Computers & Graphics (2018)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Sketching out the Details: Sketch-based Image Retrieval using Convolutional Neural Networks with Multi-stage Regression

Tu Bui[a,*], Leonardo Ribeiro[b], Moacir Ponti[b], John Collomosse[a]

[a]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey — Guildford, United Kingdom, GU2 7XH
[b]Institute of Mathematical and Computer Sciences (ICMC), Universidade de São Paulo — São Carlos/SP, Brazil, 13566-590

ARTICLE INFO

Article history:
Received January 2, 2018

Keywords: Sketch based image retrieval (SBIR), Deep learning, Cross-domain modelling, Compact feature representations, Multi-stage regression, Contrastive and triplet losses

ABSTRACT

We propose and evaluate several deep network architectures for measuring the similarity between sketches and photographs, within the context of the sketch based image retrieval (SBIR) task. We study the ability of our networks to generalize across diverse object categories from limited training data, and explore in detail strategies for weight sharing, pre-processing, data augmentation and dimensionality reduction. In addition to a detailed comparative study of network configurations, we contribute by describing a hybrid multi-stage training network that exploits both contrastive and triplet networks to exceed state of the art performance on several SBIR benchmarks by a significant margin.
Datasets and models are available at `www.cvssp.org`.

1. Introduction

Sketches are an intuitive modality for communicating everyday concepts, and are finding increased application on modern touch-screen interfaces (e. g. on tablets, phones) where gestural interaction is natural. Such devices are now the platform on which the majority of today's visual content is consumed, motivating research into sketch as a medium for searching images and video.

This paper addresses the problem of sketch based image retrieval (SBIR); searching a collection of photographs (images) for a particular visual concept using a free-hand sketched query. We explore SBIR from the perspective of a cross-domain modelling problem, in which a low dimensional embedding is learned between the space of sketches and photographs. Traditionally, SBIR has been addressed using sparse feature extraction and dictionary learning, following the successful application of the same to recognition and search in natural images [1, 2, 3]. Deep convolutional neural networks (CNNs) have since gained traction as a powerful and flexible tool for machine perception problems [4], and recently have been explored for SBIR particularly within fine-grain retrieval tasks, e.g. to find a specific shoe within a dataset of shoes [5, 6]. Despite early, promising results, it is unclear how suitable embeddings learned by these multi-branch networks are for generalizing across object categories [3, 2]. For example, enabling a user to search for visual attributes within datasets containing diverse objects (e. g.

*Corresponding author
e-mail: `t.bui@surrey.ac.uk` (Tu Bui)

a specific furniture form, a spotted dog, or particular building structure); a problem explored more extensively by prior work [2, 3, 7].

The technical contributions of this paper are two-fold. First, we present a comprehensive investigation of triplet embedding strategies evaluating these against popular SBIR benchmarks (Flickr15k [3], TU-Berlin [2]). In the spirit of recent 'details' papers studying deep networks for object recognition [8], we explore appropriate CNN architectures, weight sharing schemes and training methodologies to learn a low-dimensional embedding for the representation of both sketches and photographs — in practical terms, a space amenable to fast approximate nearest neighbor (ANN) search (e. g. L^2 norm) for SBIR. Second, we describe a novel triplet architecture and training methodology capable of generalizing across hundreds of object categories, and show this to outperform existing SBIR methods by a significant margin on leading benchmarks [3, 2].

Concretely, we explore several important questions around effective learning of deep representations for SBIR:

1. **Generalization:** Given the diversity of visual concepts in the wild ($\sim 10^5$ categories) and the challenges of annotating large sketch datasets (current best $\sim 10^2$ categories [2]) how well can a network generalize beyond its training to unseen sketched object categories? Are class diversity and volume of exemplars equally important?

2. **Input Modality:** SBIR and the related task of sketched image classification variously employ edge extraction as a preprocessing step to align the statistics of sketch and photo distributions. Is this a beneficial strategy when learning a SBIR feature embedding?

3. **Architecture:** Recent exploration of SBIR has indicated triplet loss CNNs as a promising archetype for SBIR embedding, however what kind of loss objective should be considered and where, and which weight sharing strategies are most effective? What is the best way to enforce a low dimensional embedding for efficient SBIR indexing?

2. Related Work and Contributions

Sketch based Image Retrieval (SBIR) began to gain momentum in the early nineties with color-blob based query systems such as Flickner *et al.*'s QBIC [9] that matched coarse attributes of color, shape and texture using region adjacency graphs. Several global image descriptors for matching blob based queries were subsequently proposed, using spectral signatures derived from Haar Wavelets [10] and the Short-Time Fourier Transform [11]. This early wave of SBIR systems was complemented in the late nineties by algorithms accepting line-art sketches, more closely resembling the free-hand sketches casually generated by lay users in the act of sketching a throw-away query [12]. Such systems are characterised by their optimization based matching approach; fitting the sketch under a deformable model to measure the support for sketched structure within each photograph in the database [13, 14]. Despite good accuracy, such approaches are slow and scale at best linearly. It was not until the 2010 decade that global image descriptors were derived from line-art sketches, enabling more scalable indexing solutions.

2.1. SBIR with shallow features

Mirroring the success of gradient domain features and dictionary learning methods in photo retrieval, both Eitz *et al.* [15] and Hu *et al.* [1] extended Bag of Visual Words (BoVW) to SBIR, also proposing the Flickr15k benchmark [3]. Sparse features including the Structure Tensor [16], SHoG [15], Gradient Field Histogram of Oriented Gradients (GF-HOG) [3] and its extended version [17] are extracted from images pre-processed via Canny edge detection. Chamfer Matching was employed in Mindfinder [18], later adopted by Sun *et al.* [19] for scalable SBIR indexing billions of images. Qi *et al.* [20] implemented an alternative edge detection pre-process delivering a performance gain in cluttered scenes. Mid-level features were explored through the HELO and key-shapes schemes of Saavedra and Barrios [21, 7, 22]. Their latest work [7] uses learned key-shapes and leads the shallow learning approaches.

2.2. SBIR with deep networks

SketchANet [23] was among the earliest deep networks for sketch, exploring recognition (rather than search) using a single-branch network resembling a short-form AlexNet [4]. SketchANet forms a component of the very recent work of Bhattacharjee *et al.* [24], coupled with a complex pipeline including object proposals, and query expansion. Although we also explored SketchANet, and compare with several other contemporary architectures which we show yield superior performance in a triplet framework (Sec. 4).

An early work exploring multi-branch networks for sketch retrieval (of 3D objects) was the contrastive loss network of Wang *et al.* [25] which independently learned branch weights to bridge the domains of sketch and 2D renderings of silhouette edges. In a recent short paper, Qi *et al.* [26] also propose a two-branch Siamese network with contrastive loss. Their results, although comparable with other methods using shallow features, are still far behind state-of-the-art [24, 6] by a large margin. As we show later, learning a single function to map disparate domains to the search space appears to under-perform designs where branch weights are learned independently or semi-independently.

Triplet CNNs employ three branches [27]: (i) an anchor branch, which models the reference object, (ii) one branch representing positive examples (which should be similar to the anchor) and (iii) another modeling negative examples (which should differ from the anchor). The triplet loss function is responsible for guiding the training stage considering the relationship between the three models. Triplet CNNs have recently been explored for face identification [28], tracking [29], photographic visual search in[27, 30] and for sketched queries in order to refine search within a single object class (e. g. fine-grain search within a dataset of shoes) [5]. Similarly, a fine-grained approach to SBIR was adopted by the recent Sketchy system of Sangkloy *et al.* [6] in which careful reproduction of stroke detail is invited for object instance search. In the former work [5], the authors train one model for each target category, and the embedding is learned using an edgemap extracted from a relatively clutter-free image. They report that using a fully-shared network was better than use two branches without weight sharing. However, the authors in [6] suggest it is more beneficial to avoid sharing any layers in a cross-category retrieval context. Recently, a hybrid design was explored by Bui *et al.* [31] using the same architecture on both branches but sharing certain layers. However, as their model learns mapping between sketch and edgemap (rather than image directly) its performance is limited. Furthermore, it is still unclear whether triplet loss works better than contrastive loss, with [6, 31] supporting the former but [32] claiming the latter. Open questions remain around optimal training methodology, architecture, weight-sharing strategies, and loss functions, as well as the generalization capability of deep models for SBIR.

Our work explores these open questions, and broadens the investigation of deep learning to SBIR beyond intra-class or instance level search to retrieval across multiple object categories. To avoid confusion we hereafter refer as *no-share* or Heterogeneous those multi-branch networks for which there are no shared weights between layers [25]; as *full-share* or Siamese those for which all branches have shared weights in all layers [5, 27]; and *partial-share* or Hybrid those for which only a subset of layers are shared.

Our contributions for this paper are three-fold:

- A generic multi-stage training methodology for cross-domain learning that leverages multiple loss functions in training shared networks as illustrated in Figure 1.
- An extensive evaluation of convnet architectures and weight sharing strategies.
- State-of-the-art performance on three standard SBIR benchmarks, outperforming other approaches by a significant margin.

