Unifying Deep Local and Global Features for Image Search

Bingyi Cao, André Araujo(✉), and Jack Sim

Google Research, Mountain View, USA
bingyi@google.com, andrearaujo@google.com, jacksim@google.com

Abstract. Image retrieval is the problem of searching an image database for items that are similar to a query image. To address this task, two main types of image representations have been studied: global and local image features. In this work, our key contribution is to unify global and local features into a single deep model, enabling accurate retrieval with efficient feature extraction. We refer to the new model as DELG, standing for DEep Local and Global features. We leverage lessons from recent feature learning work and propose a model that combines generalized mean pooling for global features and attentive selection for local features. The entire network can be learned end-to-end by carefully balancing the gradient flow between two heads – requiring only image-level labels. We also introduce an autoencoder-based dimensionality reduction technique for local features, which is integrated into the model, improving training efficiency and matching performance. Comprehensive experiments show that our model achieves state-of-the-art image retrieval on the Revisited Oxford and Paris datasets, and state-of-the-art single-model instance-level recognition on the Google Landmarks dataset v2. Code and models are available at https://github.com/tensorflow/models/tree/master/research/delf.

Keywords: Deep features · Image retrieval · Unified model

1 Introduction

Large-scale image retrieval is a long-standing problem in computer vision, which saw promising results [26,38,43,44] even before deep learning revolutionized the field. Central to this problem are the representations used to describe images and their similarities.

B. Cao and A. Araujo—Contributed equally to this work.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58565-5_43) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12365, pp. 726–743, 2020.
https://doi.org/10.1007/978-3-030-58565-5_43

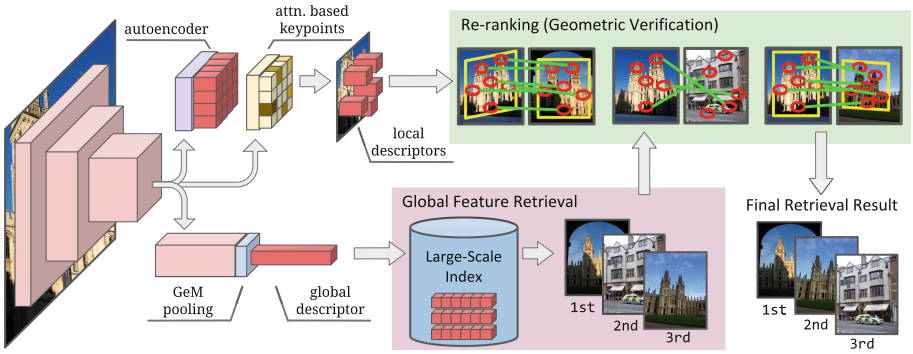

Fig. 1. Our proposed **DELG (DEep Local and Global features)** model (left) jointly extracts deep local and global features. Global features can be used in the first stage of a retrieval system, to efficiently select the most similar images (bottom). Local features can then be employed to re-rank top results (top-right), increasing precision of the system. The unified model leverages hierarchical representations induced by convolutional neural networks to learn local and global features, combined with recent advances in global pooling and attentive local feature detection.

Two types of image representations are necessary for high image retrieval performance: global and local features. A global feature [1,17,26,46,47], also commonly referred to as "global descriptor" or "embedding", summarizes the contents of an image, often leading to a compact representation; information about spatial arrangement of visual elements is lost. Local features [7,28,34,39,62], on the other hand, comprise descriptors and geometry information about specific image regions; they are especially useful to match images depicting rigid objects. Generally speaking, global features are better at recall, while local features are better at precision. Global features can learn similarity across very different poses where local features would not be able to find correspondences; in contrast, the score provided by local feature-based geometric verification usually reflects image similarity well, being more reliable than global feature distance. A common retrieval system setup is to first search by global features, then re-rank the top database images using local feature matching – to get the best of both worlds. Such a hybrid approach gained popularity in visual localization [49,54] and instance-level recognition problems [42,61].

Today, most systems that rely on both these types of features need to separately extract each of them, using different models. This is undesirable since it may lead to high memory usage and increased latency, e.g., if both models require specialized and limited hardware such as GPUs. Besides, in many cases similar types of computation are performed for both, resulting in redundant processing and unnecessary complexity.

Contributions. (1) Our first contribution is a unified model to represent both local and global features, using a convolutional neural network (CNN), referred to as DELG (DEep Local and Global features) – illustrated in Fig. 1. This allows

for efficient inference by extracting an image's global feature, detected keypoints and local descriptors within a single model. Our model is enabled by leveraging hierarchical image representations that arise in CNNs [64], which we couple to generalized mean pooling [46] and attentive local feature detection [39]. (2) Second, we adopt a convolutional autoencoder module that can successfully learn low-dimensional local descriptors. This can be readily integrated into the unified model, and avoids the need of post-processing learning steps, such as PCA, that are commonly used. (3) Finally, we design a procedure that enables end-to-end training of the proposed model using only image-level supervision. This requires carefully controlling the gradient flow between the global and local network heads during backpropagation, to avoid disrupting the desired representations. Through systematic experiments, we show that our joint model achieves state-of-the-art performance on the Revisited Oxford, Revisited Paris and Google Landmarks v2 datasets.

2 Related Work

We review relevant work in local and global features, focusing mainly on approaches related to image retrieval.

