 32, 5], exploiting the temporal dimension in video [64, 65], predicting the spatial layout of image patches [11], or using existing computer vision pipelines to match unstructured image collections pairwise [15, 42].

Most recent deep metric learning approaches can learn powerful embeddings but still use class labels. This is unsatisfying not only because we miss the opportunity of learning from unlabeled data, but learned representations of each class are unimodal [44]. Therefore, whatever the loss function [62, 36]) or sampling [18, 34], the problem remains supervised classification. On the other hand, conventional nonlinear dimension reduction or manifold learning methods exploit the manifold structure of the data starting from nearest neighbor graphs and are otherwise unsupervised [56, 45, 7]. However, most do not learn an explicit mapping from the input to the embedding space and have difficulties in generalizing to new data.



(a) Euclidean NN (orange) (b) Manifold NN (purple)

(c) Hard positives (green) (d) Hard negatives (red)

Figure 1. Given an anchor point (black) and its k nearest Euclidean (NN_k^e) and manifold (NN_k^m) neighbors in a dataset, we choose positive samples as $\text{NN}_k^m \setminus \text{NN}_k^e$, and negative as $\text{NN}_k^e \setminus \text{NN}_k^m$. Data is unlabeled and the selection is fully unsupervised, including anchors.

We attempt to bridge this gap in this work. In particular, we propose a novel method of hard training sample mining in a fully unsupervised manner, simply from an unordered collection of images *relevant* to the final task. We observe that a similarity between two images is improved by considering all, even unlabelled, available data. In particular, we exploit similarity measured on a manifold estimated by a random walk process [21].

The learning starts from the initial representation space of unlabeled data. Given an anchor point that is part of the data, neighbors on the manifold that are not Euclidean neighbors are considered positive samples. In the new learned representation space, the positive sample should be attracted to the anchor to reflect the similarity measured on the manifold. Conversely, Euclidean neighbors that are not neighbors on the manifold are considered negative and should be repelled. The idea is illustrated in Figure 1.

The advantage of learning a new representation over using the estimated manifold similarity is twofold. First, estimating the manifold similarity has additional computational and memory requirements at query time. Second, we show that the novel embedding generalizes better not only to previously unseen queries, but also to unseen datasets.

We apply the proposed method to fine-grained classification and to particular object retrieval. Our models obtained by unsupervised fine-tuning of pre-trained networks are on par or are outperforming prior models that are human supervised or use additional domain-specific expertise.

The paper is organized as follows. Section 2 discusses related work. Sections 3 and 4 present our learning method and applications, respectively. Experiments are given in Section 5 conclusions are drawn in Section 6.

2. Related work

This section contains a brief overview of related work on metric learning, embeddings for instance retrieval and representation learning without human labeled data.

Metric learning. Recent approaches based on low-dimensional CNN-based embeddings achieve promising results on this task [47, 36]. A key ingredient of some approaches is *hard negative mining*, which comes from the days of SVMs in object detection [13]. This principle has been known much earlier as *bootstrapping* [54]. Instead of sampling all negative instances for an anchor point, the most challenging negative instances are mined which finally offer faster but equally accurate learning. However, this process is not trivial. Simple hard negative mining from a different class label might corrupt the training process due to mislabeled images [47, 33].

Schroff *et al*. [47] use triplet loss and propose *semi-hard* mining in an attempt to solve this problem. They sample negatives within the batch, such that they are close to the anchor point but further away from positives. This concept is widely adapted in other metric learning approaches [38, 36]. Wu *et al*. [67] improve it by uniformly sampling negative instances weighted by their distance. This allows them to sample points from various regions of the feature space instead of certain clusters.

Other methods, such as *lifted-structure* [36] and n-pair loss [52], focus on the loss function and define constraints on the pairwise distances of all points in a batch. More recently, several methods try to optimize the learned embedding based on the global distribution of the data. Song *et al*. [53] try to optimize the clustering quality. Harwood *et al*. [18] combine triplet loss and global loss using an efficient hard mining procedure. All these methods use class labels during sampling.

Particular object retrieval is another application where descriptors are trained similarly to metric learning. However, class labels do not usually exist as it is intractable to enumerate all possible instances or their viewpoints. Traditional instance search algorithms use hand-designed descriptors [51, 39, 57, 25], but recent advances show the interest of feature learning [42, 15]. Transfer learning from category-level classification is one case [43]. Labeling at landmark level has been attempted as well, treating this task as classification, but the labels are quite noisy [3].

The state-of-the-art approaches start with an off-the-shelf network, and fine-tune it with algorithmic supervision [42, 15]. They make use of geometric matching to mine matching and non-matching image pairs. They involve local feature extraction and require an expensive pre-processing of data. Furthermore, they assume that the training set contains landmarks and buildings that perform well with geometric matching of local features. We make no such assumptions nor require additional computer vision system to perform the mining. Our only assumption is that there are multiple object instances in the training set.

Incomplete, noisy, or unavailable labels are handled in various ways in the literature. In the contrastive loss paper of Hadsell *et al*. [16], the authors show that it is possible to separate different categories to different subspaces just by assigning the Euclidean nearest neighbors as positive training instances. In recent works, the labeling is guided by the information of different modalities or the data collection process. Arandjelović *et al*. [1] learn visual descriptors for location recognition by assigning labels based on the GPS location of each image. Wang and Gupta [64] sample frames from videos and assume that frames from same videos will be positive to each other. Isola *et al*. [22] group objects based on their co-occurence within the same spatial or temporal context. Similarly, learning from the spatial arrangement of the patches within an image is shown feasible [10, 35]. Finally, other cases utilize different modalities, such as learning visual descriptors from text information [14], or learning audio descriptors from unlabeled videos [2]. By contrast, we make no assumptions on the available modalities or contextual information.

