Deep Shape Matching

Filip Radenović Giorgos Tolias Ondřej Chum

Visual Recognition Group, FEE, CTU in Prague
{filip.radenovic, giorgos.tolias, chum}@cmp.felk.cvut.cz

Abstract. We cast shape matching as metric learning with convolutional networks. We break the end-to-end process of image representation into two parts. Firstly, well established efficient methods are chosen to turn the images into edge maps. Secondly, the network is trained with edge maps of landmark images, which are automatically obtained by a structure-from-motion pipeline. The learned representation is evaluated on a range of different tasks, providing improvements on challenging cases of domain generalization, generic sketch-based image retrieval or its fine-grained counterpart. In contrast to other methods that learn a different model per task, object category, or domain, we use the same network throughout all our experiments, achieving state-of-the-art results in multiple benchmarks.

Keywords: shape matching · cross-modal recognition and retrieval

1 Introduction

Deep neural networks have recently become very popular for computer-vision problems, mainly due to their good performance and generalization. These networks have been first used for image classification by Krizhevsky *et al.* [3], then their application spread to other related problems. A standard architecture of a classification network starts with convolutional layers followed by fully connected layers. Convolutional neural networks (CNNs) became a popular choice

Fig. 1. Three examples where **shape** is the only relevant information: sketch, artwork, extreme illumination conditions. Top retrieved images from the Oxford Buildings dataset [1]: CNN with an RGB input [2] (left), and our shape matching network (right). Query images are shown with black border.

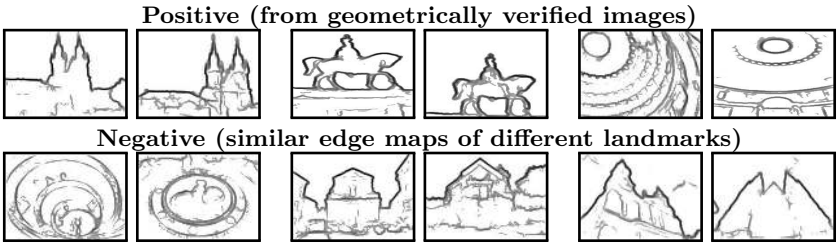

Fig. 2. Edge maps extracted from matching and non-matching image pairs that serve as training data for our network.

of learning image embeddings, *e.g.* in efficient image matching – image retrieval. It has been observed that the convolutional part of the classification network captures well *colours*, *textures* and *structures* within the receptive field.

In a number of problems, the colour and/or the texture is not available or misleading. Three examples are shown in Figure 1. For sketches or outlines, there is no colour or texture available at all. For artwork, the colour and texture is present, but often can be unrealistic to stimulate certain impression rather than exactly capture the reality. Finally, in the case of extreme illumination changes, such as a day-time versus night images, the colours may be significantly distorted and the textures weakened. On the other hand, the image discontinuities in colour or texture, as detected by modern edge detectors, and especially their shapes, carry the information about the content, independent of, or insensitive to, the illumination changes, artistic drawing and outlining.

This work is targeting at shape matching, in particular the goal is to extract a descriptor that captures the shape depicted in the input. The shape descriptors are extracted from image edge maps by a CNN. Sketches, black and white line drawings or cartoons are simply considered as a special type of an edge map.

The network is trained without any human supervision or image, sketch or shape annotation. Starting from a pre-trained classification network stripped off the fully connected layers, the CNN is fine-tuned using a simple contrastive loss function. Matching and non-matching training pairs are extracted from automatically generated 3D models of landmarks [2]. Edge maps detected on these images provide training data for the network. Examples of positive and negative pairs of edge maps are shown in Figure 2.

We show the importance of shape matching on two problems: 1) modality invariant representation, *i.e.* classification for domain generalization, and 2) cross modality matching of sketches to images.

In the domain generalization, some of the domains are available, but some are completely unseen during the training phase. We evaluate on domain generalization by performing object recognition. We extract the learned descriptors and train a simple classifier on the seen domains, which is later used to classify images of the unseen domain. We show, that for some combinations of seen-unseen domains, such as artwork and photograph, descriptors using colour and texture are useful. However, for some combinations, such as photograph and line

drawing, the shape information is crucial. Combining both types of descriptors outperforms the state-of-the-art approach in all settings.

In the cross modality matching task, it is commonly assumed that annotated training data is available for both modalities. Even for this task, we apply the domain generalization approach, using the descriptors learned on edge maps of building images. We evaluate the cross modality matching on sketch based image retrieval datasets. Modern sketch-based image retrieval take the path of object recognition from human sketches. Rather than performing shape matching, the networks are trained to recognize simplified human drawings. Such an approach requires very large number of annotated images and drawn sketches for each category of interest. Even though the proposed network is *not trained* to recognize human-drawn object sketches, our experiments show that it performs well on standard benchmarks.

2 Related work

Shape matching is shown useful in several computer vision tasks such as object recognition [4], object detection [5], 3D shape matching [6,7] and cross-modal retrieval [8,9]. In this section we review prior work related to sketch-based image retrieval, a particular flavor of cross-modal retrieval, where we apply the proposed representation. Finally, we discuss domain generalization approaches since our method is directly applicable on this problem handling it simply by learning shape matching.

Sketch-based image retrieval has been, until recently, handled with hand-crafted descriptors [10,11,12,13,14,15,16,17,18,19]. Deep learning methods have been applied to the task of sketch-based retrieval [20,21,22,23,24,25,26] much later than to the related task of image retrieval. We attribute the delay to the fact that the training data acquisition for sketch-based retrieval is much more tedious compared to image-based retrieval because it not only includes labeling the images, but also sketches must be drawn in large quantities. Methods with no learning typically carry no assumptions on the depicted categories, while the learning based methods often include category recognition into training. The proposed method aims at generic sketch-based retrieval, not limited to a fixed set of categories; it is, actually, not even limited to objects.