3. Methodology

We propose a multi-stage training methodology and investigate several network designs, comparing the Siamese architecture with the Heterogeneous and Hybrid ones. Inspired from [31], we aimed to develop a training strategy for partial sharing networks. However, unlike [31] who employed a single training phase with a single loss function to concurrently

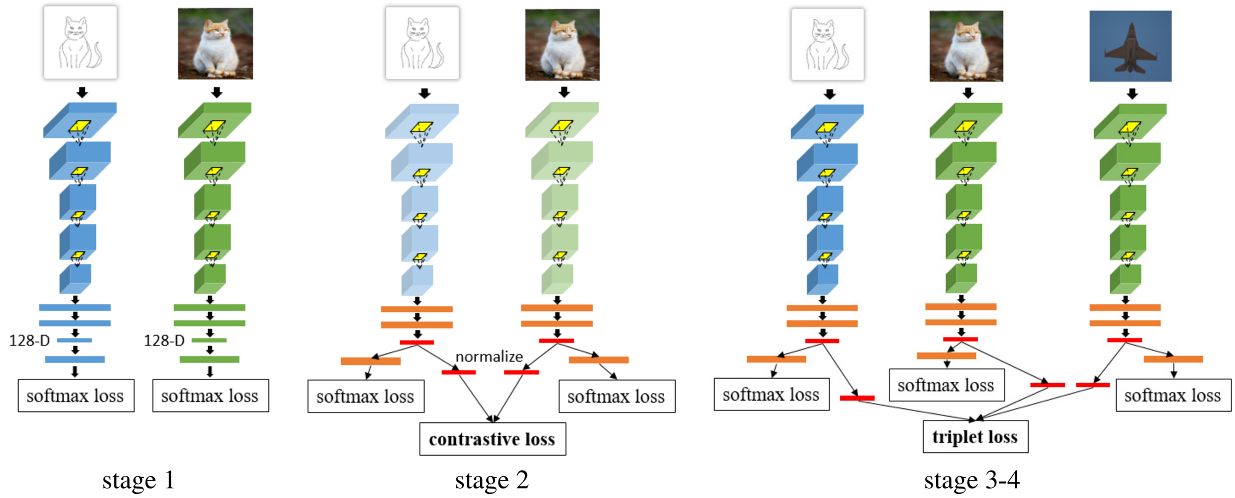


Fig. 1. Our training procedure illustrated with a SketchANet-AlexNet architecture: pre-training the unshared layers (stage 1), and the shared layers (stage 2) separately before plugging those into a triplet network (stage 3 and 4).

train both shared and unshared parts of their sketch-edgemap network, we believe training a sketch-photo network should require more complex procedures. Additionally, we integrate the two most widely used regression functions in deep convnet, the contrastive loss and triplet loss, in our training procedure.

3.1. Network architecture

When learning a cross domain mapping between sketch and photo using deep convnet, at least two CNN branches are required to deliver feature embedding for these domains. The sketch branch and image branch may have the same or different architecture. Let $\mathbf{X}^S = \{\mathbf{x}^S\}$ and $\mathbf{X}^I = \{\mathbf{x}^I\}$ be collections of training sketches and images. Supposed $F^S_{\theta_S,\theta_C}(\mathbf{x}^S)$ and $F^I_{\theta_I,\theta_C}(\mathbf{x}^I)$ are the embedding functions for sketch and image domains respectively. Parameters $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_I$ represent domain-specific layers; while $\boldsymbol{\theta}_C$ are the common/shared parts. In the scope of this paper, we investigated SketchANet [23], AlexNet [4], VGG 16 layers (VGG16) [33] and InceptionV1 (GoogLeNet) [34] for the sketch branch $\{\boldsymbol{\theta}_S, \boldsymbol{\theta}_C\}$; and AlexNet, VGG16 and InceptionV1 for the image branch $\{\boldsymbol{\theta}_I, \boldsymbol{\theta}_C\}$, although other architectures can also be employed using the same methodology.

Differences in design can also arise from the degree to which layers within the two branches share weights. Most of the existing approaches eliminate either $\{\boldsymbol{\theta}_S, \boldsymbol{\theta}_I\}$ (i. e. full-share) as in [35, 26, 5], or $\boldsymbol{\theta}_C$ (i. e. no share) in [6, 25]. It was shown in [36, 37] that low-level features are often learned in bottom layers of a CNN network while higher semantic features tend to emerge from top layers. Therefore, intuitively we want to share the top layers so that the feature embedding is learned across domains considering the semantics (e.g. categories/classes), and let the bottom layers be learned separately for each domain. If the sketch and image branch architectures are completely different, we possibly need one or several fully-connected (FC) layers unifying the branches, as well as loss functions pre- and post- unification. We explore several design permutations, evaluating their performance in Sec. 4 with the aim of testing the generalization capability of the network across categories, and identifying the best performing architecture (CNN architecture, loss) and training strategy to optimize retrieval accuracy.

At certain training stages, a contrastive loss or triplet loss can be employed. We normalize inputs prior to these losses. The contrastive loss function accepts a pair of input examples $(\mathbf{x}^S, \mathbf{x}^I)$ and regress their embedding closer or push them away depending on whether or not $\mathbf{x}^S$ and $\mathbf{x}^I$ are similar [38]. Let Y represents the label of a training pair $(\mathbf{x}^S, \mathbf{x}^I)$ such that:

$$Y = \begin{cases} 0 & \text{if } (\mathbf{x}^S, \mathbf{x}^I) \text{ are similar} \\ 1 & \text{if } (\mathbf{x}^S, \mathbf{x}^I) \text{ are dissimilar} \end{cases} \tag{1}$$

The cross-domain Euclidean distance between two branch's outputs is defined as follows:

$$D(\mathbf{x}^S, \mathbf{x}^I) = \left\| F^S_{\theta_S,\theta_C}(\mathbf{x}^S) - F^I_{\theta_I,\theta_C}(\mathbf{x}^I) \right\|_2 \tag{2}$$

Then the **contrastive loss** can be written as:

$$\begin{aligned}\mathcal{L}_C(Y, \mathbf{x}^S, \mathbf{x}^I) = \frac{1}{2}(1 - Y)D^2(\mathbf{x}^S, \mathbf{x}^I) + \\ \frac{1}{2}Y\{m - D^2(\mathbf{x}^S, \mathbf{x}^I)\}_+\end{aligned} \tag{3}$$

where $\{.\}_+$ is hinger loss function. Parameter m is a margin defining an acceptable threshold for $\mathbf{x}^S$ and $\mathbf{x}^I$ to be considered as dissimilar.

The **triplet loss**, on the other hand, maintains a relative distance between the anchor example and both a similar example and a dissimilar example. The function accepts an input triplet of form $(\mathbf{x}^S, \mathbf{x}^I_+, \mathbf{x}^I_-)$ consisting an *anchor* sketch example $\mathbf{x}^S$, a similar image $\mathbf{x}^I_+$, and a dissimilar one $\mathbf{x}^I_-$. The triplet is then given by:

$$\mathcal{L}_T(\mathbf{x}^S, \mathbf{x}^I_+, \mathbf{x}^I_-) = \frac{1}{2}\{m + D^2(\mathbf{x}^S, \mathbf{x}^I_+) - D^2(\mathbf{x}^S, \mathbf{x}^I_-)\}_+ \tag{4}$$

To accommodate the input triplet $(\mathbf{x}^S, \mathbf{x}^I_+, \mathbf{x}^I_-)$, the CNN network consists of three branches: a sketch branch (anchor) and two identical image branches (positive and negative). The value of margin m is fixed at 0.2 in all of our experiments.

3.2. Dimensionality reduction

A compact representation is often desirable to allow viable implementation of visual search in systems with processing, battery and memory constraints. In order to learn the dimensionality reduction during the training stage we add an intermediate fully-connected (FC) layer without post-activation. As illustrated in Fig. 1 for the SketchANet-AlexNet, an embedding layer *lowerdim* is added between layer FC7 ($D = 4096$) and the output layer FC8 ($D = 250$). By not adding an activation (ReLU) layer, we prevent the embedding layer to become a bottleneck in the network. Note that from the perspective of the softmax-loss layer the connection from FC7 to FC8 is linear. We empirically verify that during training the performance of the classification layer is not affected whether *lowerdim* is integrated in the architecture or not. Dimensionality reduction is tested in subsec. 4.5. Further gains in compactness could be explored e.g. via product quantization as [31] but such optimizations are beyond the scope of this paper.