Unsupervised methods and manifold learning. There are very few methods on deep metric learning that are unsupervised. Examples are two methods on learning fine-grained similarity by exploiting mutual proximity [6] and ranking [5] in the Euclidean space. Both utilize some form of clustering and splitting the training set in different groups, which is an artificial constraint, and none takes the underlying manifold structure into account. Our method is conceptually simpler and compatible with any loss function requiring positive/negative samples.

The work of Li *et al*. [30] is similar to ours in following a graph-based mining approach. In our comparisons, we show that choosing hard examples is essential and results in better performance for our approach. Recently, Pai *et*

al. [37] learn the manifold structure by encouraging pairs of points to have a Euclidean distance in the embedding space equal to the geodesic distance on the graph. We use a more relaxed objective that is compatible to most metric learning formulations and loss functions, and a manifold similarity that is more scalable than the geodesic distance.

Embeddings are learned to approximate ranking on manifolds with *fast spectral ranking* [20]. However, this approach is dataset specific and does not generalize to unseen images. Improvements on this aspect are introduced by *iterative manifold embedding* [68]. Nevertheless, the extension can handle a query image, or potentially a small set of unseen images, while a larger dataset increase significantly affects the manifold structure.

3. Method

In this section we describe our learning problem, briefly discuss the required background, and present our unsupervised training subset selection and the training process.

3.1. Problem formulation

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{X}$ be an unordered and unlabeled collection of items, where $\mathcal{X}$ is the original *input* space of all possible items. Depending on the application, an item corresponds to an image, video, image region, local patch, *etc*. Function $f(\cdot;\theta) : \mathcal{X} \to \mathbb{R}^d$ maps an item $\mathbf{x}$ to a vector $\mathbf{z} = f(\mathbf{x}, \theta)$ in a d-dimensional *embedding* space, where θ is a set of parameters to be learned.

The input items are represented by a set of features $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\} \subset \mathcal{Y}$, where $\mathcal{Y}$ is a *feature* space and $\mathbf{y} = g(\mathbf{x})$ for $\mathbf{x} \in X$. Depending on the application, g may be the identity mapping, *i.e*. learn directly from input space, $f(\cdot;\theta_0)$, *i.e*. learn from the same model with initial parameters θ_0, or a different model. An existing model may have been supervised (in any way) or not.

Two items are *matching* if they are considered visually similar, otherwise *non-matching*. Our goal is to learn the model parameters such that matching items are mapped to nearby points in the embedding space, while non-matching items are well separated. This corresponds to a typical metric learning scenario, and common practice is to use manually defined labels in order to construct a training set of matching and non-matching pairs of items [9, 62, 18]. In this work, we only assume that the input items X and their features Y are available at training time.

A training pair is defined w.r.t. to a *reference* (*anchor*) item $\mathbf{x}^r$. A matching pair consists of the anchor and a *positive* item $\mathbf{x}^+$. Similarly a non-matching pair consists of the anchor and a *negative* item $\mathbf{x}^-$. Alternatively, we may use a triplet $(\mathbf{x}^r, \mathbf{x}^+, \mathbf{x}^-)$. Our goal is to mine such training pairs without any supervision [9, 62, 18, 18, 34], without any complementary computer vision system [15, 42] or assumptions on the nature of the training data [64, 65, 11].

3.2. Preliminaries

By $\mathrm{NN}_k^e(\mathbf{y})$ we denote the k *Euclidean nearest neighbors* of $\mathbf{y} \in Y$, *i.e*. the k most similar items in Y according to some *Euclidean similarity* function $s_e : \mathcal{Y}^2 \to \mathbb{R}$. Similarly, by $\mathrm{NN}_k^m(\mathbf{y})$ we denote the k *manifold nearest neighbors* of $\mathbf{y} \in Y$, *i.e*. the k most similar features in Y according to a *manifold similarity* $s_m : Y^2 \to \mathbb{R}$.[1]

Given s_e, we employ a random walk on the Euclidean nearest neighbor graph G to measure the manifold similarity s_m [69]. The graph has Y as nodes. It is is undirected, weighted, represented by sparse symmetric adjacency matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$. Edges correspond to reciprocal k-nearest neighbors, with weights

$$a_{ij} = \begin{cases} s_e(\mathbf{y}_i, \mathbf{y}_j), & \text{if } \mathbf{y}_i \in \mathrm{NN}_k^e(\mathbf{y}_j) \wedge \mathbf{y}_j \in \mathrm{NN}_k^e(\mathbf{y}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

There are no loops in the graph, *i.e*., the diagonal elements of A are zero. Starting from an arbitrary vector $\mathbf{f}^{(0)} \in \mathbb{R}^n$, the random walk for a given feature $\mathbf{y}_i \in Y$ follows the iteration

$$\mathbf{f}_i^{(t)} = \alpha \hat{A} \mathbf{f}^{(t-1)} + (1 - \alpha)\mathbf{e}_i, \tag{2}$$

where $\mathbf{e}_i$ is the i-th column of an $n \times n$ identity matrix, $\hat{A} = D^{-1/2} A D^{-1/2}$ with $D = \mathrm{diag}(A\mathbf{1})$ being the degree matrix, and $\alpha \in [0, 1)$. Iteration (2) converges to the solution $\mathbf{f}_i^\star$ of the linear system

$$(I - \alpha \hat{A})\mathbf{f} = (1 - \alpha)\mathbf{e}_i. \tag{3}$$

Following Iscen *et al*. [21], we use the conjugate gradient method to solve this system efficiently in practice, since $I - \alpha\hat{A}$ is positive-definite. We define the *manifold similarity*

$$s_m(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{f}_i^\star(j), \tag{4}$$

i.e., the j-th element of $\mathbf{f}_i^\star$. Observe that s_m is symmetric because in fact $s_m(\mathbf{y}_i, \mathbf{y}_j)$ is the (i, j)-element of matrix $(1 - \alpha)(I - \alpha\hat{A})^{-1}$, which is symmetric. This matrix is dense but we never compute it; we only compute its columns as needed. For instance, given $\mathbf{y}_i \in Y$, its manifold nearest neighbors $\mathrm{NN}_k^m(\mathbf{y}_i)$ are the k maximum elements of the i-th column.