Local Features. Hand-crafted techniques such as SIFT [28] and SURF [7] have been widely used for retrieval problems. Early systems [28,32,40] worked by searching for query local descriptors against a large database of local descriptors, followed by geometrically verifying database images with sufficient number of correspondences. Bag-of-Words [53] and related methods [24,43,44] followed, by relying on visual words obtained via local descriptor clustering, coupled to TF-IDF scoring. The key advantage of local features over global ones for retrieval is the ability to perform spatial matching, often employing RANSAC [15]. This has been widely used [3,43,44], as it produces reliable and interpretable scores. Recently, several deep learning-based local features have been proposed [6,14,29, 33,34,39,41,48,62]. The one most related to our work is DELF [39]; our proposed unified model incorporates DELF's attention module, but with a much simpler training pipeline, besides also enabling global feature extraction.

Global Features excel at delivering high image retrieval performance with compact representations. Before deep learning was popular in computer vision, they were developed mainly by aggregating hand-crafted local descriptors [25–27, 57]. Today, most high-performing global features are based on deep convolutional neural networks [1,4,5,17,46,47,58], which are trained with ranking-based [9,19, 50] or classification losses [11,60]. Our work leverages recent learned lessons in global feature design, by adopting GeM pooling [46] and ArcFace loss [11]. This leads to improved global feature retrieval performance compared to previous techniques, which is further boosted by geometric re-ranking with local features obtained from the same model.

Joint Local and Global CNN Features. Previous work considered neural networks for joint extraction of global and local features. For indoor localization,

Taira et al. [54] used NetVLAD [1] to extract global features for candidate pose retrieval, followed by dense local feature matching using feature maps from the same network. Simeoni et al.'s DSM [52] detected keypoints in activation maps from global feature models using MSER [30]; activation channels are interpreted as visual words, in order to propose correspondences between a pair of images. Our work differs substantially from [52,54], since they only post-process pre-trained global feature models to produce local features, while we jointly train local and global. Sarlin et al. [49] distill pre-trained local [12] and global [1] features into a single model, targeting localization applications. In contrast, our model is trained end-to-end for image retrieval, and is not limited to mimicking separate pre-trained local and global models. To the best of our knowledge, ours is the first work to learn a non-distilled model producing both local and global features.

Dimensionality Reduction for Image Retrieval. PCA and whitening are widely used for dimensionality reduction of local and global features in image retrieval [4,39,47,58]. As discussed in [23], whitening downweights co-occurrences of local features, which is generally beneficial for retrieval applications. Mukundan et al. [35] further introduce a shrinkage parameter that controls the extent of applied whitening. If supervision in the form of matching pairs or category labels is available, more sophisticated methods [18,31] can be used. More recently, Gordo et al. [16] propose to replace PCA/whitening by a fully-connected layer, that is learned together with the global descriptor.

In this paper, our goal is to compose a system that can be learned end-to-end, using only image-level labels and without requiring post-processing stages that make training more complex. Also, since we extract local features from feature maps of common CNN backbones, they tend to be very high-dimensional and infeasible for large-scale problems. All above-mentioned approaches would either require a separate post-processing step to reduce the dimensionality of features, or supervision at the level of local patches – making them unsuitable to our needs. We thus introduce an autoencoder in our model, which can be jointly and efficiently learned with the rest of the network. It requires no extra supervision as it can be trained with a reconstruction loss.

3 DELG

3.1 Design Considerations

For optimal performance, image retrieval requires semantic understanding of the types of objects that a user may be interested in, such that the system can distinguish between relevant objects versus clutter/background. Both local and global features should thus focus only on the most discriminative information within the image. However, there are substantial differences in terms of the desired behavior for these two feature modalities, posing a considerable challenge to jointly learn them.

Global features should be similar for images depicting the same object of interest, and dissimilar otherwise. This requires high-level, abstract representations that are invariant to viewpoint and photometric transformations. Local features, on the other hand, need to encode representations that are grounded to specific image regions; in particular, the keypoint detector should be equivariant with respect to viewpoint, and the keypoint descriptor needs to encode localized visual information. This is crucial to enable geometric consistency checks between query and database images, which are widely used in image retrieval systems.

Besides, our goal is to design a model that can be learned end-to-end, with local and global features, without requiring additional learning stages. This simplifies the training pipeline, allowing faster iterations and wider applicability. In comparison, it is common for previous feature learning work to require several learning stages: attentive deep local feature learning [39] requires 3 learning stages (fine-tuning, attention, PCA); deep global features usually require two stages, e.g., region proposal and Siamese training [17], or Siamese training and supervised whitening [46], or ranking loss training and PCA [47].

3.2 Model

We design our DELG model, illustrated in Fig. 1, to fulfill the requirements outlined above. We propose to leverage hierarchical representations from CNNs [64] in order to represent the different types of features to be learned. While global features can be associated with deep layers representing high-level cues, local features are more suitable to intermediate layers that encode localized information.

Given an image, we apply a convolutional neural network backbone to obtain two feature maps: $\mathcal{S} \in \mathcal{R}^{H_S \times W_S \times C_S}$ and $\mathcal{D} \in \mathcal{R}^{H_D \times W_D \times C_D}$, representing shallower and deeper activations respectively, where H, W, C correspond to the height, width and number of channels in each case. For common convolutional networks, $H_D \leq H_S$, $W_D \leq W_S$ and $C_D \geq C_S$; deeper layers have spatially smaller maps, with a larger number of channels. Let $s_{h,w} \in \mathcal{R}^{C_S}$ and $d_{h,w} \in \mathcal{R}^{C_D}$ denote features at location h, w in these maps. For common network designs, these features are non-negative since they are obtained after the ReLU nonlinearity, which is the case in our method.

