The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies

Patsorn Sangkloy[1], Nathan Burnell[2], Cusuh Ham[1], James Hays[1]

[1]Georgia Institute of Technology [2]Brown University

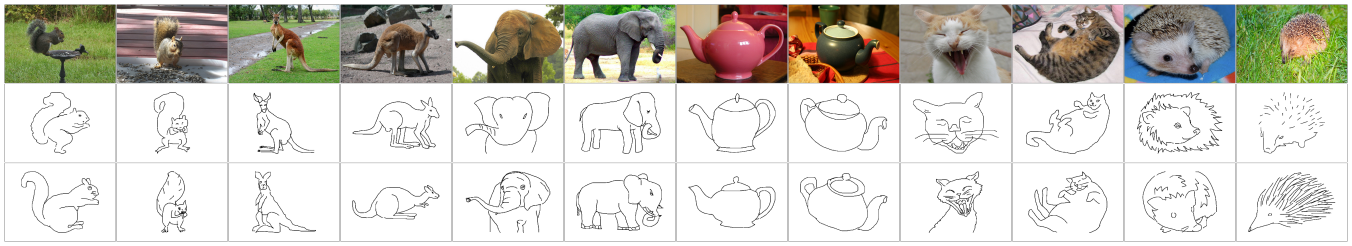

Figure 1: *Samples of photo–sketch pairs from the Sketchy database. We show two photos from the* Squirrel, Kangaroo, Elephant, Teapot, Cat, *and* Hedgehog *categories. Below each photo are two sketches produced by crowd workers. Notice the variation in sketches across object instances and between artists. We use the Sketchy database as training and test data for sketch-based image retrieval.*

Abstract

We present the *Sketchy database*, the first large-scale collection of sketch-photo pairs. We ask crowd workers to sketch particular photographic objects sampled from 125 categories and acquire 75,471 sketches of 12,500 objects. The Sketchy database gives us fine-grained associations between particular photos and sketches, and we use this to train cross-domain convolutional networks which embed sketches and photographs in a common feature space. We use our database as a benchmark for fine-grained retrieval and show that our learned representation significantly outperforms both hand-crafted features as well as deep features trained for sketch or photo classification. Beyond image retrieval, we believe the Sketchy database opens up new opportunities for sketch and image understanding and synthesis.

Keywords: Sketch-based image retrieval, Deep learning, Siamese network, Triplet network, Image synthesis

Concepts: •**Computing methodologies** → *Image representations; Image processing;*

1 Introduction

The goal of Sketch-based image retrieval is to allow non-artist users to draw visual content (usually objects) and then find matching examples in an image collection. Sketch-based image retrieval is an alternative or a complement to widely used language-based image querying (e.g. Google image search). In the computer graphics community, sketch-based image retrieval has been used to drive image synthesis approaches such as Sketch2Photo [Chen et al. 2009] and PhotoSketcher [Eitz et al. 2011b]. Sketch-based image retrieval has been studied for nearly 25 years [Kato et al. 1992], but is especially relevant as touch and pen-based devices have proliferated over the last few years.[1]

Sketch-based image retrieval is challenging because it requires comparison across two domains (sketches and photos) that are individually difficult to understand and have distinct appearance. Typical approaches propose a hand-designed feature, usually focused on edges or gradients, which is somewhat invariant across domains. But as we will show (Section 5) there is a lot of room for improvement over such features.

The primary reason sketch-to-image comparison is hard is that humans are not faithful artists. We tend to draw only salient object structures and we tend to draw them poorly. Shapes and scales are distorted. Object parts are caricatured (big ears on an elephant), anthropomorphized (smiling mouth on a spider), or simplified (stick-figure limbs). Nonetheless these sketches are usually understandable to other humans.

It is appealing to try and *learn* the correspondence between human sketches and photographic objects since the task seems to require high-level understanding. Recent progress in deep convolutional networks has enabled better understanding of sketches and object images, individually. The 250 categories of sketches collected by Eitz et al. [2012a] and the 1000 categories in the ImageNet challenge [Russakovsky et al. 2015] can be recognized with roughly human-level accuracy. This suggests a "retrieval by categorization" approach in which relevant images are returned if they appear to be the same category as a query sketch. In fact, this strategy is consistent with common sketch-based image retrieval benchmarks [Hu and Collomosse 2013] where retrieval results are correct as long as they are the same category.