3.3. Training procedure

We now describe a multi-stage training strategy for all network configurations. Although this strategy is designed for sketch-photo mapping, it can be applied to other cross-domain learning problems. Inspired from curriculum learning [39], we trained our model by giving it multiple learning tasks, one-by-one with increasing difficulties. Denote $\mathcal{L}_E$ and $\mathcal{L}_R$ the cross-entropy and regularization losses:

$$\mathcal{L}_E(\mathbf{z}) = -\,log(\frac{e^{z_y}}{\sum_i e^{z_i}}) \tag{5}$$

$$\mathcal{L}_R(\boldsymbol{\theta}) = \frac{1}{2}\sum_i \theta_i^2 \tag{6}$$

Our training procedure consists of 4 stages (Fig. 1):

– **Stage 1: train unshared layers** Train the sketch and photo branches independently using a softmax loss, using pre-trained model if possible. This is purely a classification task which focuses on learning a representative model for each domain:

$$\arg\min_{\theta_S, \theta_C} \sum_i \mathcal{L}_E(F^S(\mathbf{x}^S_i)) + \lambda \mathcal{L}_R(\boldsymbol{\theta}_S, \boldsymbol{\theta}_C) \tag{7}$$

$$\arg\min_{\theta_I, \theta_C} \sum_i \mathcal{L}_E(F^I(\mathbf{x_i}^I)) + \lambda \mathcal{L}_R(\boldsymbol{\theta}_I, \boldsymbol{\theta}_C) \tag{8}$$

where λ is the weight decay term. Note: in eqn. 7 and 8 $\boldsymbol{\theta}_C$ was learned independently since no joint training is implemented at this stage.

– **Stage 2: train shared layers** We form a double-branch network, freeze the unshared layers which were already learned during stage 1. Next, we use contrastive loss together with softmax loss to train the shared layers. The use of softmax loss helps the sharing layers to learn discriminative features from both domains, whilst contrastive loss (eqn. 3) provides an early step of regression to bring the two domains together:

$$\begin{aligned}\arg\min_{\theta_C} \sum_i \mathcal{L}_E(F^S(\mathbf{x}^S_i)) + \sum_i \mathcal{L}_E(F^I(\mathbf{x}^I_i)) + \\ \alpha \sum_i \mathcal{L}_C(Y_i, \mathbf{x}^S_i, \mathbf{x}^I_i) + \lambda \mathcal{L}_R(\boldsymbol{\theta}_C)\end{aligned} \tag{9}$$

where α is weight of the regression term. We set $\alpha = 2.0$ in all experiments.

– **Stage 3: train the whole network** Unfreeze all frozen layers, form a triplet network and train it with triplet (eqn. 4) and softmax loss functions. We begin the training with two losses contributing equally, then later increase loss weight of the triplet function ($\alpha = 2.0$) to steer the learning towards regression:

$$\underset{\theta_S,\theta_I,\theta_C}{\arg\min} \sum_i \mathcal{L}_E(F^S(\mathbf{x}_i^S)) + \sum_i \mathcal{L}_E(F^I(\mathbf{x}_{i+}^I)) + \sum_i \mathcal{L}_E(F^I(\mathbf{x}_{i-}^I)) + \alpha \sum_i \mathcal{L}_T(\mathbf{x}_i^S, \mathbf{x}_{i+}^I, \mathbf{x}_{i-}^I) + \lambda \mathcal{L}_{\mathcal{R}}(\boldsymbol{\theta}_S, \boldsymbol{\theta}_I, \boldsymbol{\theta}_C) \tag{10}$$

– **Stage 4: (Optional)** Repeat stage 3 on any auxiliary sketch-photo datasets available to further refine the model.

Our proposed training procedure allows the shared and unshared layers to be learned independently at separate stages. The unshared layers of each branch should learn unique features distinctive for its domain without being polluted from other domain (stage 1). The shared layers should learn common features (usually high level) between the two domains by comparing and contrasting low level features from both domains (stage 2). Finally, the whole network is adjusted/refined using triplet loss (stage 3-4).

Although contrastive and triplet losses are crucial in regression learning, we find them not tight enough to regulate the training. That is why a softmax loss layer is always included in our network at every training stage since it provides a stricter regularization. Our findings are consistent with the work in [6, 35] claiming the softmax loss plays an important part for convergence of the training. On the other hands, our approach differs from [6, 35] in that it allows partial sharing across branches; which further reduces overfitting (since number of training parameters are significantly reduced) while retaining the learning flexibility for each domain.

3.4. Data augmentation

Data augmentation plays an important role in preventing overfitting, especially when training data is limited. In all experiments we apply the following augmentation techniques for both sketch and photo: random crop (crop size 225x225 for SketchANet, 227x227 for Alexnet and 224x224 for VGG and Inception), random rotation in range $[-5, 5]$ degrees, random scaling in range $[0.9, 1.1]$ and random horizontal flip.

We also propose an augmentation method applicable for sketches only. For sketches with at least N strokes ($N = 10$ in our experiments) we divide them into four equal groups of strokes in drawing order. The first group contains the most important strokes — related to the more coarse structure of the object — and it is always kept. A new sketch is created by randomly discarding some of the other groups. This technique is inspired by Yu *et al.* [23, 5] who observe that people tend to draw sketches in stages at distinct levels of abstraction. We observed a $\sim$1% mAP improvement across the board using this random stroke removal augmentation method on the Flickr15k benchmark.

4. Experiments

We evaluated our training strategies on all variants of the sketch and image architectures and weight sharing schemes to determine the best performing embedding for SBIR. In particular we evaluated the ability of the network to generalize beyond the categories to which it is exposed during training. This is important for SBIR in the wild, where one cannot reasonably train with a sufficiently diverse sample of potential query images. We also investigated the impact of volume of sketch data used to train the network, and the impact of using photos or their edge-maps during training (in addition to the various weight sharing variants).

The structure of this section is as follows. We introduces train and test datasets in subsec. 4.1, experimental settings in subsec. 4.2. We evaluate generalization properties in subsec. 4.3, network architectures and sharing in subsec. 4.4, and dimensional reduction in subsec. 4.5. Finally, subsec. 4.6 compares our proposed approach with state-of-art algorithms.

4.1. Datasets

We trained and evaluated our networks using five sketch datasets:

– **TU-Berlin-Class** [2] (training stage 1-3) for sketch classification comprising 250 categories of sketches, 80 per category, crowd-sourced from 1350 different non-expert participants with diverse drawing styles;

– **TU-Berlin-Retr** [15] (testing) takes into account not only the category of the retrieved images but also the relative order of

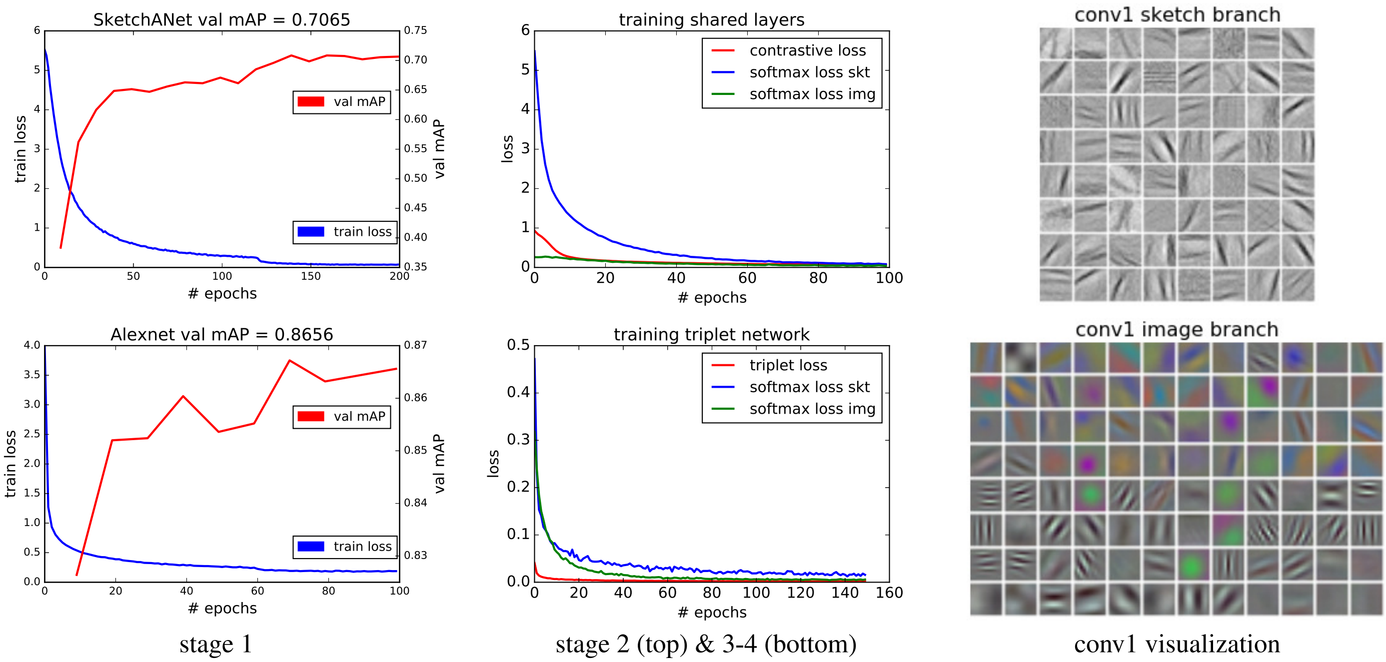


Fig. 2. 4-stage training of the SketchANet-AlexNet model and visualization of the first convolution layer on sketch and image branch.