3.3. Manifold-guided selection of training samples

We are guided by the manifold similarity to select training samples. In particular, we exploit the differences between Euclidean and manifold nearest neighbors of anchor items. We first describe the selection of positives and negatives given an anchor. Then we discuss anchor selection.

[1] In fact, s_e can be any symmetric function but is only a function of two elements of $\mathcal{Y}$; while s_m is a function of two elements in Y but also of the entire set Y.

Positives. Given an anchor item $\mathbf{x}^r$ and the corresponding feature $\mathbf{y}^r = g(\mathbf{x}^r)$, which is used as query, we choose as positives the items that correspond to the manifold nearest neighbors of $\mathbf{y}^r$ that are not Euclidean neighbors. Such difference provides evidence of a matching item that is not retrieved well in the feature space, as illustrated in Figure 1(c). In the embedding space, positives should be attracted to the anchor so that Euclidean and manifold neighbors agree.

We therefore simply compare the sets $\text{NN}_k^m(\mathbf{y}^r)$ and $\text{NN}_k^e(\mathbf{y}^r)$ and select the input items that correspond to their set difference

$$P^+(\mathbf{x}^r) = \{\mathbf{x} \in X : g(\mathbf{x}) \in \text{NN}_k^m(\mathbf{y}^r) \setminus \text{NN}_k^e(\mathbf{y}^r)\} \quad (5)$$

as the *positive pool* of anchor $\mathbf{x}^r$. The value of k controls the visual diversity of positives, with larger values giving the *harder* examples. In practice, we maintain the pool ordered by descending manifold similarity, so that we may truncate by keeping the examples with highest confidence.

Mining hard positive examples, in contrast to negatives, is not common. Apart from positives being fewer than negatives, this is due to the large intraclass variability; it may result in positives that are too hard to learn. It can be achieved in cases with known geometry of the scene such that extreme cases are avoided [48, 42]. In our case, the hardness is controlled by the manifold similarity according to the current model g, so drifting into very tough examples is less likely.

Negatives. Similarly, and as illustrated in Figure 1(d), we choose as negatives the items that correspond to the Euclidean nearest neighbors of $\mathbf{y}^r$ that are not manifold neighbors. Such difference provides evidence of a non-matching item that is too close in the feature space. In the embedding space, positives should be repelled from the anchor. The *negative pool* of anchor $\mathbf{x}^r$ is defined accordingly as

$$P^-(\mathbf{x}^r) = \{\mathbf{x} \in X : g(\mathbf{x}) \in \text{NN}_k^e(\mathbf{y}^r) \setminus \text{NN}_k^m(\mathbf{y}^r)\}. \quad (6)$$

It is common practice, and known to be beneficial [49], to select hard negative examples. By construction, its size is controlled by k. Again, we maintain the pool ordered by descending Euclidean similarity to keep the hardest negative examples.

Anchors. We are interested in anchors that have many relevant images in the collection, which facilitates propagating on the manifold and discovering differences with the Euclidean neighborhood. We are also interested in anchors that are diverse, so that there is as little redundancy during training. Both conditions are satisfied by the modes of the nearest neighbor graph G, which we compute as follows.

We first compute the stationary probability distribution $\boldsymbol{\pi}$ of a random walk [8] on G. This is achieved by the power iteration method [28] yielding the leading left eigenvector of the transition matrix $P = D^{-1}A$, such that $\boldsymbol{\pi}P = \boldsymbol{\pi}$. The probability reflects the *importance* of each node in the graph, as expressed by the probability of a random walker visiting it. We find the local maxima of the stationary distribution on G and out of those, we keep a fixed number having the maximum probability. This is defined as the *anchor set* $\mathcal{A}$. This method has been previously used for image graph visualization [12].

3.4. Training

The *complete training pool* $\mathcal{P}$ is the union of the anchor set $\mathcal{A}$ and the positive and negative pools $P^+(\mathbf{x}), P^-(\mathbf{x})$ for each $\mathbf{x} \in A$. We follow the common practice in metric learning and train a network with two or three branches and use contrastive or triplet loss, respectively. All branches share weights, while the particular network architecture is application specific and is discussed in Section 4.

In both cases of contrastive or triplet loss, we form training tuples of one anchor $\mathbf{x}^r \in A$, one positive $\mathbf{x}^+ \in P^+(\mathbf{x}^r)$, and one negative item $\mathbf{x}^- \in P^-(\mathbf{x}^r)$. At each epoch, the embedding $\mathbf{z} = f(\mathbf{x}; \theta)$ for each $\mathbf{x} \in \mathcal{P}$, where θ is the current set of parameters. For each anchor $\mathbf{x}^r$, a positive item $\mathbf{x}^+$ is drawn at random from its positive pool, and one negative $\mathbf{x}^-$ is drawn at random from a subset of its negative pool. This subset consists of the items corresponding to the Euclidean nearest neighbors of $\mathbf{z}^r = f(\mathbf{x}^r; \theta)$ in the embedding space. Thus, while the manifold neighbors and the training pool are computed once at the beginning, hard negative sampling uses the current network representation. Finally, the training set for this epoch is the set of such tuples $(\mathbf{x}^r, \mathbf{x}^+, \mathbf{x}^-)$.