Learning-free methods have followed the same initial steps as in the traditional image search. These include the construction of either global [27,12,16] or local [28,29,11,30,14] image and/or sketch representations. Local representations are also using vector quantization to create a Bag-of-Words model [31]. Further cases are symmetry-aware and flip invariant descriptors [30], descriptors that are based on local contours [29] or line segments [14], and kernel descriptors [19]. Transformation invariance is often sacrificed for the sake of scalability [8,9]. In contrast, the method proposed in this paper is fully translation invariant, and scalable, because it reduces to a nearest-neighbor search in a descriptor space.

Learning-based methods require annotated data in both domains, typically for a fixed set of object categories, making the methods [6,20,21,22,23,24,25,26,32] to be category specific. End-to-end learning methods are applied to both category

level [25,26] and to fine-grained, *i.e.* sub-category level retrieval [22,23,24,32], while sometimes a different model per category has to be learned [33,22,34,32]. A sequence of different learning and fine-tuning stages is applied [22,23,24,25,26], involving massive manual annotation effort. For example, the Sketchy dataset [23] required collectively 3,921 hours of sketching. On the contrary, our proposed fine-tuning does not require any manual annotation.

Domain generalization is handled in a variety of ways, ranging from learning domain invariant features [35,36] to learning domain invariant classifiers [37,38] or both [39,40]. Several methods focus on one-way shift between two domains, such as sketch-based retrieval described earlier or learning on real photos and testing on art [41,42]. An interesting benchmark is released in the work of Li *et al.* [39], where four domains of increasing visual abstraction are used, namely *photos*, *art*, *cartoon*, and *sketches* (PACS). Prior domain generalization methods [35,36,37] are shown effective on PACS, while simply training a CNN on all the available (seen) domains is a very good baseline [39]. We tackle this problem from the representation point of view and focus on the underlined shapes. Our shape descriptor is extracted and the labels are used only to train a linear classifier. In this fashion, we are able to train on a single domain and test on all the rest, while common domain generalization approaches require different domains present in the training set.

3 Method

In this section we describe the proposed approach. The process of fine-tuning the CNN is described in Section 3.1, while the final representation and the way it is used for retrieval and classification is detailed in Section 3.2.

We break the end-to-end process of image description into two parts. In the first part, the images are turned into edge maps. In particular, throughout all our experiments we used the edge detector of Dollár and Zitnick [43] due to its great trade-off between efficiency and accuracy, and the tendency not to consider textured regions as edges. Our earlier experiments on sketch-based image retrieval with a CNN-based edge detector [44] did not show any significant changes in the performance. An image is represented as an edge map, which is a 2D array containing the edge strength in each image pixel. The edge strength is in the range of $[0, 1]$, where 0 represents background. Sketches, in the case of sketch-to-image retrieval, are represented as a special case of an edge map, where the edge strength is either 0 for the background or 1 for a contour.

The second part is a fully convolutional network extracting a global image descriptor. The two part approach allows, in a simple manner, to unify all modalities at the level of edge maps. Jointly training these two parts, *e.g.* in the case of a CNN-based edge detector [44], can deliver an image descriptor too. However, this descriptor may not be based on shapes. It is unlikely that such an optimization would end in a state where the representation between the two parts actually corresponds to edges. Enforcing this with additional training data in the form of edge maps and a loss on the output of the first part is exactly what we are avoiding and improving in this work.

3.1 Training

We use a network architecture previously proposed for image classification [45], in particular, we use all convolutional layers and the activations of the very last one, *i.e.*, the network is stripped of the fully-connected layers. The CNN is initialized by the parameters learned on a large scale annotated image dataset, such as ImageNet [46]. This is a fairly standard approach adopted in a number of problems, including image search [47,2,48]. The network is then fine-tuned with pairs of image edge maps.

The network. The image classification network expects an RGB input image, while the edge maps are only two dimensional. We sum the first convolution filters over RGB. Unlike in RGB input, no mean pixel subtraction is performed to the input data. To obtain a compact, shift invariant descriptor, a global max-pooling [49] layer is appended after the last convolutional layer. This approach is also known as Maximum Activations of Convolutions (MAC) vector [50]. After the MAC layer, the vectors are ℓ_2 normalized.

Edge filtering. A typical output of edge detectors is a strength of an edge in every pixel. We introduce an edge filtering layer to address two frequent issues with edge responses. First, the background often contains close-to-zero responses, which typically introduce noise into the representation. This issue is commonly handled by thresholding the response function. Second, the strength of the edges provides ordering, *i.e.* higher edge response implies that the edge is more likely to be present, however its value typically does not have practical interpretation. Prior to the first convolution layer, a continuous and differentiable function is pre-pended. This layer is trained together with the rest of the network to transform the edge detector output with soft thresholding by a sigmoid and power transformation. Denote the edge strength by $w \in [0, 1]$. Edge filtering is performed as

$$f(w) = \frac{w^p}{1 + e^{\beta(\tau - w)}}, \tag{1}$$

where p controls the contrast between strong and weak edges, τ is the threshold parameter, and β is the scale of the sigmoid choosing between hard thresholding and a softer alternative. The final function (1) with learned parameters is plotted

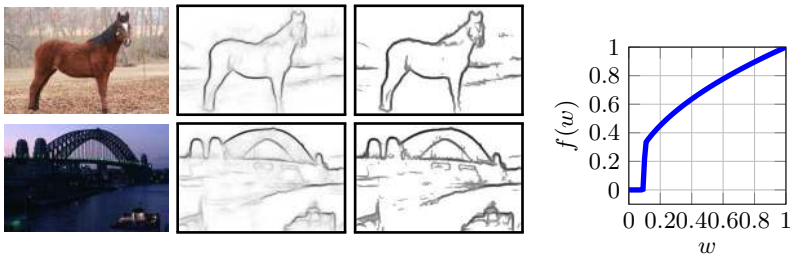

Fig. 3. Sample images, the output of the edge detector, the filtered edge map, and the edge-filtering function.

in Figure 3 (right). The figure also visually demonstrates the effect of application of the filtering. The weak edges are removed on the background and the result appearance is closer to a rough sketch, while the uncertainty in edges is still preserved.