the relevant images. The dataset consists of 31 sketches and 40 ranked images for each sketch (1240 total images), mixed with a set of 100,000 distracting Flickr photos. The authors propose a Kendal score as the evaluation method;

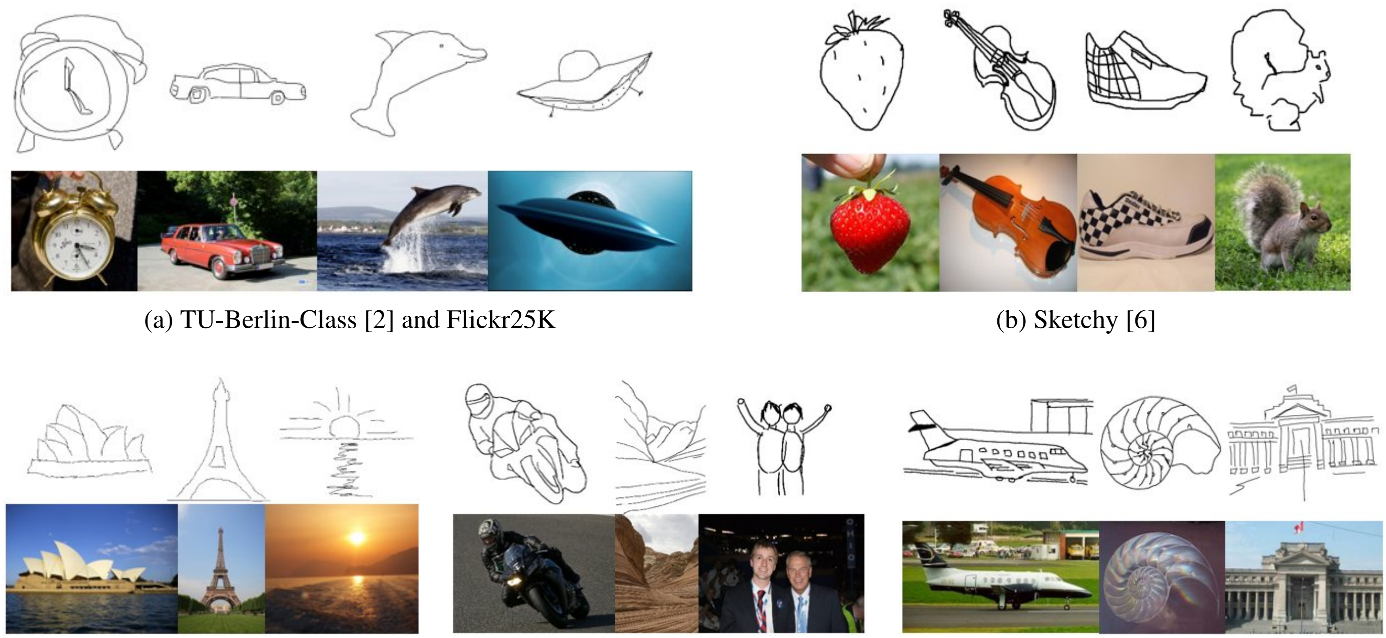
– **Sketchy** [6] (model fine-tuning at stage 4) is a fine-grained dataset in which each photo image has ~5 instance-level matching sketches drawn by different subjects. In total it has 12,500 photo images and 75,471 corresponding sketches of 125 categories of which 100s exist in the TU-Berlin-Class and 25s are the new categories;

– **Flickr15K** [3] (testing) is a large scale category-level dataset. It has labels for 33 categories of sketches, 10 sketches per category drawn by 10 non-expert sketchers. It also has a different number of photo images per category totalling 15,024 images crawled from FlickR. The authors suggest to use Mean Average Precision (mAP) as the performance metric;

– **Saavedra-SBIR** [40] (testing) another category-level dataset, consisting 53 sketches and 1326 images organized into 50 classes. Similar to Flickr15K, the authors recommended mAP for evaluation.

It is important to note that the Flickr15K and TU-Berlin-Retr datasets are independent from the training ones in term of not only categories but also depiction styles. The TU-Berlin-Class and Sketchy covers common objects frequently encountered in daily life (stationary, vehicles, food, bird, mammal,...). The Flickr15K contains mostly landmarks and buildings (e.g. Eiffel tower, Colosseum, Taj Mahal,...) while the TU-Berlin-Retr tends to be scenery specific (Fig. 3 (a-d)). On the other hand, Saavedra-SBIR happens to share 30 common categories with TU-Berlin-Class, but its query set contains distinct sketches with exceptionally high level of details (Fig. 3 (e)). These settings motivate a need for good generalization beyond training. Additionally, it helps to avoid bias when comparing with non-learning methods which do not require any training data.

As TU-Berlin-Class comprises only sketches, in order to obtain our training triplets we automatically generated per-category photograph sets by querying the 250 category names on Creative Commons image repositories. The Flickr API was used to download images from 184 categories. Google and Bing engines were used for the remaining 66 categories which are mainly human body parts (e.g. brain, tooth, skeleton) and fictional objects (e.g. UFO, mermaid, dragon) where Flickr content is sparse. We manually selected the 100 most relevant photos per category, forming a 25k training corpus (Flickr25K).



(a) TU-Berlin-Class [2] and Flickr25K

(b) Sketchy [6]

(c) Flickr15K [3]

(d) TU-Berlin-Retr [15]

(e) Saavedra-SBIR [40]

Fig. 3. Example sketches and images of the training (a-b) and test (c-e) datasets.

4.2. Experimental settings

We followed the four training stages outlined in subsec. 3.3. Photo images are first resized retaining aspect ratio so that maximum dimension is 256 pixels, then padded with duplicate pixels along the edges to form unified 256x256 input data. Sketches are also centred in 256x256 canvas such that the longest side of its bounding box is fixed at 200 pixels. Since the training procedure involves multiple sketch datasets whose stroke thickness may vary, all sketches are skeletonized to have 1-pixel stroke width using the morphological thinning method described in [41].

Data augmentation is implemented as in subsec. 3.3. One exception is the implementation of random flip in stage 4 where the finegrained Sketchy dataset is being used. To keep the finegrain properties, random flip is performed jointly over the anchor-positive pair. We do not do the same with random rotation and scaling since the rotation range [-5,5] and resizing scale [0.9, 1.1] are relatively small and can account for the alignment error between the images and corresponding sketches.

We used Caffe the deep-learning library [42] to train our models. When training the contrastive and triplet networks (stage 2 onward), the anchor-positive and negative pairs are selected randomly. However, depending on the dataset, a pair/triplet can be of either categorical-level (where the positive image has the same category label as the anchor's and the negative image is from a different category) or instance-level (the positive image has the same instance label i.e. same object, while the negative image has the same category label but different instance's). We used categorical-level pair for stage 2 and categorical-level triplet for stage 3 since the TU-Berlin-Class dataset only supports category matching. For the Sketchy dataset (stage 4), we combined both categorical and instance-levels in triplet formation. Specifically, for a given training sketch there is 20% chance a categorical triplet is formed and 80% chance for an instance-level triplet. This helps to learn a model that is both intra- and inter-categorical representative. Our idea is similar to the Quadruplet network [35] but instead of introducing a new quaduplet input format and a new loss function we achieve it via data selection. We do not implemented hard negative mining since the instance-level selection of triplets in stage 4 is already hard enough for the training to properly converge. An example of training a SketchANet-AlexNet model is illustrated in Fig. 2.

4.3. Generalization

We first report the results of generalization capability of our triplet networks when varying amount of training data. A series of experiments was carried out, starting with a subset of 20 random training categories and 20 sketches per category, up to the whole training dataset. As the TU-Berlin-Class has 80 sketches per category, the remaining sketches of the chosen categories were used for validation. For simplicity we used SketchANet for the sketch branch and AlexNet for image branch. We modified the SketchANet design to enable sharing with AlexNet. Specifically, layers 1-3 of the sketch branch have SketchANet architecture, layers 6-7 mirror AlexNet while the middle layers 4-5 we have modified from SketchANet as a hybridization of the two designs. The modified sketch branch is trained from scratch while the image branch is initialized using the ImageNet pre-trained model [4]. Apart from testing generalization we aimed to compare and contrast this partial sharing design with the fully shared and no-share architectures; also to verify whether our sketch-photo direct matching is better than the sketch-edgemap reported in [31].

Fig. 4 (top) shows that the performance is benefited by increasing the number of training categories. All five network designs achieved near-linear improvement of retrieval performance against Flickr15k benchmark (discarding the four intersecting categories with the training set) with exposure to more diverse category set during training. The mAP of all models jumped by ~20% when raising training data from 20 to 250 categories. Fig. 4 (middle) has similar trend when we keep number of training categories fixed at 250s and vary number of training sketches per category. As the results of seeing more data during training, all models achieve an improvement of up to 4% mAP on Flickr15k. Fig. 4 (bottom) depicts that number of training samples is not the only factor that matters most. Here we increase the number of categories from 20s to 80s while at the same time decreasing per category samples, keeping the training volume fixed at 4800 sketches. The general trend is an improvement as number of categories increase. We conclude that category diversity is crucial for training a generalized network.