In order to aggregate deep activations into a global feature, we adopt generalized mean pooling (GeM) [46], which effectively weights the contributions of each feature. Another key component of global feature learning is to whiten the aggregated representation; we integrate this into our model with a fully-connected layer $F \in \mathcal{R}^{C_F \times C_D}$, with learned bias $b_F \in \mathcal{R}^{C_F}$, similar to [17]. These two components produce a global feature $g \in \mathcal{R}^{C_F}$ that summarizes the discriminative contents of the whole image:

$$g = F \times \left(\frac{1}{H_D W_D} \sum_{h,w} d_{h,w}^p \right)^{1/p} + b_F \tag{1}$$

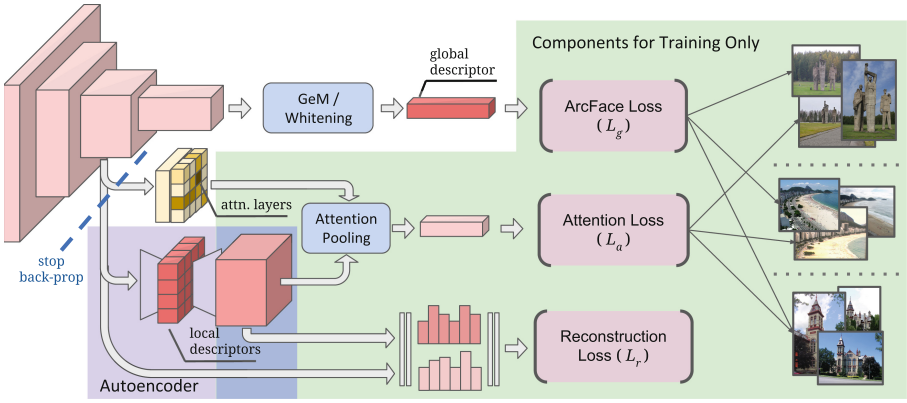

Fig. 2. Illustration of our **training pipeline**. The components highlighted in green are used solely during training. There are two classification losses: ArcFace for global feature learning (L_g), and softmax for attention learning (L_a). In both cases, the classification objective is to distinguish different landmarks (an instance-level recognition problem). The autoencoder (purple) is further trained with a reconstruction loss (L_r). The whole model is learned end-to-end, and benefits substantially from stopping gradient back-propagation from L_a and L_r into the CNN backbone. (Color figure online)

where p denotes the generalized mean power parameter, and the exponentiation $d_{h,w}^p$ is applied elementwise.

Regarding local features, it is important to select only the relevant regions for matching. This can be achieved by adopting an attention module M [39], whose goal is to predict which among the extracted local features are discriminative for the objects of interest. This is performed as $\mathcal{A} = M(\mathcal{S})$, where M is a small convolutional network and $\mathcal{A} \in \mathcal{R}^{H_S \times W_S}$ denotes the attention score map associated to the features from $\mathcal{S}$.

Furthermore, since hundreds to thousands of local features are commonly used, they must be represented compactly. To do so, we propose to integrate a small convolutional autoencoder (AE) module [21], which is responsible for learning a suitable low-dimensional representation. The local descriptors are obtained as $\mathcal{L} = T(\mathcal{S})$, where $\mathcal{L} \in \mathcal{R}^{H_S \times W_S \times C_T}$, and T is the encoding part of the autoencoder, corresponding to a 1×1 convolutional layer with C_T filters. Note that, contrary to $\mathcal{S}$, the local descriptors $\mathcal{L}$ are not restricted to be non-negative.

Each extracted local feature at position h, w is thus represented with a local descriptor $l_{h,w} \in \mathcal{L}$ and its corresponding keypoint detection score $a_{h,w} \in \mathcal{A}$. Their locations in the input image are set to corresponding receptive field centers, which can be computed using the parameters of the network [2].

The global and local descriptors are L_2-normalized into $\hat{g}$ and $\hat{l}_{h,w}$, respectively.

3.3 Training

We propose to train the model using only image-level labels, as illustrated in Fig. 2. In particular, note that we do not require patch-level supervision to train local features, unlike most recent works [14,29,36,48].

Besides the challenge to acquire the annotations, note that patch-level supervision could help selecting repeatable features, but not necessarily the discriminative ones; in contrast, our model discovers discriminative features by learning which can distinguish the different classes, given by image-level labels. In this weakly-supervised local feature setting, it is very important to control the gradient flow between the global and local feature learning, which is discussed in more detail below.

Global Features. For global feature learning, we adopt a suitable loss function with L_2-normalized classifier weights $\hat{\mathcal{W}}$, followed by scaled softmax normalization and cross-entropy loss [59]; this is sometimes referred to as "cosine classifier". Additionally, we adopt the ArcFace margin [11], which has shown excellent results for global feature learning by inducing smaller intra-class variance. Concretely, given $\hat{g}$, we first compute the cosine similarity against $\hat{\mathcal{W}}$, adjusted by the ArcFace margin. The ArcFace-adjusted cosine similarity can be expressed as $\mathrm{AF}(u, c)$:

$$\mathrm{AF}(u, c) = \begin{cases} \cos(\mathrm{acos}(u) + m), & \text{if } c = 1 \\ u, & \text{if } c = 0 \end{cases} \tag{2}$$

where u is the cosine similarity, m is the ArcFace margin and c is a binary value indicating if this is the ground-truth class. The cross-entropy loss, computed using softmax normalization can be expressed in this case as:

$$L_g(\hat{g}, y) = -\log\left(\frac{\exp(\gamma \times \mathrm{AF}(\hat{w}_k^T \hat{g}, 1))}{\sum_n \exp(\gamma \times \mathrm{AF}(\hat{w}_n^T \hat{g}, y_n))}\right) \tag{3}$$

where γ is a learnable scalar, $\hat{w}_i$ refers to the L_2-normalized classifier weights for class i, y is the one-hot label vector and k is the index of the ground-truth class ($y_k = 1$).