However, we argue that sketch-based image retrieval needs to go beyond category recognition. If a user wants objects of a particular category, then language already gives them an efficient way to find huge amounts of relevant imagery. The appeal of sketching is that it allows us to specify *fine-grained* aspects of an object – pose, parts, sub-type (e.g. office chair versus dining chair) – and many of these fine-grained attributes are clumsy to specify with language

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
SIGGRAPH '16 Technical Paper,, July 24 - 28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925954

[1]Touch-enabled smartphones and tablets outnumber desktop and portable PCs by 4 to 1 as of 2014, while the market was roughly equal in 2010 [Mainelli et al. 2015].

(e.g. "office chair with no arms viewed from slightly above and to the side" or "castle with two pointy towers and a crenelated battlement joining them"). Even if such text descriptions did encapsulate user intent, fine-grained text-based image retrieval is an open problem. This need to retrieve semantically and structurally appropriate objects leads Sketch2Photo [Chen et al. 2009] to synthesize scenes using a combination of language and sketch – language typically specifies the category and sketch specifies the shape. We believe sketches alone can be sufficient.

Our goal in this paper is to learn a cross-domain representation for sketches and photos which is reliable not just at retrieving objects of the correct category, but also objects with *fine-grained* similarity to the sketch. We utilize deep learning techniques which have led to dramatic improvements in various recognition tasks, but which require large amounts of supervised training data. In particular, for our cross-domain deep learning we will need thousands of pairs of matching sketches and photos. Because no such database exists we ask crowd workers to draw thousands of photographic objects spanning 125 categories. We do not allow participants to trace photos, but instead reveal and then hide photos so that they must sketch from memory similar to the way in which a user of a sketch-based image retrieval system would be drawing based on some mental image of a desired object. The cross-domain representation we learn proves effective for sketch-based image retrieval and we believe the Sketchy database supports many interesting investigations into sketch and image understanding.

We make the following contributions:

- We develop a crowd data collection technique to collect representative (often bad!) object sketches prompted by but not traced from particular photos. The Sketchy database contains 75,471 sketches of 12,500 objects spanning 125 categories.[2]
- We demonstrate the first deep learning approach to sketch-based image retrieval. We learn a common feature space for sketches and photos which enables not only sketch-based image retrieval but also image-based sketch retrieval or sketch-to-sketch and photo-to-photo comparisons.
- We show that our learned representation leads to significantly better fine-grained sketch retrieval than existing methods and even outperforms a hypothetical ideal "retrieval by categorization" method.

Outline. In the next section we discuss related work. Section 3 describes the creation of the Sketchy database. Section 4 describes our deep learning experiments. Section 5 evaluates our learned representation for the task of sketch-based image retrieval. Section 6 shows sketch-constrained average images inspired by AverageExplorer [Zhu et al. 2014]. Section 7 discusses limitations and possible future applications of the Sketchy database.

2 Related Work

Sketch-based image retrieval. Numerous representations have been proposed to retrieve images from sketch queries [Kato et al. 1992; Del Bimbo and Pala 1997; Sclaroff 1997] or color drawings [Jacobs et al. 1995]. See Smeulders et al. [2000] for a survey of classical approaches. More recent methods propose increasingly sophisticated feature representations, often inspired by object recognition approaches in the computer vision community [Cao et al. 2011; Shrivastava et al. 2011; Cao et al. 2013; Saavedra and Barrios 2015].

[2]the database can be downloaded from http://sketchy.eye.gatech.edu/

Benchmarks for sketch-based image retrieval are fairly small, e.g. 43 sketch-photo pairs [Eitz et al. 2010] or 31 sketches each with 40 photos ranked by similarity [Eitz et al. 2011a]. The Flick15k dataset [Hu and Collomosse 2013] contains 330 sketches and 14,660 photos spanning 33 categories, but there are no fine-grained associations across domain – the benchmark is equivalent to categorization. These benchmarks have been useful to the field but are not large enough to *learn* from.