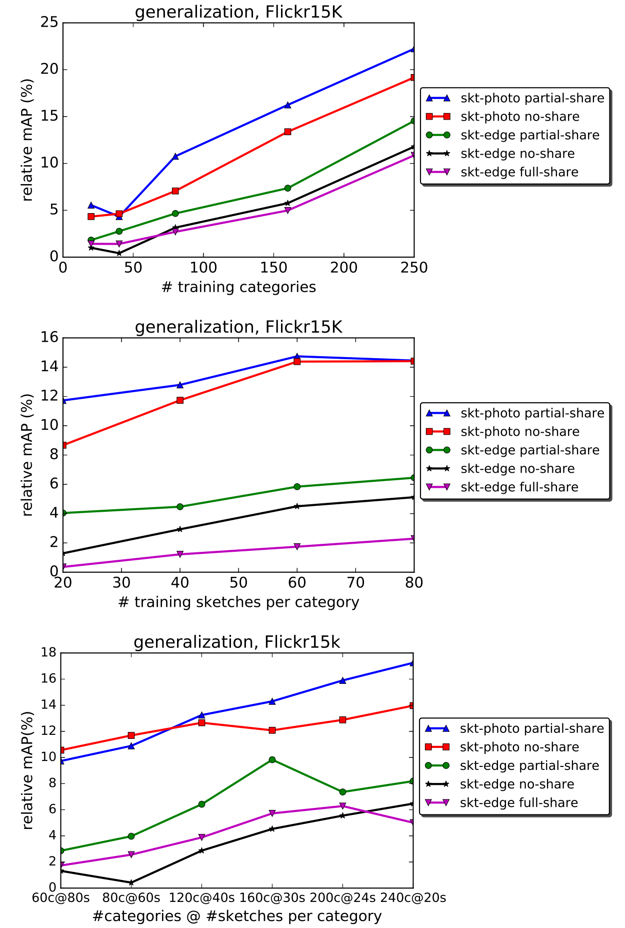


Fig. 4. Experiments with generalisation capability of our learned models w.r.t. (top) number of training categories (20 sketches per category); (middle) number of training sketches per category (250 categories); (bottom) fixed training volume (fixed 4800 training samples); tested on the Flickr15K benchmark.

All three above figures report the superior performance of the partially shared triplet architecture against the no-share and fully shared networks regardless of its matching formats (sketch-edgemap or sketch-photo). Also, the sketch-photo models outperforms the sketch-edgemap ones by a large margin. This is understandable since working directly on photo images enable the network access full information from raw data. In contrast, during edge extraction, certain information such as colour and texture that may be distinctive to identify the objects of interest will be lost, leaving the network with less informative data to learn from.

For completeness, Fig. 5 compares our multi-stage training method (subsec. 3.3) with Siamese and Triplet models using one-shot training. The network design is the same i. e.

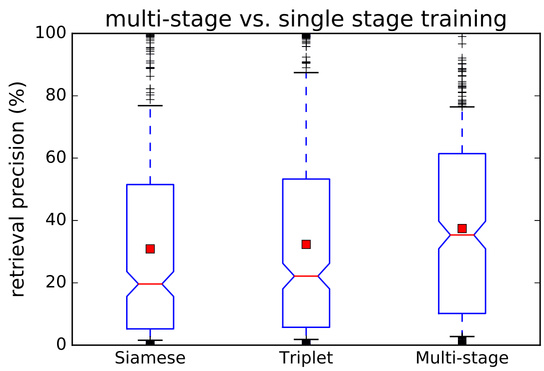


Fig. 5. Multi-stage training compared with single-stage models, tested on Flickr15K.

SketchANet-AlexNet for three models but the Siamese and Triplet models are trained within a single training stage (with weights also initialized from pretrained models). We observed a 5% improvement in mAP with our multi-stage model. Note all three box-plots have large interquartile range (IQR) and whiskers, which illustrates a great performance diversity among sketch queries e. g. clean sketches can achieve 100% retrieval precision while messy sketches may end up ~0% performance.

4.4. Convnet architecture settings and parameter sharing

We experimented various architectures among SketchANet, AlexNet, VGG16 and InceptionV1 for sketch and image branches. For each sketch-image architecture combination, we test all possible sharing options and report the best performed model. For example, the fully connected layer 7 (FC7) and later in AlexNet and VGG are share-able while SketchANet and InceptionV1 can only share parameters after the dimensional reduction layer (*lowerdim* in Fig. 1).

Table 1 shows the performance of all available combinations of sketch-image designs on the Flickr15k benchmark. Again, we found that for certain sketch-photo architecture combinations there always exists a partial sharing configuration better than the full-share and no-share ones. For example, AlexNet-VGG16 has the highest performance (39.77% mAP) when sharing from layer FC7, SketchANet-AlexNet performs the best at sharing from FC6. InceptionV1 has a distinct architecture however we found that sharing all layers following *lowerdim* (i.e. the n-way classifier FC layer) results in a better mAP.

It is worth noting that the sketch branch should not be more complex than the image branch. The AlexNet-VGG16, AlexNet-InceptionV1 and VGG16-InceptionV1 designs all outperform their VGG16-AlexNet, InceptionV1-AlexNet and InceptionV1-VGG16 counterparts by 2-7% mAP. Additionally, when InceptionV1 is selected for the image branch, choosing SketchANet for the sketch branch is more efficient than AlexNet or VGG16 although SketchANet is simpler and has fewer parameters than the two others. We hypotheses that having an over-complicated design for the sketch branch can cause it over-trained in a contrastive or triplet network, especially with limited training data.

Nevertheless, using identical architecture for both sketch and image branches results in the highest performance (the diagonal line of Table 1). We conjecture that partially shared sketch and image branches may enable more balanced weight updates during back-propagation, mitigating against over-training in a single branch. This may prove a useful strategy more generally in combating over-fitting alongside popular methods such as regularization and dropout.

Details of the weight sharing experiments for identical branch (i.e. homogeneous) triplet networks are shown in Fig. 6. The best sharing configurations for AlexNet-AlexNet and VGG16-VGG16 are from conv5 and block5 respectively. For InceptionV1-InceptionV1, there is a drop in performance at Inception block 4d where the second auxiliary classifier (*tendrill*) is attached. Removing the auxiliary classifiers (the main classifiers at top of the network remain shared), we achieve peak performance when sharing from inception layer 4e. In all three cases the no-share configuration under-performs both the full-share and partial-sharing performace (the performance gain ranges from 7% for VGG16 to 14% for AlexNet).

4.5. Dimensionality reduction

Fig. 7 reports the mAP and retrieval time of our best model in Table 1 (InceptionV1-InceptionV1) when varying output dimension within range $D \in [64, 1024]$. In general the mAP steadily improves as size of *lowerdim* increases. We achieve a record performance of 55.06% mAP on Flickr15K at D=1024.

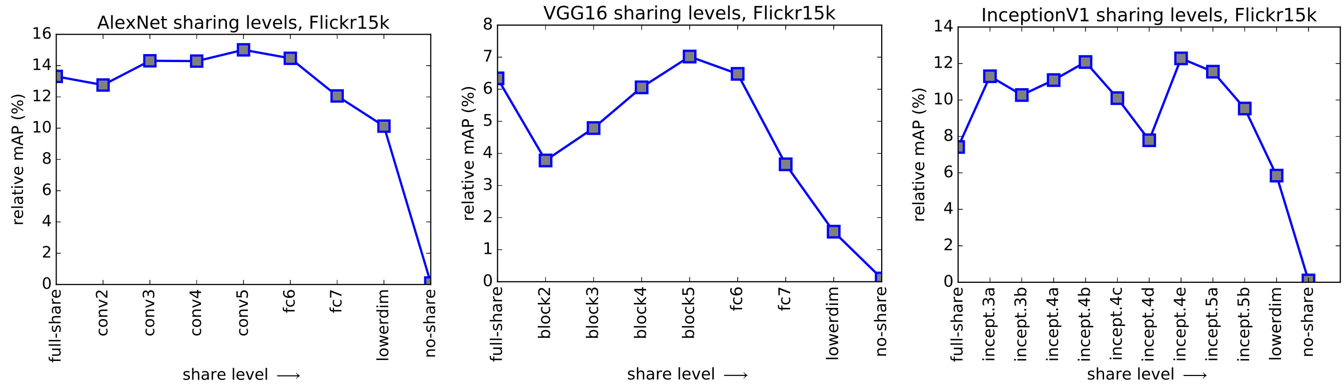


Fig. 6. From full-share to no-share: effects of partial sharing on accuracy (a) AlexNet-AlexNet; (b) VGG16-VGG16; and (c) InceptionV1-InceptionV1 networks, evaluated over FlickR15K.