Local Features. To train the local features, we use two losses. First, a mean-squared error regression loss that measures how well the autoencoder can reconstruct $\mathcal{S}$. Denote $\mathcal{S}' = T'(\mathcal{L})$ as the reconstructed version of $\mathcal{S}$, with same dimensions, where T' is a 1×1 convolutional layer with C_S filters, followed by ReLU. The loss can be expressed as:

$$L_r(\mathcal{S}', \mathcal{S}) = \frac{1}{H_S W_S C_S} \sum_{h,w} \|s'_{h,w} - s_{h,w}\|^2 \tag{4}$$

Second, a cross-entropy classification loss that incentivizes the attention module to select discriminative local features. This is done by first pooling the reconstructed features $\mathcal{S}'$ with attention weights $a_{h,w}$:

$$a' = \sum_{h,w} a_{h,w} s'_{h,w} \tag{5}$$

Then using a standard softmax-cross-entropy loss:

$$L_a(a', k) = -\log\left(\frac{\exp(v_k^T a' + b_k)}{\sum_n \exp(v_n^T a' + b_n)}\right) \quad (6)$$

where v_i, b_i refer to the classifier weights and biases for class i and k is the index of the ground-truth class; this tends to make the attention weights large for the discriminative features. The total loss is given by $L_g + \lambda L_r + \beta L_a$.

Controlling Gradients. Naively optimizing the above-mentioned total loss experimentally leads to suboptimal results, because the reconstruction and attention loss terms significantly disturb the hierarchical feature representation which is usually obtained when training deep models. In particular, both tend to induce the shallower features $\mathcal{S}$ to be more semantic and less localizable, which end up being sparser. Sparser features can more easily optimize L_r, and more semantic features may help optimizing L_a; this, as a result, leads to underperforming local features.

We avoid this issue by stopping gradient back-propagation from L_r and L_a to the network backbone, i.e., to $\mathcal{S}$. This means that the network backbone is optimized solely based on L_g, and will tend to produce the desired hierarchical feature representation. This is further discussed in the experimental section that follows.

4 Experiments

4.1 Experimental Setup

Model Backbone and Implementation. Our model is implemented using TensorFlow, leveraging the Slim model library [51]. We use ResNet-50 (R50) and ResNet-101 (R101) [20]; R50 is used for ablation experiments. We obtain the shallower feature map $\mathcal{S}$ from the *conv4* output, and the deeper feature map $\mathcal{D}$ from the *conv5* output. Note that the Slim implementation moves the *conv5* stride into the last unit from *conv4*, which we also adopt – helping reduce the spatial resolution of $\mathcal{S}$. The number of channels in $\mathcal{D}$ is $C_D = 2048$; GeM pooling [46] is applied with parameter $p = 3$, which is not learned. The whitening fully-connected layer, applied after pooling, produces a global feature with dimensionality $C_F = 2048$. The number of channels in $\mathcal{S}$ is $C_S = 1024$; the autoencoder module learns a reduced dimensionality for this feature map with $C_T = 128$. The attention network M follows the setup from [39], with 2 convolutional layers, without stride, using kernel sizes of 1; as activation functions, the first layer uses ReLU and the second uses Softplus [13].

Training Details. We use the training set of the Google Landmarks dataset (GLD) [39], containing 1.2M images from 15k landmarks, and divide it into two subsets 'train'/'val' with 80%/20% split. The 'train' split is used for the actual learning, and the 'val' split is used for validating the learned classifier as training progresses. Models are initialized from pre-trained ImageNet weights. The images

first undergo augmentation, by randomly cropping/distorting the aspect ratio; then, they are resized to 512×512 resolution. We use a batch size of 16, and train using 21 Tesla P100 GPUs asynchronously, for 1.5M steps (corresponding to approximately 25 epochs of the 'train' split). The model is optimized using SGD with momentum of 0.9, and a linearly decaying learning rate that reaches zero once the desired number of steps is reached. We experiment with initial learning rates within $[3 \times 10^{-4}, 10^{-2}]$ and report results for the best performing one. We set the ArcFace margin $m = 0.1$, the weight for L_a to $\beta = 1$, and the weight for L_r to $\lambda = 10$. The learnable scalar for the global loss L_g is initialized to $\gamma = \sqrt{C_F} = 45.25$.

Evaluation Datasets. To evaluate our model, we use several datasets. First, Oxford [43] and Paris [44], with revisited annotations [45], referred to as $\mathcal{R}$Oxf and $\mathcal{R}$Par, respectively. There are 4993 (6322) database images in the $\mathcal{R}$Oxf ($\mathcal{R}$Par) dataset, and a different query set for each, both with 70 images. Performance is measured using mean average precision (mAP). Large-scale results are further reported with the $\mathcal{R}$1M distractor set [45], which contains 1M images. As in previous papers [37,47,55], parameters are tuned in $\mathcal{R}$Oxf/$\mathcal{R}$Par, then kept fixed for the large-scale experiments. Second, we report large-scale instance-level retrieval and recognition results on the Google Landmarks dataset v2 (GLDv2) [61], using the latest ground-truth version (2.1). GLDv2-retrieval has 1129 queries (379 validation and 750 testing) and 762k database images; performance is measured using mAP@100. GLDv2-recognition has 118k test (41k validation and 77k testing) and 4M training images from 203k landmarks; the training images are only used to retrieve images and their scores/labels are used to form the class prediction; performance is measured using μAP@1. We perform minimal parameter tuning based on the validation split, and report results on the testing split.