The tendency to evaluate sketch-based image retrieval as category retrieval was noted by Li et al. [2014] and they propose a "fine-grained" retrieval method based on deformable part models [Felzenszwalb et al. 2010] trained on the 14 overlapping categories of the Pascal VOC [Everingham et al. 2010] and Eitz 2012 datasets. At training time there is still no instance level sketch-photo association, but their test set scores sketch-photo pairs in terms of viewpoint, zoom, pose, and shape similarity. We share the motivation for "fine-grained" retrieval and we evaluate our representation learned from the Sketchy database on their benchmark (Table 1).

The most similar related work is the concurrent "Sketch Me that Shoe" [Yu et al. 2016] which also collects a database of sketch-photo pairs and uses deep learning to learn a shared embedding. Their dataset is smaller – 1,432 sketch-photo pairs of shoes and chairs – but more densely annotated with 32,000 triplet rankings. Because the dataset is smaller, edge detection is used to render photos like sketches instead of *learning* the entire cross-domain transformation as we do. The paper also proposes more aggressive, domain-specific "jittering" to amplify the value of each training sketch.

Sketch classification. "How do Humans Sketch Objects?" [Eitz et al. 2012a] introduced a dataset of 20,000 sketches spanning 250 categories and demonstrated that bag-of-features representations built on gradient features could achieve reasonable classification accuracy (56%). Schneider and Tuytelaars [2014] showed that Fisher vector encoding of local features significantly improved recognition accuracy (69%). Sketch-a-Net [Yu et al. 2015] showed that deep features can surpass human recognition accuracy – 75% compared to the 73% accuracy from crowd workers in Eitz 2012. Su et al. [2015] focus on 3D model classification with deep features but show that their network can be adapted to match the state of the art in sketch recognition or sketch-based 3D model retrieval. We use the Eitz 2012 dataset to pre-train the sketch half of our cross-domain embedding approach.

Deep learning for cross-domain embedding. Our technical approach is similar to recent cross-domain embedding methods that train deep networks to learn a common feature space for Sketches and 3D models [Wang et al. 2015], ground and aerial photographs [Lin et al. 2015], Iconic and in-the-wild product photos [Bell and Bala 2015], and Images and 3D models [Li et al. 2015]. We experiment with the tools used in these works such as Siamese networks trained with contrastive loss [Chopra et al. 2005; Hadsell et al. 2006], but find that triplet or ranking loss [Wang et al. 2014] performs better. Our best performing method combines the Triplet loss with a classification loss similar to Bell and Bala's [2015] combination of Siamese and classification losses.

Sketch-based image retrieval for image synthesis. We are motivated by Sketch2Photo [Chen et al. 2009] and PhotoSketcher [Eitz et al. 2011b], which synthesize scenes by compositing objects and backgrounds retrieved based on user sketches. PoseShop [Chen et al. 2013], follow on work from Sketch2Photo, addresses the fact that bad user sketches cannot be matched reliably by letting users pose a 2D skeleton and then generating a contour query from a mesh attached to the 2D skeleton. Improved sketch-based object retrieval

Figure 2: *A spectrum of "sketchability" for three ImageNet categories:* horse*,* apple*, and* rabbit*. This subjective ranking is determined by answering the question, "How easily could a novice artist capture the subject category and pose?" The most difficult photographs, such as those shown on the far right of the figure, are excluded from our data set.*

would make these approaches simpler or more reliable.

Sketch-photo pairs for sketch synthesis. Berger et al. [2013] and Limpaecher et al. [2013] collect expert and amateur portrait drawings, respectively, to aid in the creation of new portraits. The 14,270 face portraits collected through a Facebook game by Limpaecher et al. [2013] is one of the largest collections of sketches to date.

3 Creating the Sketchy Database

In this section we describe the creation of the Sketchy database, which spans 125 categories and consists of 12,500 unique photographs of objects and 75,471 human sketches of objects inspired by those photographs.

3.1 Category Selection

In order to choose the 125 categories in our data set, we use the same criteria defined in "How Do Humans Sketch Objects?" [Eitz et al. 2012a]: exhaustive, recognizable, and specific. That is, the categories should cover a large number of common objects and each category should have recognizable sketch representations. In addition, we add a "sketchability" criterion (Figure 2). Some object categories may be fairly easy to sketch from memory, but photographs of those objects tend to be too challenging to sketch or the resulting sketches may be too uninformative due to the nature of photographs common to those objects. This concept is illustrated in Figure 2 and further discussed in the following sections.