Given a tuple $(\mathbf{x}^r, \mathbf{x}^+, \mathbf{x}^-)$, we compute the embeddings $\mathbf{z}^r = f(\mathbf{x}^r; \theta)$ $\mathbf{z}^+ = f(\mathbf{x}^r; \theta)$ and $\mathbf{z}^- = f(\mathbf{x}^-; \theta)$, and use the *constrastive loss* [16], combining one positive and one negative pair in a single input,

$$l_c(\mathbf{z}^r, \mathbf{z}^+, \mathbf{z}^-) = \|\mathbf{z}^r - \mathbf{z}^+\|^2 + [m - \|\mathbf{z}^r - \mathbf{z}^-\|]_+^2, \quad (7)$$

or the *triplet loss* [62]

$$l_t(\mathbf{z}^r, \mathbf{z}^+, \mathbf{z}^-) = [m + \|\mathbf{z}^r - \mathbf{z}^+\|^2 - \|\mathbf{z}^r - \mathbf{z}^-\|]_+^2, \quad (8)$$

where $[\cdot]_+$ denotes the positive part and m is a *margin* parameter. We also consider a *weighted* variant of both loss functions, where the loss is multiplied by the manifold similarity $s_m(\mathbf{x}^r, \mathbf{x}^+)$ of the positive sample to the anchor. Thus we down-weigh the contribution of the tuples where the positive sample is too hard.

Of course, given the positive and negative pools, there are many more possibilities in sampling positives and negatives and forming losses that are functions of more than two or three items [29, 36, 5, 27, 59, 67]. It is also possible to iterate our approach, updating the graph and the pools based on the current embedding space, updating the embeddings,

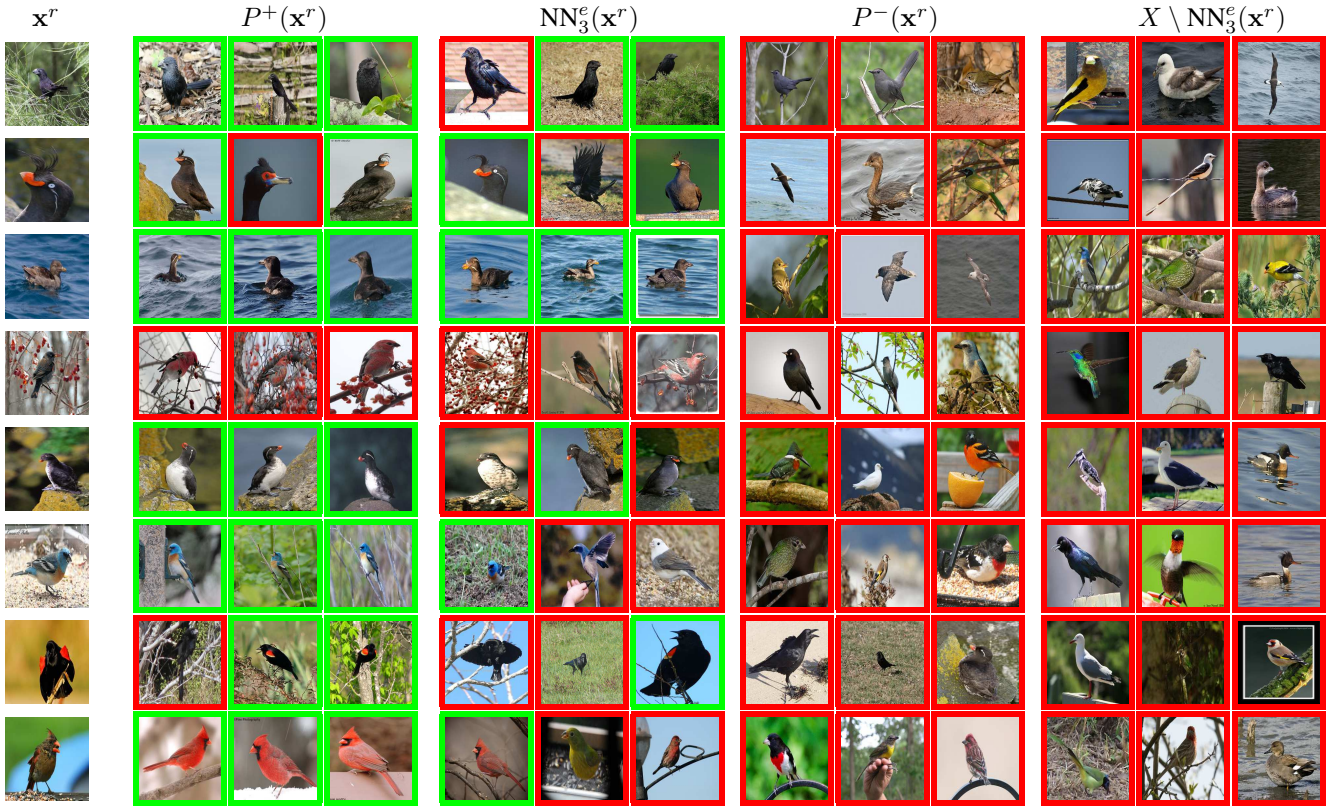


Figure 2. Sample CUB200-2011 anchor images ($\mathbf{x}^r$), positive images from our method ($P^+(\mathbf{x}^r)$) and baseline ($\text{NN}_3^e(\mathbf{x}^r)$), and negative images from our method ($P^-(\mathbf{x}^r)$) and baseline ($X \setminus \text{NN}_3^e(\mathbf{x}^r)$). The baseline is Euclidean nearest neighbors and non-neighbors [16]. Positive (negative) ground-truth framed in green (red). Labels are only used for qualitative evaluation and not during training.

and so on. In this case, given current parameters θ, the set $Z = \{f(\mathbf{x};\theta) : \mathbf{x} \in X\}$ plays the role of Y for the following iteration. This alternating training scheme is common for methods involving a global dataset structure like a graph or a set of clusters [44, 4, 68]. Our idea is orthogonal to most concurrent improvements on metric learning.