Flickr15K SBIR mAP(%)		Image branch			
		SketchANet [31]	AlexNet	VGG16	InceptionV1
Sketch branch	SketchANet	24.45	37.41	36.80	41.99
	AlexNet	-	**45.16**	39.77	41.65
	VGG16	-	36.22	**49.99**	40.74
	InceptionV1	-	34.98	38.77	**51.11**

Table 1. Performance of various network designs on the FlickR15K benchmark. Note: (i) SketchANet-SketchANet is the only sketch-edgemap model (reported in [31]), the rest are sketch-photo models; (ii) lowerdim is fixed at 128-D for all models.

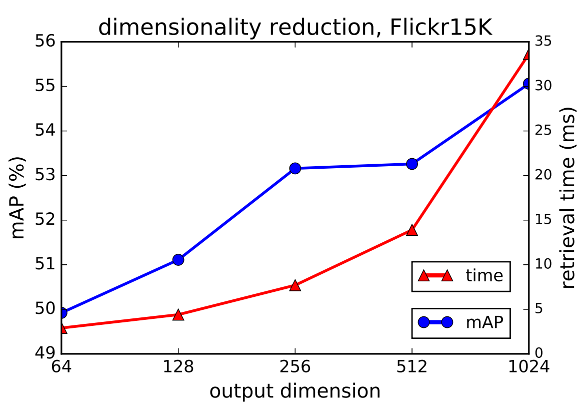


Fig. 7. Accuracy and speed performance of InceptionV1-InceptionV1 model with different output dimension.

However, retrieval time also linearly increases (note the x-axis of Fig. 7 is log scale). On a commodity 2.80GHz Intel i7 workstation, a simple linear search using a single CPU thread takes from 3ms to 34ms per query when increasing *lowerdim*'s dimension from 64-D to 1024-D.

Considering the trade off between speed and accuracy we selected D=256 as our final model (53.26% mAP, 4.4ms retrieval time). This allows us to encode the whole Flickr15K dataset using just 15MB of memory, or 1MB footprint for every 1K images. Since the linear search complexity is $O(ND)$ and feature extraction time is averagely 15.2ms per query (on a GeForce GTX 1070 GPU), in theory our model can retain interactive speed (i. e. retrieval time less than 1 second) when querying up to 3M images. For larger datasets, more efficient indexing methods e. g. kd-tree, inverted index,... are recommended.

4.6. Benchmark evaluation

We compare our selected model (InceptionV1-InceptionV1 with partial sharing from inception block 4e, output dimension 256-D) with other approaches in the literature. The first benchmark is the defacto Flickr15k [3] datasets used in ~20 published SBIR algorithms and variants. Some key approaches are:

- Hand-crafted approaches: these methods use hand-crafted features and often dictionary learning to deliver global fingerprint for each image. Notable algorithms include Structure Tensor [16], Shape Context [43], Self Similarity (SSIM) [44], SHoG [15], SHELO and its variants [22, 45], HLR and its variants [46], KeyShapes [7], GF-HoG and its color version [17, 3] and Perceptual Edge [20].

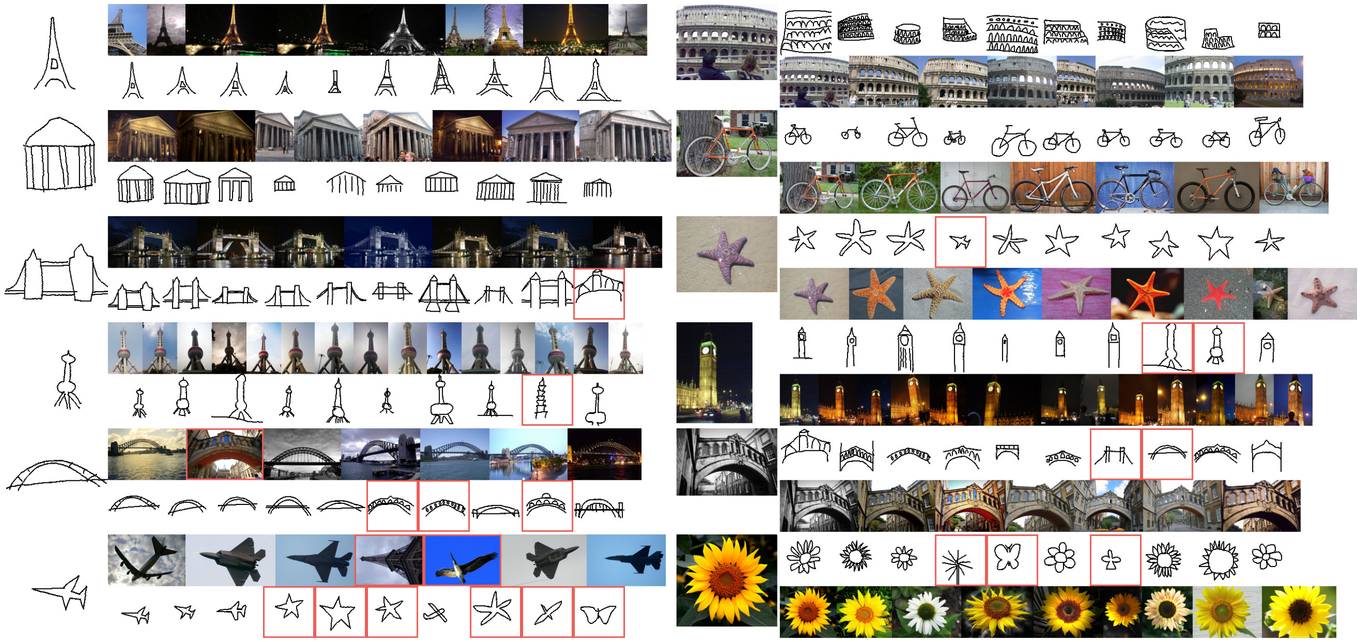


Fig. 8. Representative SBIR results on Flickr15K using (left) sketches and (right) images as queries. For each query, two sets of results are returned, one for intra-domain and the other for cross domain search. Red bounding boxes indicate false positives.

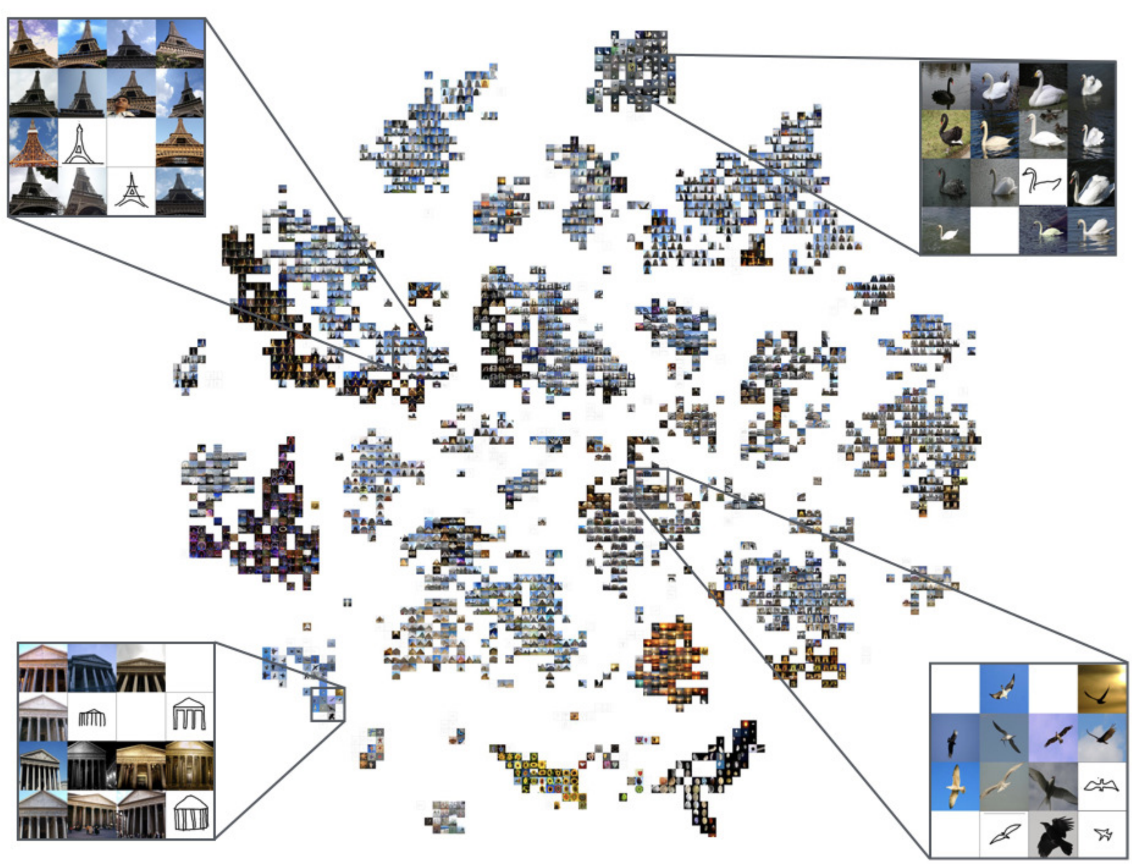


Fig. 9. t-sne visualization of the Flickr15K dataset within our best performing embedding (InceptionV1-InceptionV1). Sketches and photographs of objects are mapped to similar locations in 128-D space.