Feature Extraction and Matching. We follow the convention from previous work [17,39,46] and use an image pyramid at inference time to produce multi-scale representations. For global features, we use 3 scales, $\{\frac{1}{\sqrt{2}}, 1, \sqrt{2}\}$; L_2 normalization is applied for each scale independently, then the three global features are average-pooled, followed by another L_2 normalization step. For local features, we experiment with the same 3 scales, but also with the more expensive setting from [39] using 7 image scales in total, with range from 0.25 to 2.0 (this latter setting is used unless otherwise noted). Local features are selected based on their attention scores $\mathcal{A}$; a maximum of 1k local features are allowed, with a minimum attention score τ, where we set τ to the median attention score in the last iteration of training, unless otherwise noted. For local feature matching, we use RANSAC [15] with an affine model. When re-ranking global feature retrieval results with local feature-based matching, the top 100 ranked images from the first stage are considered. For retrieval datasets, the final ranking is based on the number of inliers, then breaking ties using the global feature distance; for the recognition dataset, we follow the exact protocol from the GLDv2 paper [61] to combine local and global scores, aggregating scores for different classes based on the top-ranked images. Our focus is on improving global and local features for

Table 1. Local feature ablation. Comparison of local features, trained separately or jointly, with different methods for dimensionality reduction (DR). We report average precision (AP) results of matching image pairs from the Google Landmarks dataset (GLD).

DR method	λ	Jointly trained	Stop gradients	GLD-pairs AP (%)
PCA [39]	-	✗	-	51.48
FC	-			52.67
AE [ours]	0	✗	-	49.95
	1			51.28
	5			52.26
	10			54.21
	20			53.51
	10	✓	✗	37.05
	10		✓	53.73

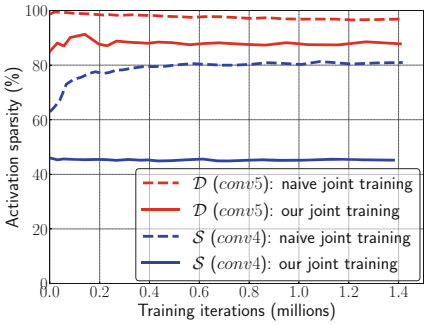

Fig. 3. Evolution of **activation sparsity** over training iterations for $\mathcal{D}$ (*conv5*) and $\mathcal{S}$ (*conv4*), comparing the naive joint training method and our improved version that controls gradient propagation. The naive method leads to much sparser feature maps.

Table 2. Global feature ablation. Comparison of global features, trained separately or jointly, with different pooling methods (SPoC, GeM) and loss functions (Softmax, ArcFace). We report mean average precision (mAP %) on the $\mathcal{R}$Oxf and $\mathcal{R}$Par datasets.

		Jointly trained	Stop gradients	Medium		Hard	
Pooling	Loss			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
SPoC	Softmax	✗	–	51.2	72.0	26.3	47.7
SPoC	ArcFace	✗	–	59.8	80.8	35.6	61.7
GeM	ArcFace	✗	–	69.3	**82.2**	44.4	**64.0**
GeM	ArcFace	✓	✗	68.8	78.9	42.4	58.3
GeM	ArcFace	✓	✓	**69.7**	81.6	**45.1**	63.4

retrieval/recognition, so we do not consider techniques that post-process results such as query expansion [10,46] or diffusion/graph traversal [8,22]. These are expensive due to requiring additional passes over the database, but if desired could be integrated to our system and produce stronger performance.

4.2 Results

First, we present ablation experiments, to compare features produced by our joint model against their counterparts which are separately trained, and also to discuss the effect of controlling the gradient propagation. For a fair comparison, our jointly trained features are evaluated against equivalent separately-trained models, with the same hyperparameters as much as possible. Then, we compare our models against state-of-the-art techniques. See also the appendices for more details, visualizations and discussions.

Local Features. As an ablation, we evaluate our local features by matching image pairs. We select 200k pairs, each composed of a test and a train image from GLD, where in 1k pairs both images depict the same landmark, and in 199k pairs the two images depict different landmarks. We compute average precision (AP) after ranking the pairs based on the number of inliers. All variants for this experiment use τ equals to the 75^{th} percentile attention score in the last iteration of training. Results are presented in Table 1.

First, we train solely the attention and dimensionality reduction modules, for 500k iterations, all methods initialized with the same weights from a separately-trained global feature model. These results are marked as not being jointly trained. It can be seen that our AE outperforms PCA and a simpler method using only a single fully-connected (FC) layer. Performance improves for the AE as λ increases from 0 to 10, decreasing with 20. Then, we jointly train the unified model; in this case, the variant that does not stop gradients to the backbone suffers a large drop in performance, while the variant that stops gradients obtains similar results as in the separately-trained case.

The poor performance of the naive jointly trained model is due to the degradation of the hierarchical feature representation. This can be assessed by observing the evolution of activation sparsity in $\mathcal{S}$ (*conv4*) and $\mathcal{D}$ (*conv5*), as shown in Fig. 3. Generally, layers representing more abstract and high-level semantic properties (usually deeper layers) have high levels of sparsity, while shallower layers representing low-level and more localizable patterns are dense. As a reference, the ImageNet pre-trained model presents on average 45% and 82% sparsity for these two feature maps, respectively, when run over GLD images. For the naive joint training case, the activations of both layers quickly become much sparser, reaching 80% and 97% at the end of training; in comparison, our proposed training scheme preserves similar sparsity as the ImageNet model: 45% and 88%. This suggests that the *conv4* features in the naive case degrade for the purposes of local feature matching; controlling the gradient effectively resolves this issue.