Fine tuning. The CNN is trained with Stochastic Gradient Descent in a Siamese fashion with contrastive loss [51]. The positive training pairs are edge maps of matching images (similarity of the edge maps is not considered), while the negative pairs are similar edge maps (according to the current state of the network) of non-matching images.

Given a pair of vectors $\mathbf{x}$ and $\mathbf{y}$, the loss is defined as their squared Euclidean distance $||\mathbf{x}-\mathbf{y}||^2$ for positive examples, and as $\max\{(m-||\mathbf{x}-\mathbf{y}||)^2, 0\}$ for negative examples. Hard-negative mining is performed several times per epoch which has been shown to be essential [2,48].

Training data. The training images for fine tuning the network are collected in a fully automatic way. In particular, we use the publicly available dataset used in Radenovic *et al.* [2] and follow the same methodology, briefly reviewed in the following. A large unordered image collection is passed through a 3D reconstruction system based on local features and Bag-of-Words retrieval [52,53]. The outcome consists of a set of 3D models which mostly depict outdoor landmarks and urban scenes. For each landmark, a maximum of 30 six-tuples of images are being selected. The six-tuple consists of: one image as the training query, then one matching image to the training query, and five similar non-matching images. This gives arise to one positive and five negative pairs. The geometry of the 3D models, including camera positions, allows to mine matching images, *i.e.* those that share adequate visual overlap. Negative-pair mining is facilitated by the 3D models, too: negative images are chosen only if they belong to a different model.

Data augmentation. A standard data-augmentation, *i.e.* random horizontal flipping (mirroring) procedure is applied to introduce further variance in the training data and to avoid over-fitting. The training query and the positive example are jointly mirrored with 50% probability. Negative examples are sought after eventual flipping. We propose an additional augmentation technique for the selected training queries. Their edge map responses are thresholded with a random threshold uniformly chosen from $[0, 0.2]$ and the result is binarized. Matching images (in positive examples) are left unchanged; negative images are selected after the transformation. This augmentation process is applied with a probability of 50%. It offers a level of shape abstraction and mimics the asymmetry of sketch-to-edge map matching. The randomized threshold can be also seen as an approximation of the stroke removal in [22].

3.2 Representation, Search and Classification

We use the trained network to extract image and sketch descriptors capturing the underlying shapes, which are then used to perform cross-modal image retrieval, in particular sketch-based, and object recognition via transfer learning, in particular domain generalization.

Representation. The input to the descriptor extraction process is always resized to a maximum dimensionality of 227×227 pixels. A multi-scale representation is performed by processing at 5 fixed scales, *i.e.*, re-scaling the original input by a factor of $1/2$, $1/\sqrt{2}$, 1, $\sqrt{2}$, 2, and, with the additional mirroring, 10 final instances are produced. *Images* undergo edge detection and the resulting edge map [43] is fed to the CNN[1]. *Sketches* come in the form of strokes, thin line drawings, or brush drawings, depending on the input device or the dataset. To unify the sketch input, a simple morphological filter is applied to a binary sketch image. Specifically, a morphological thinning followed by dilation is performed. After the pre-processing, the sketch is treated as an edge map. As a consequence of the rescaling and mirroring, an image/sketch is mapped to 10 high dimensional vectors. We refer to these ℓ_2 normalized vectors as EdgeMAC descriptors. They are subsequently sum-aggregated or indexed separately, depending on the evaluation benchmark, see Section 4 for more details.

Search. An image collection is indexed by simply extracting and storing the corresponding EdgeMAC descriptors for each image. Search is performed by nearest-neighbors search of the query descriptor in the database. This makes retrieval compatible with approximate methods [54,55] that can speed up search and offer memory savings.

Classification. We extract EdgeMAC descriptors from labeled images and train a multi-class linear classifier [56] to perform object recognition. This is especially useful for transfer learning when the training domain is different from the target/testing one. In this case, no labeled images of the training domain are available during the training of our network and no labeled images of the target domain are available during classifier training.

3.3 Implementation details

In this section we discuss implementation details. The training dataset used to train our network is presented. We train a single network, which is then used for different tasks. Training sets provided for specific tasks are not exploited.

Training data. We use the same training set as in the work of Radenovic *et al.* [2][2] which comprises landmarks and urban scenes. There are around 8k tuples. Due to the overlap of landmarks contained in the training set and one of the test sets involved in our evaluation, we manually excluded these landmarks from our training data. We end up with with 5,969 tuples for training and 1,696 for validation. Hard negatives are re-mined 3 times per epoch [2] from a pool of around 22k images.

Training implementation. We use the MatConvNet toolbox [57] to implement the learning. We initialize the convolutional layers by VGG16 [45] (results in 512D EdgeMAC descriptor) trained on ImageNet and sum the filters of the first

[1] We perform zero padding by 30 pixels to avoid border effects.
[2] Training data available at cmp.felk.cvut.cz/cnnimageretrieval

layer over the feature maps dimension to accommodate for the 2D edge map input instead of the 3D image. The edge-filtering layer is initialized with values $p = 0.5$, $\tau = 0.1$ and β is fixed and equal to 500 so that it always approximates hard thresholding. Additionally, the output of the egde-filtering layer is linearly scaled from $[0, 1]$ to $[0, 10]$. Initial learning rate is $l_0 = 0.001$ with an exponential learning rate decay $l_0 \exp(-0.1j)$ over epoch j; momentum is 0.9; weight decay is 0.0005; contrastive loss margin is 0.7; and batch size is equal to 20 training tuples. All training images are resized so that the maximum extent is 200 pixels, while keeping the original aspect ratio.