To start, we consider all ImageNet [Russakovsky et al. 2015] categories for which there are bounding box annotations, with preference given to categories contained within the Eitz 2012 sketch data set. We hope our data set will both complement and extend these data sets. Ultimately, 125 categories are included in our data set. Of these, 100 categories exist within the Eitz 2012 data set. Where appropriate, multiple, related ImageNet categories (e.g. specific dog breeds) are combined into a single category in order to add visual diversity and increase the number of "sketchable" photographs.

3.2 Photograph Selection

We cannot expect novice artists to draw photographs of horse eyes and crocodile teeth in meaningful ways, yet these extreme photographs are prevalent in Internet-scale image data sets. In order

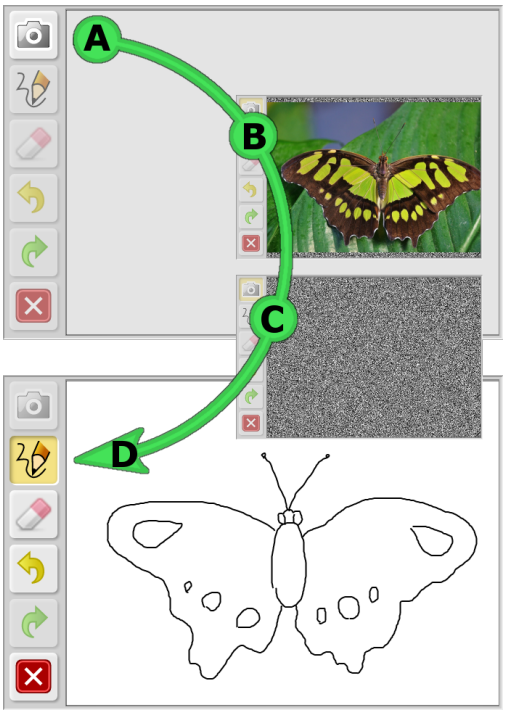

Figure 3: *Sketch collection interface. A participant initially sees a blank canvas. (a) Pressing a button reveals (b) a photo for 2 seconds followed by (c) a noise mask for one second. (d) The participant then uses pencil, eraser, and undo tools to create their sketch.*

to select appropriate photographs for our data set, we first eliminate all photographs that do not have exactly one bounding box annotation. Next, we manually review the remaining photographs and eliminate those with 1) disturbing or inappropriate content, 2) poor or degraded image quality, 3) significant manipulation or watermarks, 4) incorrect category label, and/or 5) ambiguous content due to occlusion or object pose. Overall, we review a total of 69,495 photographs and deem 24,819 as "sketchable". This process results in a median of 147 sketchable photographs per category. Categories with fewer than 100 sketchable photographs are not included in our data set. As part of this process, a volunteer annotates each remaining photograph with a subjective "sketchability" score, ranging from 1 (very easy to sketch) to 5 (very difficult to sketch). Each category's 100 photographs are chosen at random from this pool with a targeted distribution of 40 very easy, 30 easy, 20 average, 10 hard, and 0 very hard photographs.

3.3 Sketch Collection

The creation of sketch-photo pairs is the most critical challenge of creating the Sketchy database. There are two broad strategies – prompt the creation of sketches from particular photos [Eitz et al. 2010] or have people associate existing sketches to photos, e.g. by ranking a list of potentially matching photos [Eitz et al. 2011a]. We choose the first strategy because it is better able to create fine-grained, instance-level associations. With the second strategy, there may not exist a photo that is particularly similar to a sketch.

However, the first strategy, photo-prompted sketch creation, is somewhat the inverse of motivating task – sketch-based image retrieval. Does a user drawing of a particular photo resemble a sketch retrieval query? For instance, a naive approach to sketch creation would be to have users trace over a particular photo. Such draw-

ings would effectively be boundary annotations as in the Berkeley Segmentation Dataset [Martin et al. 2001]. If we thought faithful object boundaries were satisfactory "sketch" training data we could use existing segment annotations from datasets such as LabelMe/SUN [Xiao et al. 2014] or MS COCO [Lin et al. 2014]. As Figure 1 shows, the sketches we collect are very different from such annotations. We also attempt to train on MS COCO boundaries with no success in Section 5.