4. Applications

We apply the proposed method to learn visual representations on two different tasks: fine-grained categorization [66, 18] and instance-based image retrieval [15, 42]. In both cases, both the features Y and the initial model $f(\cdot,\theta)$ are based on a pre-trained model. We assume there are no labeled images available.

4.1. Fine-grained categorization

We use the CUB200-2011 dataset [61] comprising 200 bird species. The goal is to learn embeddings that better discriminate instances of the same class from instances of different classes. Following the setup of [36], the training set contains half (100) classes on which embedding is learned, while the rest are used for testing. Given a test query image, the remaining test images of the same species should be top-ranked w.r.t. Euclidean distance to the query.

All prior approaches are fully supervised and use the manually assigned labels of the training set. Our method is unsupervised, but otherwise we choose the same settings with the literature in our comparisons. In particular, we initialize the network by GoogLeNet [55] as pre-trained on ImageNet and add a fully connected layer right after the average pooling layer, reducing the embedding dimensionality to $d = 64$. We perform training with the triplet loss using all training images as anchors, which is affordable due to the small size (6k images) of the training set.

Our initial features are formed by R-MAC [58] on the last convolutional feature map of the pre-trained GoogLetNet, right before the average pooling layer, aggregated over 3 input image scales and whitened. The feature set Y contains all such vectors $\mathbf{y} = g(\mathbf{x})$ for $\mathbf{x}$ in the entire training set X. In Figure 2 we show examples of anchors and subsets of their positive and negative pools. Despite the absence of labels and the challenges of fine-grained similarity, we achieve a very clean negative pool and a reasonably clean positive one.

Figure 3. Examples of anchor images selected by the proposed method for the image retrieval application. Random samples from the top 100 (1000) anchors are shown in the top (bottom) row according to the node importance.