Training time. Training is performed for at most 20 epochs and the best network is chosen based on the performance on validation tuples. The whole training takes about 10 hours on a single GeForce GTX TITAN X (Maxwell) GPU with 12GB of memory.

4 Experiments

We evaluate EdgeMAC descriptor on domain generalization and sketch-based image retrieval. We train a single network and apply it on both tasks proving the generic nature of the representation.

4.1 Domain Generalization through Shape Matching

We extract EdgeMAC descriptors from labeled images, sum-aggregate descriptors of rescaled and mirrored instances and ℓ_2 normalize to produce one descriptor per image, and train a linear classifier [56] to perform object recognition. We evaluate on domain generalization to validate the effectiveness of our representation on shape matching.

PACS dataset was recently introduced by Li *et al.* [39]. It consists of images coming from 4 domains with varying level of abstraction, namely, *art* (painting), *cartoon*, *photo*, and *sketch*. Images are labeled according to 7 categories, namely, dog, elephant, giraffe, guitar, horse, house, and person. Each time, one domain is considered unseen, otherwise called target or test domain, while the image of the other 3 are used for training. Finally, multi-class accuracy is evaluated on the unseen domain. In our work, we additionally perform classifier training using a single domain and then test on the rest. We find this scenario to be realistic, especially in the case of training on photos and testing on the rest. The domain of realistic photos is the richest in terms of annotated data, while others such as sketches and cartoons are very sparsely annotated.

Baselines. We are interested in translation invariant representations and consider the two following baselines. First, MAC [50] descriptors extracted using a network that is pre-trained on ImageNet. Second, MAC descriptors extracted by a network that is fine-tuned for image retrieval in a siamese manner [2]. These two baselines have the same descriptor extraction complexity as ours. They are extracted on RGB images, while ours on edge maps. Note, that we treat all

Table 1. Multi-class accuracy on PACS dataset for 4 different descriptors. The combined descriptor (pre-trained + ours) is constructed via concatenation. A: Art, C: Cartoon, P: Photo, S: Sketch, 3: all 3 other domains.

	Pre-trained (RGB)				Siamese [2] (RGB)				Ours (edge map)				Pre-trained+Ours			
Test →	A	C	P	S	A	C	P	S	A	C	P	S	A	C	P	S
Train A	N/A	59.2	95.0	33.1	N/A	59.5	86.3	42.9	N/A	55.9	61.2	65.6	N/A	61.6	94.9	38.4
Train C	71.7	N/A	86.8	37.0	61.0	N/A	77.0	51.6	45.2	N/A	57.3	74.8	69.3	N/A	85.0	55.3
Train P	72.5	33.3	N/A	24.8	66.0	38.0	N/A	31.9	45.4	42.3	N/A	46.3	73.3	34.0	N/A	27.61
Train S	31.9	49.5	42.5	N/A	38.7	49.3	44.4	N/A	34.8	63.0	43.3	N/A	33.7	59.3	43.4	N/A
Train 3	78.0	68.0	94.4	47.1	71.5	64.3	85.1	56.0	53.8	67.9	64.5	74.7	80.0	68.7	93.7	62.7
Mean 3		71.9				69.2				65.2				76.2		

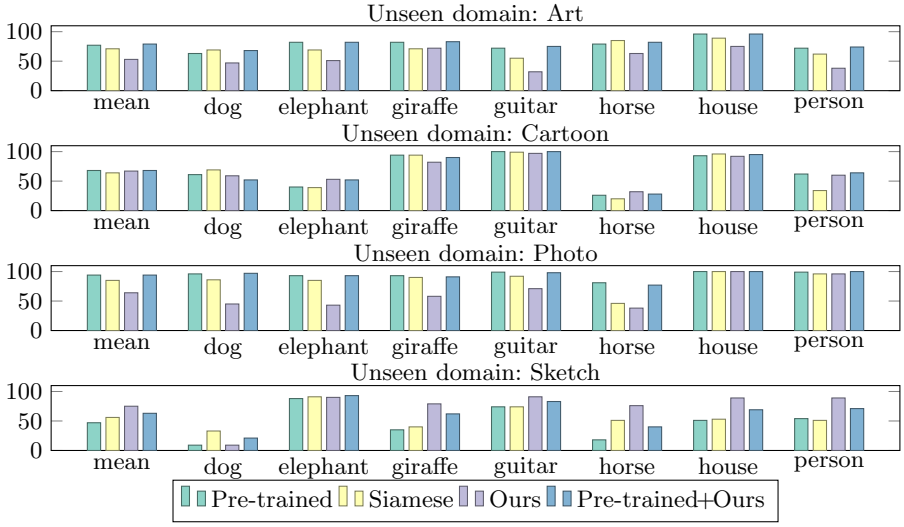

Fig. 4. Classification accuracy on PACS dataset with different descriptors. Testing is performed on 1 unseen domain each time, while training is performed on the other 3.

domains as images with our approach and extract edge maps, *i.e.* we do not perform any special treatment on sketches as in the case of sketch retrieval.