The key to collecting "realistic" sketches is to prompt workers with a particular photo but then hide it so they must draw from memory. This strategy was used in Eitz et al. [2010] to collect 43 sketches and also in Antol et al. [2014] to collect "'clipart" annotations which correspond to the most salient scene structures. The choice of which object structures people preserve (or hallucinate) when sketching a particular photo is interesting in itself, independent from the goal of sketch-based image retrieval, and could support research on human visual memory of objects [Brady et al. 2008; Brady et al. 2013]. We use the fact that the Sketchy database implicitly encodes image salience to produce visualizations in Section 6.

Figure 3 shows our sketch collection interface. Each participant is provided with a randomly selected category name and a blank canvas on which to sketch. Upon pressing a specified button, the participant is shown a photograph containing an example of the selected category. The photograph is only visible for two seconds, but the participant can view it as many times as needed. However, each viewing clears the sketch canvas. In order to discourage rote boundary reproduction, a noise mask is displayed after the photograph is revealed and before the participant may begin sketching. Noise masking, often used in psychology experiments, aims to destroy low-level visual representations in visual working memory [Grill-Spector and Kanwisher 2005; Nieuwenstein and Wyble 2014]. In our case, this prevents participants from "tracing" an afterimage in visual working memory and hopefully produces more diverse and realistic sketches.

A participant is instructed to 1) sketch the named subject object with a pose similar to that of the object in the photograph, 2) sketch only the subject object, and 3) avoid shading in regions. Since we have bounding box details for the object in the photograph, we are not concerned with sketch size nor location in the canvas as they can be aligned after the fact. The sketch canvas supports touch-enabled devices and provides a means to sketch, undo, redo, clear, and erase. Each sketch is stored as an SVG file. We extend the SVG format to include high resolution time details; not only do we record the stroke start and end times, but also fine-grained timing along each stroke. Figure 4 examines the drawing tendencies of our participants. Prior research has shown that stroke order and length are important in determining the relative importance of a given stroke [Yu et al. 2015]. If the speed at which a stroke is drawn indicates care and/or concentration, then speed, too, may be useful in determining the relative importance of a stroke. The vector representation of strokes allows us to re-render the drawings in various ways (different stroke widths, stroke width related to velocity, etc.) but we maintain the same fixed width stroke representation as used in the data gathering through all experiments. Varying the rendering style did not significantly influence the learning experiments in Section 4, but we believe there is potential to artificially increase the training set size through numerous stroke-based augmentations as in concurrent work [Yu et al. 2016].

Participants. When it comes to sketching a photograph, there is no single correct answer. The artist's skill, motivation, subject familiarity, and input device can all influence the resultant sketch. In order to capture this diversity we collect five sketches per photograph each from a different participant. We use Amazon Mechanical Turk (AMT) to increase the number of potential participants and resultant sketch diversity. We use a qualification test to ensure each potential participant understands the sketching process. The qualification test asks the candidate to sketch three different photographs, each carefully chosen to test the candidate's understanding of one or more rules. We manually review these qualification results for compliance and allow those who pass to work on our sketch collection tasks. We receive 1,204 qualification test submissions, from which we identify 644 qualified individuals. Over the course of six months, these participants collectively spent 3,921 hours sketching for our data set.

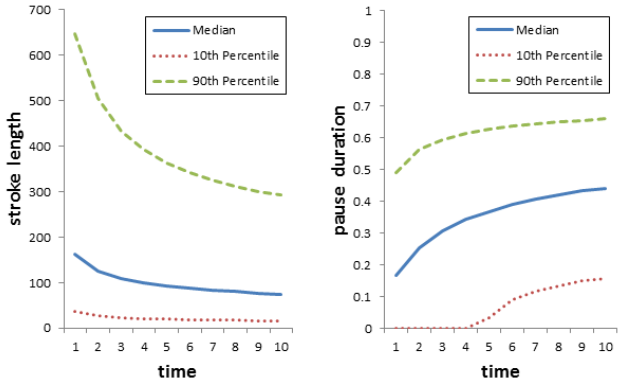

Figure 4: *Left: as in Eitz 2012 we find that the participants follow a coarse-to-fine sketching strategy and draw shorter strokes over time. Right: Over their drawing time, participants spend increasingly more time deliberating between strokes as measured by the fraction of time spent idle. For both plots, the duration of each sketch session is normalized to the same time duration.*