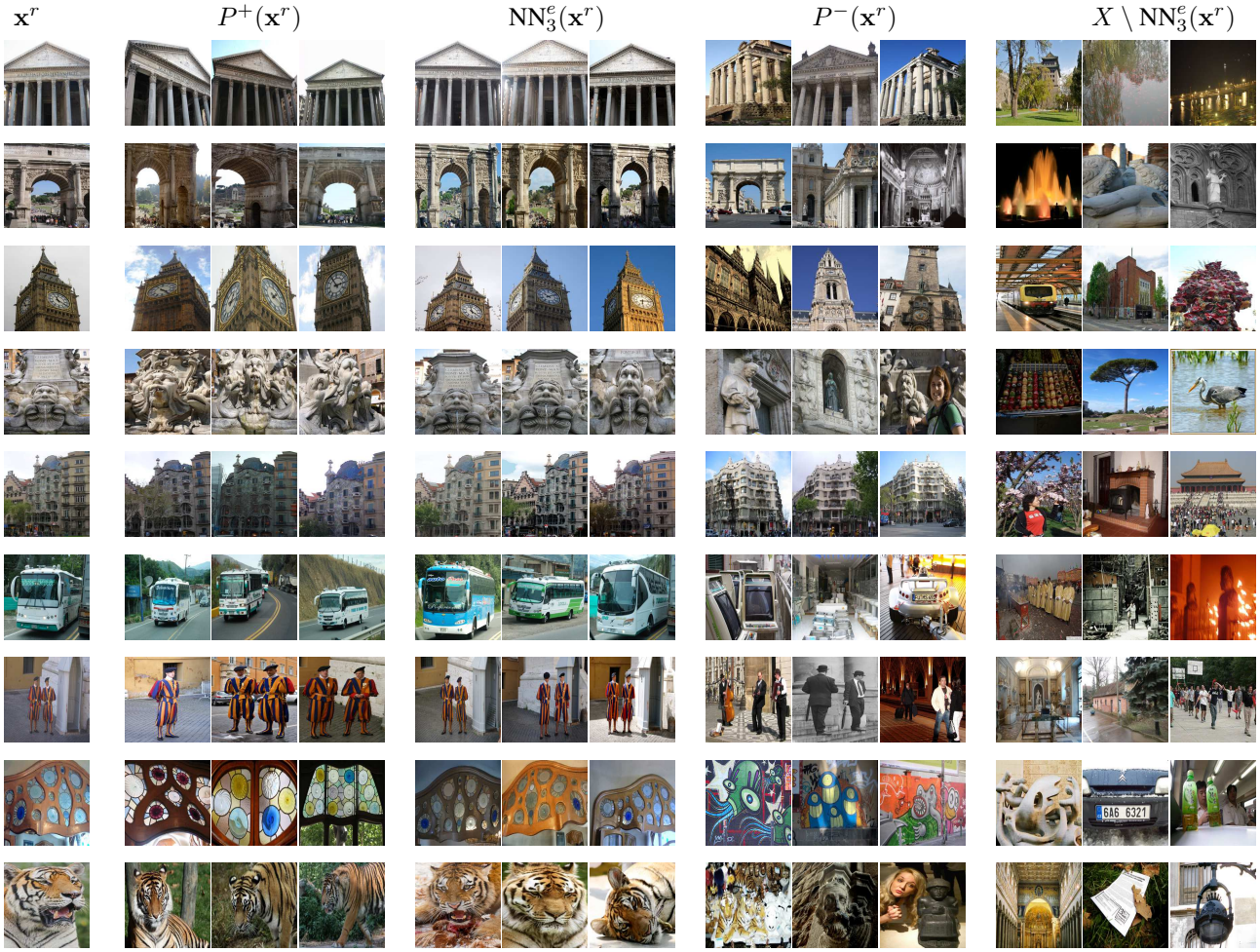


Figure 4. Sample anchor images ($\mathbf{x}^r$), positive images from our method ($P^+(\mathbf{x}^r)$) and baseline ($\mathrm{NN}_3^e(\mathbf{x}^r)$), and negative images from our method ($P^-(\mathbf{x}^r)$) and baseline ($X \setminus \mathrm{NN}_3^e(\mathbf{x}^r)$). The baseline is Euclidean nearest neighbors and non-neighbors [16].

4.2. Particular object retrieval

Particular object retrieval differs from bird species classification in that images are instances of the same object and not of the same (sub-)category, *i.e.* the similarity is even more fine-grained than in bird species. Objects are less deformable, but there is extreme diversity in viewpoint, illumination conditions, occlusion and background clutter.

Methods trained with category-level labeling do not perform well [3], while the state-of-the-art approaches use direct geometric matching [15] or Structure-from-Motion (SfM) [42] to automatically mine matching and non-matching image pairs. This is appropriate since the geometry of the scene is known, but the whole process assumes the existence of another computer vision system based on local descriptors, which is rather expensive [46].

To be comparable to the state-of-the-art fine-tuned MAC obtained through SfM by Radenovic *et al*. [42], we use similar training set and design choices. In particular, we start from the same set of 7M images, referred to as *Flickr 7M* in

the following[2], which is downloaded from Flickr with text tag queries. We limit the training set by randomly sampling 1M images. We initialize our network by VGG [50] as pre-trained on ImageNet and we fine-tune MAC representation with contrastive loss.

Our initial features are formed by R-MAC [58] on the last convolutional feature map of the pre-trained VGG network. The feature set Y contains all such vectors $\mathbf{y} = g(\mathbf{x})$ for $\mathbf{x}$ in the entire training set X. We sample an anchor set $\mathcal{A}$ and construct the complete training pool $\mathcal{P}$. Anchor selection is essential here, as using all images as queries would be very expensive. Compared to [42], our complete training pool is larger (50k vs 22k), however, we only choose 1k anchors per epoch vs 6k.

Examples of selected anchors are shown in Figure 3. They usually correspond to popular locations frequently appearing in the dataset. Examples of training pools are shown in Figure 4, comparing to a baseline where positives and negatives are Euclidean nearest neighbors and non-neighbors, respectively. The same baseline is used for comparisons in our experiments. The baseline positives contain only mild viewpoint and illumination changes, while negatives are random. On the contrary, our positives contain more challenging changes and very interesting negatives: different objects, which still look similar in one way or another. The training set variability, in terms of different objects and viewing conditions, seems a desirable property.

4.3. Implementation details

We always create graph G by considering $k = 30$ nearest neighbors in (1). Exact computation takes 80min on 12 CPU threads on the 1M retrieval training set. We use the Euclidean similarity function $s_e(\mathbf{x}_i, \mathbf{x}_j) = [\mathbf{x}_i^\top \mathbf{x}_j]_+^3$ and $\alpha = 0.99$ following Iscen *et al*. [21]. In order to create the positive pool we consider 50 neighbors in (5), while for the negative pool we use 100 and 10,000 neighbors in (6) for fine-grained categories and retrieval, respectively, due to the different size of the training dataset. We finally restrict the negative pool of each anchor to have 50 instances at most. All vector representations used for an image in the feature and the embedding space are ℓ_2-normalized.

We use stochastic gradient descent with momentum for optimization. The learning rate is initialized at 10^{-2}, and scaled by 0.1 every 10 epochs. The momentum parameter is 0.9. The margin m is set to 0.5 for triplet loss in fine-grained categorization, and to 0.7 for contrastive loss in particular object retrieval. The batch size includes 42 triplets for fine-grained categories [36] and pairs for 5 anchors in the retrieval application [42]. We train for 100 epochs on fine-grained categorization and 30 epochs on particular object retrieval experiments.

[2]Images depicting buildings that are part of the Oxford5k or Paris6k test set are removed as in [42].

Positive	Negative	CUB	Oxford5k	
Anchors		All	Random	$\mathcal{A}$
Initial		35.0	52.6	
NN_5^e	$X \setminus \text{NN}_5^e$	38.5	37.4	41.9
P^+	$X \setminus \text{NN}_5^e$	43.0	48.2	38.1
NN_5^e	P^-	42.1	57.8	71.3
P^+	P^-	43.5	64.4	73.7
$P^+ + W$	$P^- + W$	**45.3**	67.0	**76.7**

Table 1. Impact of choices of anchors and pools of positive and negative examples on Recall@1 on CUB-200-2011 and mAP on Oxford5k. On CUB, all images are used as anchors, while on Oxford5K anchors are selected either at random or by the proposed method ($\mathcal{A}$). The positive and negative pools are formed by either the baseline with Euclidean nearest neighbors (NN_5^e) [16] or our selection (P^+ and P^-), optionally with our weighted loss ($+W$).