Performance comparison. We evaluate our descriptor, the two baselines, and the concatenated version of ours and the descriptor of the pre-trained baseline network, and report results in Table 1. Our representation significantly improves sketch recognition while training on a single or all seen domains. Similar improvements are observed for cartoon recognition when training on photos or sketches, while when training on artwork the color information appears to be beneficial. We consider the case of training only on photos and testing on other domains to be the most interesting and realistic one. In this scenario, we provide improvements, compared to the baselines, for sketch recognition (15% and 22%) and cartoon recognition (4% and 9%). Finally, the combined descriptor reveals the complementarity of the representations in several cases, such as artwork and cartoon recognition while training on all seen domains, or training on single domain when artwork is involved, *e.g.*. train on P (or A) and test on A (or C). The

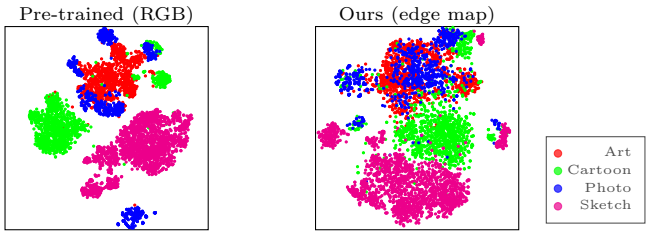

Fig. 5. Visualization of PACS images with t-SNE (more overlap is better).

best reported score on PACS is 69.2 [39] by fine-tuning AlexNet on PACS. The achieved score by our descriptor with fine-tuned VGG (PACS not used during network training) is 76.2, which is significantly higher. The same experiment with AlexNet achieves 70.9. Performance is reported per category in Figure 4. Our descriptor achieves significant improvements on most categories for sketch recognition, while the combined is a safe choice in several cases. Interestingly, our experiments reveal that the siamese baseline slightly improves shape matching, despite being trained on RGB images.

Visualization with t-SNE. We use t-SNE [58] to reduce the dimensionality of descriptors to 2 and visualize the result for the pre-trained baseline and our descriptor in Figure 5. The different modalities are brought closer with our descriptor. Observe how separated is the sketch modality with the pre-trained network that receives an RGB image for input.

4.2 Sketch-based Image Retrieval

We extract EdgeMAC descriptors to index an image collection, treat sketch queries as described in Section 3.2 and perform sketch-based image retrieval via simple nearest neighbor search.

Test datasets and evaluation protocols. The method is evaluated on two standard sketch-based image retrieval benchmarks.

Flickr15k [11] consists of 15k database images and 330 sketch queries that are related to 33 categories. Categories include particular object instances (Brussels Cathedral, Colosseum, Arc de Triomphe, *etc.*), generic objects (airplane, bicycle, bird, *etc.*), and shapes (circle shape, star shape, heart, balloon, *etc.*). The performance is measured via mean Average Precision (mAP) [1]. We sum-aggregate descriptors of rescaled and mirrored instances and ℓ_2 normalize to produce one descriptor per image. Search is performed by a cosine similarity nearest-neighbor search.

Shoes/Chairs/Handbags [22,32] datasets contain images from one category only, *i.e.* shoe/chair/handbag category respectively. It consists of pairs of a photo and a corresponding hand-drawn detailed sketch of this photo. There are 419, 297, and 568 sketch–photo pairs of shoes, chairs, and handbags, respectively. Out of these, 304, 200, and 400 pairs are selected for training, and 115, 97, and 168 for testing shoes, chairs, and handbags, respectively. The performance

Table 2. Performance evaluation of the different components of our method on Flickr15k dataset. Network: off-the-shelf (O), fine-tuned (F).

Component	Network							
	O	O	F	F	F	F	F	F
Train/Test: Edge filtering		■	■	■	■	■	■	■
Train: Query binarization				■	■	■	■	■
Test: Mirroring					■		■	■
Test: Multi-scale						■	■	■
Test: Diffusion								■
mAP	25.9	27.9	38.4	41.9	43.4	45.6	46.1	68.9

is measured via the matching accuracy at the top K retrieved images, denoted by acc.@K. The underlying task is quite different compared to Flickr15k. The photograph used to generate the sketch is to be retrieved, while all other images are considered false positives. The search protocol used in [22] is as follows: Descriptors are extracted from 5 image crops (corners and center) and their horizontally mirrored counterparts. This holds for database images and the sketch query. During search, these 10 descriptors are compared one-to-one and their similarity is averaged. For fair comparison, we adopt this protocol and do not use a single descriptor per image/sketch for this benchmark. However, instead of image crops, we extract descriptors at 5 image scales and their horizontally mirrored counterparts, as these are defined in Section 3.2.

Impact of different components. Table 2 shows the impact of different components on the final performance of the proposed method as measured on Flickr15k dataset. Direct application of the off-the-shelf CNN on edge maps already outperforms most prior hand-crafted methods (see Table 3). Adding the edge-filtering layer to the off-the-shelf network improves the precision. The initial parameters for filtering are used. Fine-tuning brings significant jump to 38.4 mAP, which is already the state-of-the-art on this dataset. Random training-query binarization and multi-scale with mirroring representation further improve the mAP score to 46.1. Finally, we boost the recall of our sketch-based retrieval by global diffusion, as recently proposed by Iscen *et al.* [59]. We construct the neighborhood graph by combining kNN-graphs built on two different similarities [60,61]: edge map similarity with EdgeMAC and image similarity with CroW descriptors [62]. This increases the performance to 68.9.

Performance evolution during learning. We report the performance of the fine-tuned network at different stages (epochs) of training. The same network is evaluated for all datasets as we train a single network for all tasks. The performance is shown in Figure 6 for both benchmarks. On all datasets, the fine-tuning significantly improves the performance already from the first few epochs.