- CNN-related approaches: use deep features with various architecture settings and loss functions. These include Siamese network [26], Triplet sketch-edgemap network [31], Asymmetric feature map (AFM) [47], Quadruplet_MT [35], Query-adaptive CNN with re-ranking [24].

The results are reported in Table 2. Our partially-shared network outperforms the rest by a significant margin even at earlier training stages. Specifically, our proposed approach leads the closest method by 17% mAP and achieves twice performance as the best hand-craft method (LKS) while having 5 times more compact descriptor. This further demonstrates the needs of a partial sharing network and the advantages of multi-stage training in solving a cross-domain problem. Table 2 explores how much improvement is obtained following each individual stage of the multiple stage training process. Table 1 and Fig.6 indicates a deeper backbone network such as InceptionV1-InceptionV1 and an appropriate partial weight sharing strategy can improve 7-15% mAP. Fig. 4 shows the importance of training data volume, especially for ensuring generalization beyond training categories, and that data augmentation can improve a further 1% mAP.

Fig. 10 depicts the precision-recall (PR) curves of our proposed approaches along with another CNN-related method and one of the state-of-art hand-crafted approaches on Flickr15k. While the PR curve of Color GF-HoG [17] is smooth the deeply learned (CNN) approaches have irregular PR curves. Nevertheless, there is an improvement in the level of smoothness from the curves stage 2 to 4, indicating potential of our model to generalise to data "in the wild", given sufficient category diversity in the training data. Fig. 9 shows the embedding of Flickr15k sketches and images. SBIR examples are given in Fig. 8.

Next, we evaluated over Saavedra-SBIR (using mAP) and TU-Berlin-Retr (using $\mathcal{T}_b$ proposed in [15]). Table 3 and 4 show our final model also achieving state-of-art performance. While the training stages 2-3 is supplied with categorical-level data only, the finetuning stage 4 on Sketchy helps to learn more detailed representation of sketches and images, contributing to an improvement of 4% mAP on Saavedra-SBIR and 1.5 $\mathcal{T}_b$ on TU-Berlin-Retr as opposed to the closest approaches.

Method	Dim.	mAP (%)
Partial sharing convnet (stage 4)	256	**53.26**
Partial sharing convnet (stage 3)	256	**41.13**
Sketchy triplet [6]†	1024	35.91
Partial sharing convnet (stage 2)	256	**34.83**
Query-adaptive re-ranking CNN [24]	5120	32.30
Quadruplet_MT [35]	1024	32.16
Asymmetric feature map (AFM) [47]	243	30.40
Learned KeyShapes (LKS) [7]	1350	24.50
Triplet sketch-edgemap [31]	100	24.45
Rst-SP-SHELO [22]	3060	20.05
Siamese with Contrastive Loss [26]	64	19.54
Perceptual Edge [20]	3780	18.37
Color GF-HoG [17]	5000	18.20
HLR+S+C+R [46]	2000	17.10
SHELO [45]	1296	12.36
GF-HoG [3]	3500	12.22
SHoG [15]	1000	10.93
SSIM [44]	500	9.57
SIFT [48]	1000	9.11
Shape Context [43]	3500	8.14
Structure Tensor [16]	500	7.98

Table 2. SBIR comparison results (mAP) on the Flickr15K benchmark. Methods that do not originally report on Flickr15K are marked with †. Our proposed convnet uses InceptionV1 architecture for both sketch and image branches with partial sharing from inception block 4e.

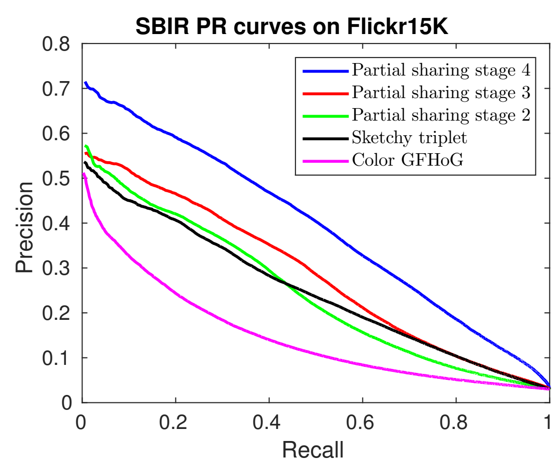


Fig. 10. PR curve of the proposed approaches compared with a state-of-the-art non-learning method [17].

In Fig. 11, we analyze the retrieval performance of the query sketches whose categories are known to the model during training and compare with those are not. The queries with seen categories indeed have better retrieval rate than those belong to unseen categories. However, our final model (stage 4) gains the highest retrieval precision on these challenging queries. Also,

Method	Dim.	$\mathcal{T}_b$
Partial sharing convnet (stage 4)	256	**44.8**
Quadruplet_MT [35]	1024	43.3
Sketchy triplet [6]†	1024	37.5
Partial sharing convnet (stage 3)	256	**35.6**
Partial sharing convnet (stage 2)	256	**31.8**
KeyShapes [21]	-	28.9
SHoG [15]	1000	27.7
Triplet sketch-edgemap [31]	100	22.3
HoG (global) [15]	768	22.3
Structure Tensor [16]	500	22.3
Spark [15]	1000	21.7
HoG (local) [49]	1000	17.5
Shape Context [43]	3500	16.1

Table 3. SBIR comparison results (using Kendal's rank correlation coefficient, $\mathcal{T}_b$) on TU-Berlin-Retr dataset [15].

Method	Dim.	mAP (%)
Partial sharing convnet (stage 4)	256	**65.99**
Partial sharing convnet (stage 3)	256	**63.37**
Sketchy triplet [6]†	1024	62.02
Partial sharing convnet (stage 2)	256	**57.15**
LKS [7]	2400	32.51
Rst-SP-SHELO [22]	3060	29.36
SHELO [45]	1296	27.66
HoG [49]	900	23.55
HELO [40]	72	14.32

Table 4. SBIR comparision results on Saavedra dataset [40].

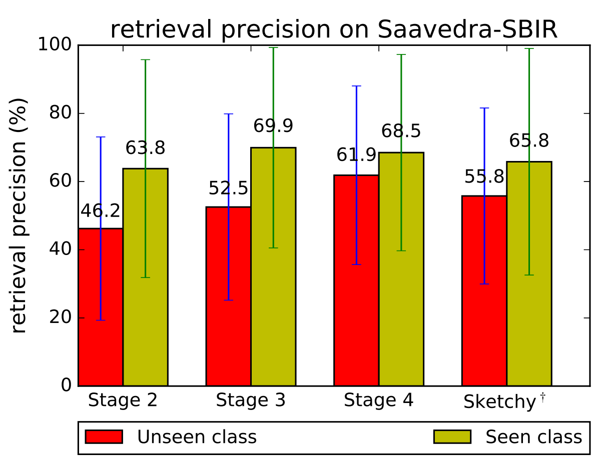


Fig. 11. Average retrieval precision by query groups on Saavedra-SBIR.

our model achieves the smallest performance gap between the "seen" and "unseen" groups, which further demonstrates the generalization capability of our model.

5. Conclusion

We proposed a hybrid CNN exploiting both contrastive and triplet loss architectures to learn a joint sketch-photo embedding suitable for measuring visual similarity in SBIR. We presented comprehensive experiments exploring variants of our triplet CNN, contrasting appropriate strategies for weight sharing, dimensionality reduction, and training data pre-processing and reporting on the generalization capabilities across categories including object categories unseen during training. Training sketches were derived from the two largest available sketch datasets: the TU-Berlin dataset of Eitz *et al.* and the Sketchy dataset of Sangkloy *et al.* [6]. The model was trained using exemplar triplets formed using these query sketches augmented by positive and negative training photos from the web. Our optimal network configuration comprised a triplet architecture with branch structure derived from GoogLeNet with partially-shared weights, and achieved 53.3% mAP over the Flickr15k benchmark; more than 17% increase in performance accuracy over the published state of the art (Table 2).

Further work might build upon this performance gain exploring multi-domain learning, for example sketch-photo-3D models mapping or multi-style work-art retrieval. Recently deep convolutional generative-adversarial networks (DC-GAN) have shown great potential for sketch driven synthesis [50] and so might offer an interesting avenue for SBIR as an alternative deep representation for sketch-photo matching. Currently DC-GANs suffer limitations in object class diversity when trained that could be investigated as here.

Acknowledgments

This work was supported in part via an EPSRC doctoral training studentship (EP/M508160/1) and in part by FAPESP (grants 2016/16111-4, 2017/10068-2 and 2013/07375-0).

References

[1] Hu, R, Barnard, M, Collomosse, JP. Gradient field descriptor for sketch based retrieval and localization. In: Proc. ICIP; vol. 10. 2010, p. 1025–1028.
[2] Eitz, M, Hays, J, Alexa, M. How do humans sketch objects? Proc ACM SIGGRAPH 2012;:44:1–44:10.