Method	Labels	R@1	R@2	R@4	R@8	NMI
Initial	No	35.0	46.8	59.3	72.0	48.1
Triplet+semi-hard [47]	Yes	42.3	55.0	66.4	77.2	55.4
Lifted-Structure (LS) [36]	Yes	43.6	56.6	68.6	79.6	56.5
Triplet+ [18]	Yes	45.9	57.7	69.6	79.8	58.1
Clustering [53]	Yes	48.2	61.4	71.8	81.9	59.2
Triplet+++[18]	Yes	49.8	62.3	74.1	83.3	59.9
Cyclic match[30]	No	40.8	52.8	65.1	76.0	52.6
Ours	No	45.3	57.8	68.6	78.4	55.0

Table 2. Recall@k and NMI on CUB-200-2011. All methods expect for ours and cyclic match [30] use ground-truth labels during training.

5. Experiments

Fine-grained categorization is evaluated on CUB200-2011 [61], where we use the training set without labels for training and then evaluate on the test set, measuring Recall@k as well as clustering quality by NMI [31]. Particular object retrieval is evaluated by mean average precision (mAP) on a challenging and diverse set of test datasets comprising landmark and building images (Oxford5k [40] and Paris6k [41]), natural landscapes (Holidays [24]), as well as planar and 3D objects (Instre [63]). Large scale experiments are performed on Oxford and Paris by adding 100k distractors [40], namely Oxford105k and Paris106k respectively. We first evaluate the importance of different components of the selection strategy and then compare our method against state-of-the-art on each task.

5.1. Impact of positives, negatives, and anchors

We consider the unsupervised approach proposed by Hadsell *et al*. [16] as a baseline method. It sets positives to be the 5 nearest neighbors with Euclidean distance. All other elements are considered negatives out of which we randomly draw the negative pool of an anchor. In this case, hard negative mining per epoch is impossible and random choice is the only choice. We present the results in Table 1. Our method improves the pre-trained network in both tasks

Method	Representation	Labels	Oxford5k	Oxford105k	Paris6k	Paris106k	Holidays	Instre
Pre-trained [58]	MAC	ImageNet	58.5	50.3	73.0	59.0	79.4	48.5
CNN from BoW [42]		SfM	**79.7**	73.9	82.4	74.6	81.4	48.5
Ours		No	78.7	**74.3**	**83.1**	**75.6**	**82.6**	**55.5**
Pre-trained [58]	R-MAC	ImageNet	68.0	61.0	76.6	72.1	87.0	55.6
CNN from BoW [42]		SfM	77.8	70.1	84.1	76.8	84.4	47.7
Ours		No	**78.2**	**72.6**	**85.1**	**78.0**	**87.5**	**57.7**

Table 3. mAP on particular object retrieval datasets. We compare VGG as pre-trained on ImageNet, the fine-tuned network of Radenovic *et al*. [42], and our fine-tuned one. Fine-tuning is always performed for MAC, but at testing we evaluate both global MAC and regional R-MAC [58] representations.

without any supervision, while the weighted loss consistently helps. Furthermore, we observe that our hard negatives are beneficial and necessary, while our anchors are essential for the large scale training set. Given popular anchors, even simple nearest neighbors work well as positives. However, our harder positives further improve.

CUB's annotation allows us to measure the true positive and true negative rate in our positive and negative pools. These measurements are 40% and 96%, respectively. Additionally, we exploit the labels and train with a *positive (negative) oracle*, where we replace our positive (*resp*. negative) pool with one based on labels, yielding 46.7 (*resp*. 36.5) Recall@1. This shows that our hard positives with weighting are almost as good as true positives and that our hard negatives are better than hard annotated negatives. The latter is due to errors in annotation of CUB dataset, quantified to 4.4% [60]. In this case, hard negative mining frequently finds false negative examples, which are known to cause training failure [47].

5.2. Comparisons on fine-grained categories

To our knowledge there is no other unsupervised method that evaluates on CUB200-2011 dataset. We evaluate the unsupervised approach of Li *et al*. [30] by constructing the same graph as in our method, then using the provided code to construct the positive/negative pool and finally training the same way as in our method. We also compare to supervised methods that use ground-truth labels on the training set, but otherwise an identical experimental setup. As shown in in Table 2, our method competes or even outperforms fully supervised methods. We also outperform the only unsupervised competitor. Although their true positive rate is higher (76%), their positive pairs are mostly extremely similar and not challenging enough for training. Note that there are better-performing methods in the literature with sophisticated sampling schemes [67], which can be complementary to ours.

5.3. Comparisons on retrieval

We initialize by VGG as pre-trained on ImageNet and fine-tune using MAC representation. At testing, the fine-tuned network is evaluated with both global MAC and regional R-MAC representations [58]. This is the same process as [42], which makes it comparable in this respect, although the training set and sampling pool sizes are not the same as discussed in section 4.2.

Descriptor whitening is known to be essential for MAC and R-MAC. We follow the common practice in the literature and perform unsupervised PCA whitening [23] for the pre-trained networks, and supervised LDA-based whitening for the fine-tuned networks, learned on a subset of *Flickr 7M*. In particular, as supervision we use SfM labels for the network of Radenovic *et al*. [42], and matching pairs consisting of NN^m_{50} per anchor for our method.

As shown in Table 3, we improve over the pre-trained network on all test sets. Moreover, we outperform [42] with only one exception on Oxford5k. Remarkably, we perform better even on building or landmark oriented test sets, while their method specifically favors this kind of images. With our training pool being more diverse, we improve on Holidays and Instre test sets, where [42] shows little improvement or is even inferior to the pre-trained network.