As a sanity check, we also perform a non-standard sketch-to-sketch evaluation. On the Flickr15k dataset, each of the 330 sketches is used to query the other 329 sketches (the query sketch is removed from the evaluation), which attempts to retrieve sketches of the same category. The evolution of the performance shows similar behavior as the sketch-to-image search, *i.e.*, the learning on edge maps improves the performance on sketch-to-sketch retrieval, see Figure 6.

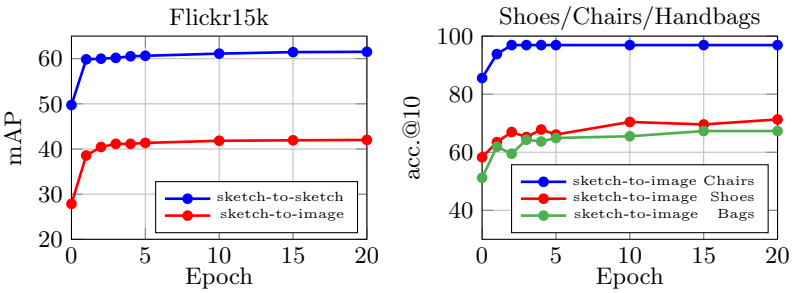

Fig. 6. Performance evaluation of the fine-tuned network over training epochs for the single-scale representation. All shown datasets and their evaluation protocols are described in Section 4.2.

Comparison with the state of the art. We extensively compare our method to the state-of-the-art performing methods on both benchmarks. Whenever code and trained models are publicly available, we additionally evaluate them on test sets they were not originally applied on. In cases that the provided code is used for evaluation on Flickr15k we center and align the sketches appropriately in order to achieve high scores, while our method is translation invariant so there is no such need. First we give a short overview of the best performing and most relevant methods to ours. Finally, a comparison via quantitative results is given.

Shoes/Chairs/Handbags networks [22,32] are trained from scratch based on the Sketch-a-Net architecture [63]. This is achieved by the following steps [22][3]: (i) Training with classification loss for 1k categories from ImageNet-1K data with edge maps input. (ii) Training with classification loss for 250 categories of TU-Berlin [64] sketch data. (iii) Training a triplet network with shared weights and ranking loss on TU-Berlin sketches and ImageNet images. (iv) Finally, training separate networks for fine-grain instance-level ranking using the Shoes/Chairs/ /Handbags training datasets. This approach is later improved [32] by adding an attention module with a coarse-fine fusion (CFF) into the architecture, and by extending the triplet loss with a higher order learnable energy function (HOLEF). Such a training involves various datasets, with annotation at different levels, and a variety of task-engineered loss functions. Note that the two models available online achieve higher performance than the ones reported in [22], due to parameter retuning. We compare our results to their best performing models.

Sketchy network [23] consists of two asymmetric sketch and image branches, both initialized with GoogLeNet. The training involves the following steps[4]: (i) Training for classification on TU-Berlin sketch dataset. (ii) Separate training of the sketch branch with classification loss on 125 categories of Sketchy dataset and training of the image branch with classification loss on the same categories with additional 1000 Flickr photos per category. (iii) Training both branches in a triplet network with ranking loss on the Sketchy sketch–photo pairs. The last part involves approximately 100k positive and a billion negative pairs.

[3] Networks/code available at github.com/seuliufeng/DeepSBIR
[4] Network/code available at github.com/janesjanes/sketchy

Table 3. Performance comparison via mean Average Precision (mAP) with the state-of-the-art sketch-based image retrieval on the Flickr15k dataset. Best result is highlighted in **red**, second best in **bold**. Query expansion methods are shown below the horizontal line and are highlighted separately. Our evaluation of the methods that do not originally report results on Flickr15 is marked with †.

Hand-crafted methods

Method	Dim	mAP
GF-HOG [11]	n/a	12.2
S-HELO [12]	1296	12.4
HLR+S+C+R [14]	n/a	17.1
GF-HOG extended [15]	n/a	18.2
PerceptualEdge [16]	3780	18.4
LKS [17]	1350	24.5
AFM [19]	243	30.4
AFM+QE [19]	755	**57.9**

CNN-based methods

Method	Dim	mAP
Sketch-a-Net+EdgeBox [20]	5120	27.0
Shoes network [22]†	256	29.9
Chairs network [22]†	256	29.8
Sketchy network [23]†	1024	34.0
Quadruplet network [24]	1024	32.2
Triplet no-share network [26]	128	**36.2**
⋆ EdgeMAC	512	**46.1**
Sketch-a-Net+EdgeBox+GraphQE [20]	n/a	32.3
⋆ EdgeMAC+Diffusion	n/a	**68.9**

Quadruplet network [24] tackles the problem in a similar way as Sketchy network, however, they use ResNet-18 [65] architecture with shared weights for both sketch and image branches. The training involves the following steps: (i) Training with classification loss on Sketchy dataset. (ii) Training a network with triplet loss on Sketchy dataset, while mining three different types of triplets.

Triplet no-share network [26] consists of asymmetric sketch and image branches initialized by Sketch-a-Net and AlexNet [3], respectively. The training involves: (i) Separate training of the sketch branch with classification loss on TU-Berlin and training of the image branch with classification loss on ImageNet. (ii) Training a triplet network with ranking loss on TU-Berlin sketches augmented with 25k corresponding photos harvested from the Internet. (iii) Training a triplet network with ranking loss on the Sketchy dataset.

Performance comparison. We compare our network with other methods on both benchmarks. Methods that have not reported scores on a particular datasets are evaluated by ourselves while using the publicly available networks.

Results on the Flickr15k dataset are presented in Table 3, where our method significantly outperforms both hand-crafted descriptors and CNN-based that are learned on a variety of training data. This holds for both plain search with the descriptors, and for methods using re-ranking techniques, such as query expansion [66] and diffusion [59].