Global Features. Table 2 compares global feature training methods. The first three rows present global features trained with different loss and pooling techniques. We experiment with standard Softmax Cross-Entropy and ArcFace [11] losses; for pooling, we consider standard average pooling (equivalent to SPoC [4]) and GeM [46]. ArcFace brings an improvement of up to 14%, and GeM of up to 9.5%. GeM pooling and ArcFace loss are adopted in our final model. Naively training a joint model, without controlling gradients, underperforms when compared to the baseline separately-trained global feature, with mAP decrease of up to 5.7%. Once gradient stopping is employed, the performance can be recovered to be on par with the separately-trained version (a little better on $\mathcal{R}$Oxf, a little worse on $\mathcal{R}$Par). This is expected, since the global feature in this case is optimized by itself, without influence from the local feature head.

Comparison to Retrieval State-of-the-Art. Table 3 compares our model against the retrieval state-of-the-art. Three settings are presented: (A) local feature aggregation and re-ranking (previous work); (B) global feature similarity

Table 3. Comparison to retrieval state-of-the-art. Results (% mAP) on the $\mathcal{R}$Oxf/$\mathcal{R}$Par datasets (and their large-scale versions $\mathcal{R}$Oxf+1M/$\mathcal{R}$Par+1M), with both Medium and Hard evaluation protocols. The top set of rows (A) presents previous work's results using local feature aggregation and re-ranking. Other sets of rows present results using (B) global features only, or (C) global features for initial search then re-ranking using local features. DELG* refers to a version of DELG where the local features are binarized. DELG and DELG* outperform previous work in setups (B) and (C) substantially. DELG also outperforms methods from setting (A) in 7 out of 8 cases.

Method	Medium				Hard			
	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M	$\mathcal{R}$Oxf	+1M	$\mathcal{R}$Par	+1M
(A) Local feature aggregation + re-ranking								
HesAff-rSIFT-ASMK*+SP [57]	60.6	46.8	61.4	42.3	36.7	26.9	35.0	16.8
HesAff-HardNet-ASMK*+SP [34]	65.6	–	65.2	–	41.1	–	38.5	–
DELF-ASMK* +SP [39,45]	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
DELF-R-ASMK*+SP (GLD) [55]	**76.0**	**64.0**	**80.2**	**59.7**	**52.4**	**38.1**	**58.6**	**29.4**
(B) Global features								
AlexNet-GeM [46]	43.3	24.2	58.0	29.9	17.1	9.4	29.7	8.4
VGG16-GeM [46]	61.9	42.6	69.3	45.4	33.7	19.0	44.3	19.1
R101-R-MAC [17]	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
R101-GeM [46]	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
R101-GeM↑ [52]	65.3	46.1	77.3	52.6	39.6	22.2	56.6	24.8
R101-GeM-AP [47]	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
R101-GeM-AP (GLD) [47]	66.3	–	80.2	–	42.5	–	60.8	–
R152-GeM (GLD) [46]	68.7	–	79.7	–	44.2	–	60.3	–
R101-GeM+SOLAR (GLD) [37]	69.9	53.5	81.6	59.2	47.9	29.9	64.5	33.4
R50-DELG **[ours]**	69.7	**55.0**	81.6	59.7	45.1	27.8	63.4	34.1
R101-DELG **[ours]**	**73.2**	54.8	**82.4**	**61.8**	**51.2**	**30.3**	**64.7**	**35.5**
(C) Global features + Local feature re-ranking								
R101-GeM↑+DSM [52]	65.3	47.6	77.4	52.8	39.2	23.2	56.2	25.0
R50-DELG* **[ours]**	–	60.4	–	60.3	–	35.3	–	34.1
R101-DELG (3 scales global & local) **[ours]**	77.2	61.7	82.4	62.3	55.4	37.5	62.7	35.3
R101-DELG* (3 scales global & local) **[ours]**	–	61.2	–	62.2	–	36.4	–	35.4
R101-DELG **[ours]**	**78.5**	**62.7**	**82.6**	**62.5**	**58.6**	**39.2**	**63.9**	**36.3**
R101-DELG* **[ours]**	–	62.2	–	62.4	–	38.3	–	36.1

search; (C) global feature search followed by re-ranking with local feature matching and spatial verification (SP).

In setting (B), the DELG global feature variants strongly outperform previous work for all cases (most noticeably in the large-scale setting), as well as outperforming concurrent work [37]. Compared to previous work, we see 7.1% improvement in $\mathcal{R}$Oxf+1M-Hard and 7.5% in $\mathcal{R}$Par+1M-Hard. Note that we obtain strong improvements even when using the ResNet-50 backbone, while the previous state-of-the-art used ResNet-101/152, which are much more

Table 4. GLDv2 evaluation. Results on the GLDv2 dataset, for the retrieval and recognition tasks, on the "testing" split of the query set. For a fair comparison, all methods are trained on GLD.

Method	Retrieval mAP (%)	Recognition μAP (%)
DELF-R-ASMK*+SP [55]	18.8	–
R101-GeM+ArcFace [61]	20.7	33.3
R101-GeM+CosFace [63]	21.4	–
DELF-KD-tree [39]	–	44.8
R50-DELG (global-only) [ours]	20.4	32.4
R101-DELG (global-only) [ours]	21.7	32.0
R50-DELG [ours]	22.3	56.8
R101-DELG [ours]	**24.3**	**58.8**

Table 5. Re-ranking experiment. Comparison of DELG against other recent local features; results (% mAP) on the $\mathcal{R}$Oxf dataset.