6. Conclusions

In this work, we depart from using discrete category level labeling in order to learn fine grained similarity. Not only we avoid the expensive or nearly impossible manual annotation, but also do not restrict the problem to supervised classification. Our unsupervised and manifold-aware sampling of training data is applied to perform metric learning. The learning attracts points that lie on the same manifold and repels points on different manifolds. The method is conceptually simple and applicable with standard contrastive and triplet loss. It is shown to be effective for fine-grained categorization and particular object retrieval, competing or surpassing fully supervised approaches.

Acknowledgments The authors were supported by the MSMT LL1303 ERC-CZ grant.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 2

[2] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *NIPS*, 2016. 2

[3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 1, 2, 6

[4] S.-Y. Bai, S. Agethen, T.-H. Chao, and W. Hsu. Semi-supervised learning for convolutional neural networks via online graph construction. In *arXiv*, 2015. 5

[5] M. A. Bautista, A. Sanakoyeu, and B. Ommer. Deep unsupervised similarity learning using partially ordered sets. In *arXiv*, 2017. 1, 2, 4

[6] M. A. Bautista, A. Sanakoyeu, E. Tikhoncheva, and B. Ommer. CliqueCNN: Deep unsupervised exemplar learning. In *NIPS*, 2016. 1, 2

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 2003. 1

[8] M. Cho and K. M. Lee. Mode-seeking on graphs via random walks. In *CVPR*, 2012. 4

[9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 1, 3

[10] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *CVPR*, 2015. 2

[11] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 1, 3

[12] M. Douze, H. Jegou, and F. Perronnin. Polysemous codes. In *ECCV*, 2016. 4

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32, September 2010. 2

[14] L. Gomez, Y. Patel, M. Rusinol, D. Karatzas, and C. V. Jawahar. Self-supervised learning of visual features through embedding images into text topic spaces. In *CVPR*, 2017. 2

[15] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. *ECCV*, 2016. 1, 2, 3, 5, 6

[16] R. Hadsell, S. Chopra, and Y. Lecun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2, 4, 5, 6, 7

[17] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015. 1

[18] B. Harwood, V. Kumar B G, G. Carneiro, I. Reid, and T. Drummond. Smart mining for deep metric learning. In *ICCV*, 2017. 1, 2, 3, 5, 7

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[20] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Fast spectral ranking for similarity search. In *CVPR*, 2018. 3

[21] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*, 2017. 1, 3, 7

[22] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. In *arXiv*, 2015. 2

[23] H. Jégou and O. Chum. Negative evidences and cooccurrences in image retrieval: The benefit of PCA and whitening. In *ECCV*, October 2012. 8

[24] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, October 2008. 7

[25] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local descriptors into compact codes. In *IEEE Trans. PAMI*, September 2012. 2

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[27] B. Kumar, G. Carneiro, I. Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *CVPR*, 2016. 4

[28] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004. 4

[29] M. Law, N. Thome, and M. Cord. Quadruplet-wise image similarity learning. In *ICCV*, 2013. 4

[30] D. Li, W.-C. Hung, J.-B. Huang, S. Wang, N. Ahuja, and M.-H. Yang. Unsupervised visual representation learning by graph-based consistent constraints. In *ECCV*, 2016. 2, 7, 8

[31] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. 7

[32] B. J. Meyer, B. Harwood, and T. Drummond. Nearest neighbour radial basis function solvers for deep neural networks. In *arXiv*, 2017. 1

[33] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 2

[34] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 1, 3

[35] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[36] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 1, 2, 4, 5, 7

[37] G. Pai, R. Talmon, and R. Kimmel. Parametric manifold learning via sparse multidimensional scaling. *arXiv preprint arXiv:1711.06011*, 2017. 3

[38] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015. 2

[39] F. Perronnin and C. R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, June 2007. 2

[40] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, June 2007. 7

[41] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, June 2008. 7

[42] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. *ECCV*, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[43] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4:251–258, 2016. 2

[44] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. In *arXiv*, 2015. 1, 5

[45] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003. 1

[46] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm. From single image query to detailed 3d reconstruction. In *CVPR*, pages 5126–5134, 2015. 6

[47] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2, 7, 8

[48] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015. 4

[49] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer. Fracking deep convolutional image descriptors. In *arXiv*, 2014. 4

[50] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 1, 7

[51] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2

[52] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, pages 1857–1865, 2016. 2

[53] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Deep metric learning via facility location. In *CVPR*, 2017. 2, 7

[54] K.-K. Sung. Learning and example selection for object and pattern detection. Technical report, MIT, 1996. 2

[55] 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*, 2015. 1, 5

[56] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1

[57] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, December 2013. 2

[58] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *ICLR*, 2016. 5, 7, 8

[59] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016. 4

[60] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, 2015. 8

[61] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 5, 7

[62] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 1, 3, 4

[63] S. Wang and S. Jiang. Instre: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11:37, 2015. 7

[64] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *CVPR*, 2015. 1, 2, 3

[65] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *arXiv*, 2017. 1, 3

[66] Y. Wang, J. Choi, V. Morariu, and L. S. Davis. Mining discriminative triplets of patches for fine-grained classification. In *CVPR*, 2016. 5

[67] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Sampling matters in deep embedding learning. In *arXiv*, 2017. 2, 4, 8

[68] J. Xu, C. Wang, C. Qi, C. Shi, and B. Xiao. Iterative manifold embedding layer learned by incomplete data for large-scale image retrieval. In *arXiv*, 2017. 3, 5

[69] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*, 2003. 3