Results on the fine-grained Shoes/Chairs/Handbags benchmark are shown in Table 4. In this experiment, we also report the performance after applying descriptor whitening which is learned in a supervised way [2] by using the descriptors of the training images of this benchmark. A single whitening transformation is learned for all three datasets. Such a process takes only a few seconds once descriptors are given. It is orders of magnitude faster than using the training set to perform network fine-tuning. We achieve the top performance in 2 out of 3 categories and the second best in the other one. The approach of [22] and [32] train a separate network per category (3 in total), which is clearly not scalable to many objects. In contrast our approach uses a single generic network. An additional drawback is revealed when we evaluate the publicly available Shoes

Table 4. Performance comparison via accuracy at rank K (acc.@K) with the state-of-the-art sketch-based image retrieval on the Shoes/Chairs test datasets. Best result is highlighted in **red**, second best in **bold**. Note that [22] and [32] train a separate network per object category. †We evaluate the publicly available networks, because the performance is higher than the one originally reported in [22].

Method	Dim	Shoes		Chairs		Handbags	
		acc.@1	acc.@10	acc.@1	acc.@10	acc.@1	acc.@10
BoW-HOG + rankSVM [22]	500	17.4	67.8	28.9	67.0	2.4	10.7
Dense-HOG + rankSVM [22]	200K	24.4	65.2	52.6	93.8	15.5	40.5
Sketch-a-Net + rankSVM [22]	512	20.0	62.6	47.4	82.5	9.5	44.1
CCA-3V-HOG + PCA [18]	n/a	15.8	63.2	53.2	90.3	–	–
Shoes net [22]†	256	52.2	**92.2**	65.0	92.8	23.2	59.5
Chairs net [22]†	256	30.4	75.7	72.2	**99.0**	26.2	58.3
Handbags net [32]	256	–	–	–	–	39.9	82.1
Shoes net + CFF + HOLEF [32]	512	**61.7**	**94.8**	–	–	–	–
Chairs net + CFF + HOLEF [32]	512	–	–	**81.4**	95.9	–	–
Handbags net + CFF + HOLEF [32]	512	–	–	–	–	**49.4**	**82.7**
⋆ EdgeMAC	512	40.0	76.5	**85.6**	95.9	35.1	70.8
⋆ EdgeMAC + whitening	512	**54.8**	**92.2**	**85.6**	**97.9**	**51.2**	**85.7**

and Chairs networks on the category they were not trained on. We observe a significant drop in performance, see Table 4.

The number of parameters. Our reported results use the VGG16 network stripped off the fully connected layers (FC), leaving ∼15M parameters. The number of parameters of Sketch-A-Net [63] is ∼8.5M parameters, while when used for SBIR in two different branches (Shoes, Chairs, Handbags [22]) there is ∼17M parameters. Triplet no-share network [26] uses two branches (Sketch-a-Net with additional FC layer and AlexNet [3]) leading to ∼115M, and Sketchy [23] uses 2× GoogLeNet leading to ∼26M parameters. Our network has the smallest number of parameters from the competing methods.

5 Conclusions

We have introduced an approach to learn shape matching by training a CNN with edge maps of matching images. The training stage does not require any manual annotation, achieved by following the footsteps of image retrieval [2], where image pairs are automatically mined from large scale 3D reconstruction.

The generic applicability of the representation is proven by validating on a variety of cases. It achieves state-of-the-art results on standard benchmarks for sketch-based image retrieval, while we have further demonstrated the applicability beyond sketch-based image retrieval. Promising results were achieved for queries with different modality (artwork) and significant change of illumination (day-night retrieval). The descriptor is shown beneficial for object recognition via transfer learning, especially to classify images of unseen domains, such as cartoons and sketches, where the amount of annotated data is limited. Remarkably, the same network is applied in all the different tasks. Training data, trained models, and code are publicly available at cmp.felk.cvut.cz/cnnimageretrieval.

Acknowledgements: This work was supported by the MSMT LL1303 ERC-CZ grant and the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “RCI”.

References

1. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR. (2007) 1, 10
2. Radenović, F., Tolias, G., Chum, O.: CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In: ECCV. (2016) 1, 2, 5, 6, 7, 8, 9, 13, 14
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1, 13, 14
4. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002) 3
5. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI (2008) 3
6. Wang, F., Kang, L., Li, Y.: Sketch-based 3d shape retrieval using convolutional neural networks. In: CVPR. (2015) 3
7. Tabia, H., Laga, H.: Covariance-based descriptors for efficient 3d shape matching, retrieval, and classification. IEEE Trans. on Multimedia (2015) 3
8. Cao, Y., Wang, C., Zhang, L., Zhang, L.: Edgel index for large-scale sketch-based image search. In: CVPR. (2011) 3
9. Sun, X., Wang, C., Xu, C., Zhang, L.: Indexing billions of images for sketch-based retrieval. In: ACM Multimedia. (2013) 3
10. Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M.: Sketch-based shape retrieval. ACM Trans. on Graphics (2012) 3
11. Hu, R., Collomosse, J.: A performance evaluation of gradient field hog descriptor for sketch based image retrieval. CVIU (2013) 3, 10, 13
12. Saavedra, J.M.: Sketch based image retrieval using a soft computation of the histogram of edge local orientations (S-HELO). In: ICIP. (2014) 3, 13
13. Parui, S., Mittal, A.: Similarity-invariant sketch-based image retrieval in large databases. In: ECCV. (2014) 3
14. Wang, S., Zhang, J., Han, T.X., Miao, Z.: Sketch-based image retrieval through hypothesis-driven object boundary selection with hlr descriptor. IEEE Trans. on Multimedia (2015) 3, 13
15. Bui, T., Collomosse, J.: Scalable sketch-based image retrieval using color gradient features. In: ICCV. (2015) 3, 13
16. Qi, Y., Song, Y.Z., Xiang, T., Zhang, H., Hospedales, T., Li, Y., Guo, J.: Making better use of edges via perceptual grouping. In: CVPR. (2015) 3, 13
17. Saavedra, J.M., Barrios, J.M., Orand, S.: Sketch based image retrieval using learned keyshapes (LKS). In: BMVC. (2015) 3, 13
18. Xu, P., Yin, Q., Huang, Y., Song, Y.Z., Ma, Z., Wang, L., Xiang, T., Kleijn, W.B., Guo, J.: Cross-modal subspace learning for fine-grained sketch-based image retrieval. Neurocomputing (2017) 3, 14
19. Tolias, G., Chum, O.: Asymmetric feature maps with application to sketch based retrieval. In: CVPR. (2017) 3, 13
20. Bhattacharjee, S.D., Yuan, J., Hong, W., Ruan, X.: Query adaptive instance search using object sketches. In: ACM Multimedia. (2016) 3, 13
21. Qi, Y., Song, Y.Z., Zhang, H., Liu, J.: Sketch-based image retrieval via siamese convolutional neural network. In: ICIP. (2016) 3
22. Yu, Q., Lie, F., Song, Y.Z., Xian, T., Hospedales, T., Loy, C.C.: Sketch me that shoe. In: CVPR. (2016) 3, 4, 6, 10, 11, 12, 13, 14

23. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. on Graphics (2016) 3, 4, 12, 13, 14
24. Seddati, O., Dupont, S., Mahmoudi, S.: Quadruplet networks for sketch-based image retrieval. In: ICMR. (2017) 3, 4, 13
25. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: CVPR. (2017) 3, 4
26. Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.: Generalisation and sharing in triplet convnets for sketch based visual search. In: arXiv:1611.05301. (2016) 3, 4, 13, 14
27. Chalechale, A., Naghdy, G., Mertins, A.: Sketch-based image matching using angular partitioning. IEEE Trans. on Systems, Man, and Cybernetics (2005) 3
28. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics (2010) 3
29. Riemenschneider, H., Donoser, M., Bischof, H.: Image retrieval by shape-focused sketching of objects. In: CVWW. (2011) 3
30. Cao, X., Zhang, H., Liu, S., Guo, X., Lin, L.: Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In: ICCV. (2013) 3
31. Ma, C., Yang, X., Zhang, C., Ruan, X., Yang, M.H., Coporation, O.: Sketch retrieval via dense stroke features. In: BMVC. (2013) 3
32. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: ICCV. (2017) 3, 4, 10, 12, 13, 14
33. Li, Y., Hospedales, T.M., Song, Y.Z., Gong, S.: Fine-grained sketch-based image retrieval by matching deformable part models. In: BMVC. (2014) 4
34. Song, J., Song, Y.Z., Xiang, T., Hospedales, T., Ruan, X.: Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In: BMVC. (2016) 4
35. Ghifary, M., Bastiaan Kleijn, W., Zhang, M., Balduzzi, D.: Domain generalization for object recognition with multi-task autoencoders. In: ICCV. (2015) 4
36. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML. (2013) 4
37. Xu, Z., Li, W., Niu, L., Xu, D.: Exploiting low-rank structure from latent domains for domain generalization. In: ECCV. (2014) 4
38. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: ECCV. (2012) 4
39. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: ICCV. (2017) 4, 8, 10
40. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: NIPS. (2016) 4
41. Crowley, E., Zisserman, A.: The state of the art: Object retrieval in paintings using discriminative regions. In: BMVC. (2014) 4
42. Crowley, E.J., Zisserman, A.: The art of detection. In: ECCV. (2016) 4
43. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV. (2013) 4, 7
44. Kokkinos, I.: Pushing the boundaries of boundary detection using deep learning. In: ICLR. (2016) 4
45. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2014) 5, 7
46. Dong, W., Socher, R., Li-Jia, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR. (2009) 5

47. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR. (2016) 5
48. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: ECCV. (2016) 5, 6
49. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual instance retrieval with deep convolutional networks. ITE Trans. on MTA (2016) 5
50. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. In: ICLR. (2016) 5, 8
51. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR. (2005) 6
52. Schönberger, J.L., Radenović, F., Chum, O., Frahm, J.M.: From single image query to detailed 3D reconstruction. In: CVPR. (2015) 6
53. Radenović, F., Schönberger, J.L., Ji, D., Frahm, J.M., Chum, O., Matas, J.: From dusk till dawn: Modeling in the dark. In: CVPR. (2016) 6
54. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP. (2009) 7
55. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. In: arXiv:1702.08734. (2017) 7
56. Perronnin, F., Akata, Z., Harchaoui, Z., Schmid, C.: Towards good practice in large-scale learning for image classification. In: CVPR. (2012) 7, 8
57. Vedaldi, A., Lenc, K.: MatConvNet: Convolutional neural networks for matlab. In: ACM Multimedia. (2015) 7
58. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. JMLR (2008) 10
59. Iscen, A., Tolias, G., Avrithis, Y., Furon, T., Chum, O.: Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In: CVPR. (2017) 11, 13
60. Bai, S., Sun, S., Bai, X., Zhang, Z., Tian, Q.: Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity. In: ECCV. (2016) 11
61. Zhang, S., Yang, M., Cour, T., Yu, K., Metaxas, D.N.: Query specific fusion for image retrieval. In: ECCV. (2012) 11
62. Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. In: ECCVW. (2016) 11
63. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-Net that beats humans. In: BMVC. (2015) 12, 14
64. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. on Graphics (2012) 12
65. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 13
66. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV. (2007) 13