Method	Hard	Medium
R50-DELG (global-only)	45.1	69.7
Local feature re-ranking		
SIFT [28]	44.4	69.8
SOSNet [56]	45.5	69.9
D2-Net [14]	47.2	70.4
R50-DELG [ours]	**53.7**	**75.4**

complex (2X/3X the number of floating point operations, respectively). To ensure a fair comparison, we present results from [46,47] which specifically use the same training set as ours, marked as "(GLD)" – the results are obtained from the authors' official codebases. In particular, note that "R152-GeM (GLD) [46]" uses not only the same training set, but also the same exact scales in the image pyramid; even if our method is much cheaper, it consistently outperforms others.

For setup (C), we use both global and local features. For large-scale databases, it may be impractical to store all raw local features in memory; to alleviate such requirement, we also present a variant, DELG*, where we store local features in binarized format, by simply applying an elementwise function: $b(x) = +1$ if $x > 0, -1$ otherwise.

Local feature re-ranking boosts performance substantially for DELG, compared to only searching with global features, especially in large-scale cases: gains of up to 8.9% (in $\mathcal{R}$Oxf+1M-Hard). We also present results where local feature extraction is performed with 3 scales only, the same ones used for global features. The large-scale results are similar, providing a boost of up to 7.2%. Results for DELG* also provide large improvements, but with performance that is slightly lower than the corresponding unbinarized versions. Our retrieval results also outperform DSM [52] significantly, by more than 10% in several cases. Different from our proposed technique, the gain from spatial verification reported in their work is small, of at most 1.5% absolute. DELG also outperforms local feature aggregation results from setup (A) in 7 out of 8 cases, establishing a new state-of-the-art across the board.

GLDv2 Evaluation. Table 4 compares DELG against previous GLDv2 results, where for a fair comparison we report methods trained on GLD. DELG achieves top performance in both retrieval and recognition tasks, with local feature re-ranking providing significant boost in both cases – especially on the recognition

task (26.8% absolute improvement). Note that recent work has reported even higher performance on the retrieval task, by learning on GLDv2's training set and using query expansion techniques [63]/ensembling [61]. On the other hand, DELG's performance on the recognition task is so far the best reported single-model result, outperforming many ensemble-based methods (by itself, it would have been ranked top-5 in the 2019 challenge) [61]. We expect that our results could be further improved by re-training on GLDv2's training set.

Re-ranking Experiment. Table 5 further compares local features for re-ranking purposes. R50-DELG is compared against SIFT [28], SOSNet [56] (HPatches model, DoG keypoints) and D2-Net [14] (trained, multiscale). All methods are given the same retrieval short list of 100 images for re-ranking (based on R50-DELG-global retrieval); for a fair comparison, all methods use 1k features and 1k RANSAC iterations. We tuned matching parameters separately for each method: whether to use ratio test or distance threshold for selecting correspondences (and their associated thresholds); RANSAC residual threshold; minimum number of inliers (below which we declare no match). SIFT and SOSNet provide little improvement over the global feature, due to suboptimal feature detection based on our observation (i.e., any blob-like feature is detected, which may not correspond to landmarks). D2-Net improves over the global feature, benefiting from a better feature detector. DELG outperforms other methods by a large margin.

Latency and Memory, Qualitative Results. Please refer to the appendices for a comparison of latency and memory requirements for different methods, and for qualitative results.

5 Conclusions

Our main contribution is a unified model that enables joint extraction of local and global image features, referred to as DELG. The model is based on a ResNet backbone, leveraging generalized mean pooling to produce global features and attention-based keypoint detection to produce local features. We also introduce an effective dimensionality reduction technique that can be integrated into the same model, based on an autoencoder. The entire network can be trained end-to-end using image-level labels and does not require any additional post-processing steps. For best performance, we show that it is crucial to stop gradients from the attention and autoencoder branches into the network backbone, otherwise a suboptimal representation is obtained. We demonstrate the effectiveness of our method with comprehensive experiments, achieving state-of-the-art performance on the Revisited Oxford, Revisited Paris and Google Landmarks v2 datasets.

References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the CVPR (2016)

2. Araujo, A., Norris, W., Sim, J.: Computing receptive fields of convolutional neural networks. Distill (2019). https://distill.pub/2019/computing-receptive-fields
3. Avrithis, Y., Tolias, G.: Hough pyramid matching: speeded-up geometry re-ranking for large scale image retrieval. Int. J. Comput. Vision **107**(1), 1–19 (2013). https://doi.org/10.1007/s11263-013-0659-3
4. Babenko, A., Lempitsky, V.: Aggregating local deep features for image retrieval. In: Proceedings of the ICCV (2015)
5. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 584–599. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_38
6. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key.Net: keypoint detection by handcrafted and learned CNN filters. In: Proceedings of the ICCV (2019)
7. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). CVIU **110**(3), 346–359 (2008)
8. Chang, C., Yu, G., Liu, C., Volkovs, M.: Explore-exploit graph traversal for image retrieval. In: Proceedings of the CVPR (2019)
9. Chopra, S., Hadsell, R., LeCun, Y.: Learning a dimilarity metric discriminatively, with application to face verification. In: Proceedings of the CVPR (2005)
10. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: Proceedings of the ICCV (2007)
11. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: Proceedings of the CVPR (2019)
12. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: self-supervised interest point detection and description. In: Proceedings of the CVPR Workshops (2018)
13. Dugas, C., Bengio, Y., Nadeau, C., Garcia, R.: Incorporating second-order functional knowledge for better option pricing. In: Proceedings of the NIPS (2001)
14. Dusmanu, M., et al.: D2-Net: a trainable CNN for joint detection and description of local features. In: Proceedings of the CVPR (2019)
15. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
16. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: learning global representations for image search. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 241–257. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_15
17. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. Int. J. Comput. Vision **124**(2), 237–254 (2017). https://doi.org/10.1007/s11263-017-1016-8
18. Gordo, A., Rodriguez-Serrano, J.A., Perronin, F., Valveny, E.: Leveraging category-level labels for instance-level image retrieval. In: Proceedings of the CVPR (2012)
19. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: Proceedings of the CVPR (2018)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the CVPR (2016)
21. Hinton, G.: Connectionist learning procedures. Artif. Intell. **40**(1–3), 185–234 (1989)
22. Iscen, A., Tolias, G., Avrithis, Y., Furon, T., Chum, O.: Efficient diffusion on region manifolds: recovering small objects with compact CNN representations. In: Proceedings of the CVPR (2017)

23. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: the benefit of PCA and whitening. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 774–787. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_55
24. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_24
25. Jégou, H., Douze, M., Schmidt, C., Perez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of the CVPR (2010)
26. Jégou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1704–1716 (2012)
27. Jegou, H., Zisserman, A.: Triangulation embedding and democratic aggregation for image search. In: Proceedings of the CVPR (2014)
28. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
29. Luo, Z., et al.: ContextDesc: local descriptor augmentation with cross-modality context. In: Proceedings of the CVPR (2019)
30. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004)
31. Mikolajczyk, K., Matas, J.: Improving descriptors for fast tree matching by optimal linear projection. In: Proceedings of the ICCV (2007)
32. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47969-4_9
33. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: local descriptor learning loss. In: Proceedings of the NIPS (2017)
34. Mishkin, D., Radenović, F., Matas, J.: Repeatability is not enough: learning affine regions via discriminability. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11213, pp. 287–304. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01240-3_18
35. Mukundan, A., Tolias, G., Bursuc, A., Jégou, H., Chum, O.: Understanding and improving kernel local descriptors. Int. J. Comput. Vision **127**(11), 1723–1737 (2018). https://doi.org/10.1007/s11263-018-1137-8
36. Mukundan, A., Tolias, G., Chum, O.: Explicit spatial encoding for deep local descriptors. In: Proceedings of the CVPR (2019)
37. Ng, T., Balntas, V., Tian, Y., Mikolajczyk, K.: SOLAR: second-order loss and attention for image retrieval. In: Proceedings of the ECCV (2020)
38. Nistér, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the CVPR (2006)
39. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the ICCV (2017)
40. Obdrzalek, S., Matas, J.: Sub-linear indexing for large scale object recognition. In: Proceedings of the BMVC (2005)
41. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: LF-Net: learning local features from images. In: Proceedings of the NIPS (2018)
42. Ozaki, K., Yokoo, S.: Large-scale landmark retrieval/recognition under a noisy and diverse dataset. arXiv:1906.04087 (2019)

43. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the CVPR (2007)
44. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases. In: Proceedings of the CVPR (2008)
45. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting Oxford and Paris: large-scale image retrieval benchmarking. In: Proceedings of the CVPR (2018)
46. Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1655–1668 (2018)
47. Revaud, J., Almazan, J., de Rezende, R.S., de Souza, C.R.: Learning with average precision: training image retrieval with a listwise loss. In: Proceedings of the ICCV (2019)
48. Revaud, J., Souze, C.D., Weinzaepfel, P., Humenberger, M.: R2D2: repeatable and reliable detector and descriptor. In: Proceedings of the NeurIPS (2019)
49. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale. In: Proceedings of the CVPR (2019)
50. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the CVPR (2015)
51. Silberman, N., Guadarrama, S.: TensorFlow-Slim Image Classification Model Library (2016). https://github.com/tensorflow/models/tree/master/research/slim
52. Simeoni, O., Avrithis, Y., Chum, O.: Local features and visual words emerge in activations. In: Proceedings of the CVPR (2019)
53. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: Proceedings of the ICCV (2003)
54. Taira, H., et al.: InLoc: indoor visual localization with dense matching and view synthesis. In: Proceedings of the CVPR (2018)
55. Teichmann, M., Araujo, A., Zhu, M., Sim, J.: Detect-to-retrieve: efficient regional aggregation for image search. In: Proceedings of the CVPR (2019)
56. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: SOSNet: second order similarity regularization for local descriptor learning. In: Proceedings of the CVPR (2019)
57. Tolias, G., Avrithis, Y., Jégou, H.: Image search with selective match kernels: aggregation across single and multiple images. Int. J. Comput. Vis. **116**, 247–261 (2016). https://doi.org/10.1007/s11263-015-0810-4
58. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: Proceeedings of the ICLR (2015)
59. Wang, F., Xiang, X., Cheng, J., Yuille, A.: NormFace: L2 hypersphere embedding for face verification. In: Proceedings of the ACM MM (2017)
60. Wang, H., et al.: CosFace: large margin cosine loss for deep dace recognition. In: Proceedings of the CVPR (2018)
61. Weyand, T., Araujo, A., Cao, B., Sim, J.: Google landmarks dataset v2 - a large-scale benchmark for instance-level recognition and retrieval. In: Proceedings of the CVPR (2020)
62. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: learned invariant feature transform. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 467–483. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_28

63. Yokoo, S., Ozaki, K., Simo-Serra, E., Iizuka, S.: Two-stage discriminative re-ranking for large-scale landmark retrieval. In: Proceedings of the CVPR Workshops (2020)
64. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53