[3] Hu, R, Collomosse, J. A performance evaluation of gradient field HOG descriptor for sketch based image retrieval. Computer Vision and Image Understanding (CVIU) 2013;117(7):790–806. doi:10.1016/j.cviu.2013.02.005.
[4] Krizhevsky, A, Sutskever, I, Hinton, G. Imagenet classification with deep CNNs. In: Proc. NIPS. 2012,.
[5] Yu, Q, Liu, F, Song, YZ, Xiang, T, Hospedales, TM, Loy, CC. Sketch me that shoe. In: Proc. CVPR. IEEE; 2016,.
[6] Sangkloy, P, Burnell, N, Ham, C, Hays, J. The sketchy database: learning to retrieve badly drawn bunnies. Proc ACM SIGGRAPH 2016;35(4):119.
[7] Saavedra, JM, Barrios, JM. Sketch based image retrieval using learned keyshapes. In: Proc. BMVC. 2015,.
[8] Chatfield, K, Simonyan, K, Vedaldi, A, Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In: Proc. BMVC. 2014,.
[9] Flickner, M, Sawhney, H, Niblack, W, Ashley, J, Huang, Q, Dom, B, et al. Query by image and video content: The qbic system. Computer 1995;28(9):23–32.
[10] Jacobs, CE, Finkelstein, A, Salesin, DH. Fast multiresolution image querying. In: Proc. ACM SIGGRAPH. 1995, p. 277–286.
[11] Sciascio, ED, Mingolla, G, Mongiello, M. Content-based image retrieval over the web using query by sketch and relevance feedback. In: Visual Information and Information Systems. Springer; 1999, p. 123–130.
[12] Collomosse, JP, McNeill, G, Watts, L. Free-hand sketch grouping for video retrieval. In: Intl. Conf on Pattern Recognition (ICPR). 2008,.
[13] Bimbo, AD, Pala, P. Visual image retrieval by elastic matching of user sketches. IEEE Trans PAMI 1997;19(2):121–132.
[14] Collomosse, JP, McNeill, G, Qian, Y. Storyboard sketches for content based video retrieval. In: Proc. ICCV. 2009, p. 245–252.
[15] Eitz, M, Hildebrand, K, Boubekeur, T, Alexa, M. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. IEEE Trans Visualization and Computer Graphics 2011;17(11):1624–1636.
[16] Eitz, M, Hildebrand, K, Boubekeur, T, Alexa, M. A descriptor for large scale image retrieval based on sketched feature lines. In: Proc. SBIM. 2009, p. 29–36.
[17] Bui, T, Collomosse, J. Scalable sketch-based image retrieval using color gradient features. In: Proc. ICCV Workshops. 2015, p. 1–8.
[18] Cao, Y, Wang, C, Zhang, L, Zhang, L. Edgel index for large-scale sketch-based image search. In: Proc. CVPR. IEEE; 2011, p. 761–768.
[19] Sun, X, Wang, C, Xu, C, Zhang, L. Indexing billions of images for sketch-based retrieval. In: Proc. ACM Multimedia. 2013,.
[20] Qi, Y, Song, YZ, Xiang, T, Zhang, H, Hospedales, T, Li, Y, et al. Making better use of edges via perceptual grouping. In: Proc. CVPR. 2015,.
[21] Saavedra, JM, Bustos, B. Sketch-based image retrieval using keyshapes. Multimedia Tools and Applications 2014;73(3):2033–2062.
[22] Saavedra, JM. Rst-shelo: sketch-based image retrieval using sketch tokens and square root normalization. Multimedia Tools and Applications 2017;76(1):931–951.
[23] Yu, Q, Yang, Y, Song, YZ, Xiang, T, Hospedales, TM. Sketch-a-net that beats humans. In: Proc. BMVC. IEEE; 2015,.
[24] Bhattacharjee, SD, Yuan, J, Hong, W, Ruan, X. Query adaptive instance search using object sketches. In: Proc. ACM Multimedia. ACM; 2016, p. 1306–1315.
[25] Wang, F, Kang, L, Li, Y. Sketch-based 3d shape retrieval using convolutional neural networks. In: Proc. CVPR. 2015, p. 1875–1883.
[26] Qi, Y, Song, YZ, Zhang, H, Liu, J. Sketch-based image retrieval via siamese convolutional neural network. In: Proc. ICIP. IEEE; 2016, p. 2460–2464.
[27] Wang, J, Song, Y, Leung, T, Rosenberg, C, Wang, J, Philbin, J, et al. Learning fine-grained image similarity with deep ranking. In: Proc. CVPR. 2014, p. 1386–1393.
[28] Schroff, F, Kalenichenko, D, Philbin, J. Facenet: A unified embedding for face recognition and clustering. In: Proc. CVPR. 2015, p. 815–823.
[29] Wang, X, Gupta, A. Unsupervised learning of visual representations using videos. In: Proc. ICCV. 2015, p. 2794–2802.
[30] Gordo, A, Almazán, J, Revaud, J, Larlus, D. Deep image retrieval: Learning global representations for image search. In: Proc. ECCV. Springer; 2016, p. 241–257.
[31] Bui, T, Ribeiro, L, Ponti, M, Collomosse, J. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. CVIU 2017;.
[32] Radenović, F, Tolias, G, Chum, O. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In: European Conference on Computer Vision. Springer; 2016, p. 3–20.
[33] Simonyan, K, Zisserman, A. Very deep convolutional networks for large-scale image recognition. CoRR 2014;abs/1409.1556.
[34] Szegedy, C, Liu, W, Jia, Y, Sermanet, P, Reed, S, Anguelov, D, et al. Going deeper with convolutions. In: Proc. CVPR. 2015, p. 1–9.
[35] Seddati, O, Dupont, S, Mahmoudi, S. Quadruplet networks for sketch-based image retrieval. In: Proc. ACM ICMA. ACM; 2017, p. 184–191.
[36] Zeiler, MD, Fergus, R. Visualizing and understanding convolutional networks. In: Proc. ECCV. Springer; 2014, p. 818–833.
[37] Yosinski, J, Clune, J, Nguyen, A, Fuchs, T, Lipson, H. Understanding neural networks through deep visualization. In: Proc. ICML workshop. 2015,.
[38] Hadsell, R, Chopra, S, LeCun, Y. Dimensionality reduction by learning an invariant mapping. In: Proc. CVPR; vol. 2. IEEE; 2006, p. 1735–1742.
[39] Bengio, Y, Louradour, J, Collobert, R, Weston, J. Curriculum learning. In: Proc. ICML. ACM; 2009, p. 41–48.
[40] Saavedra, JM, Bustos, B. An improved histogram of edge local orientations for sketch-based image retrieval. In: Joint Pattern Recognition Symposium. Springer; 2010, p. 432–441.
[41] Lam, L, Lee, SW, Suen, CY. Thinning methodologies-a comprehensive survey. IEEE Trans PAMI 1992;14(9):869–885.
[42] Jia, Y, Shelhamer, E, Donahue, J, Karayev, S, Long, J, Girshick, R, et al. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:14085093 2014;.
[43] Belongie, S, Malik, J, Puzicha, J. Shape matching and object recognition using shape contexts. IEEE Trans PAMI 2002;24(4):509–522. doi:10.1109/34.993558.
[44] Shechtman, E, Irani, M. Matching local self-similarities across images and videos. In: Proc. CVPR. 2007, p. 1–8. doi:10.1109/CVPR.2007.383198.
[45] Saavedra, JM. Sketch based image retrieval using a soft computation of the histogram of edge local orientations (s-helo). In: Proc. ICIP. IEEE; 2014, p. 2998–3002.
[46] Wang, S, Zhang, J, Han, TX, Miao, Z. Sketch-based image retrieval through hypothesis-driven object boundary selection with hlr descriptor. IEEE Trans Multimedia 2015;17(7):1045–1057.
[47] Tolias, G, Chum, O. Asymmetric feature maps with application to sketch based retrieval. Proc CVPR 2017;.
[48] Lowe, DG. Distinctive image features from scale-invariant keypoints. Intl Journal Computer Vision (IJCV) 2004;60(2):91–110.
[49] Dalal, N, Triggs, B. Histograms of oriented gradients for human detection. In: Proc. CVPR; vol. 1. 2005, p. 886–893.
[50] Isola, P, Zhu, JY, Zhou, T, Efros, AA. Image-to-image translation with conditional adversarial networks. arxiv 2016;.

Supplementary Material

Video S1. SBIR demo [CAG-D-17-00301-supplm_video.mp4].

A video demo of our proposed model, InceptionV1-InceptionV1 256-D partial sharing from inception-4e, depicts SBIR by an amateur sketcher on a tablet running Android 5.1.1. In several parts of the demo, the sketcher intentionally draws different objects by adding incremental line strokes to their existing sketches, and observes changes in the returned results. This drawing procedure helps sketchers to refine their queries, also to understand which strokes are important for retrieving desired photo images.