Data Validation. Even though we use a qualification test to screen the crowd workers, mistakes, misunderstandings, and abuses are inevitable. Instead of using the crowd to validate its own work, we opt to manually review all sketches. We use a custom software tool to display each photograph along with all its sketches and then tag each sketch in one of five ways: 1) correct, 2) contains environment details or shading, 3) incorrect pose or perspective, 4) ambiguous, or 5) erroneous. All of these sketches remain in the data set, even if deemed incorrect in some way. Inclusion is up to the judgment of the database user and is highly task dependent. The only truly incorrect sketches are those marked as erroneous. In order to increase the overall quality of the data set and bring the total number of sketches per photograph closer to five, we collect additional sketches to replace those that are not tagged as correct. This results in a total of 75,471 sketches, with 64,560 correct, 6,249 ambiguous, 2,683 with an incorrect pose, 1,061 including environment details, and 918 erroneous.

Comparison to Eitz 2012. Our participants spent a similar amount of time drawing sketches as is reported in Eitz 2012. The median sketch time is 85 seconds, with the 10th and 90th percentile at 41 and 281 seconds (compared to 86, 31, and 280 seconds, respectively). The median number of strokes per sketch is 14, compared with 13 in Eitz 2012.

While the stroke-level statistics are similar to the 20,000 sketches of Eitz 2012, we believe the distribution of sketches is quite different. Eitz 2012 prompted workers with an object category and nothing more. The resulting sketches are very iconic. For example 85% of buses are drawn directly from the side and the remainder from 45°. Not a single duck is drawn in flight. 80% of calculators are upright and viewed from the front. People chose easy-to-draw poses

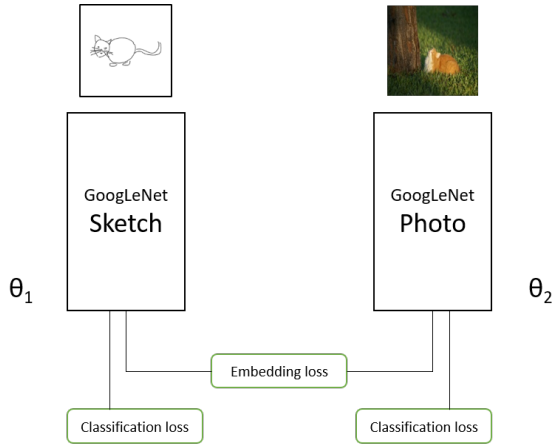

Figure 5: *Our adaptation of the GoogLeNet architecture to cross-domain embedding. The embedding loss can be either contrastive loss (as used in a Siamese network) or triplet loss. For triplet loss, the photo branch would take two inputs, one for the matching photo and one for a dissimilar photo. The sketch and photo networks learn independent weights θ_1 and θ_2.*

and viewpoints, which is partly why the dataset is fairly easy as a recognition benchmark. But the Sketchy database contains objects in a variety of poses, states, and viewpoints because workers were prompted with particular photographs. While the ImageNet photos are themselves somewhat iconic, they still lead to greater sketch diversity than free recall. Working from a particular photo also constrained the sketches to be somewhat less caricatured (fewer big teeth and ears on rabbits).

4 Learning a Cross-domain Mapping

Overview. In this section we use the Sketchy database to learn a shared embedding for sketches and photos such that distances in the learned feature space are related to structural and semantic similarity between sketches and photos. We follow the trend of recent works which use deep convolutional networks (CNNs) to learn cross-domain embeddings [Wang et al. 2015; Lin et al. 2015; Bell and Bala 2015; Li et al. 2015], but the details of our network architectures and training strategies vary considerably.

Our deep networks which achieve the highest performance on our benchmarks required rather complex training regimes. There are two reasons for this: (1) While the Sketchy database is large relative to existing sketch databases, deep networks have tens of millions of free parameters to learn and thus demand large amounts of training data. We see benefits but also complications from pre-training on additional datasets. (2) We have two separate notions of similarity – instance-level similarity between a sketch and the exact photo that elicited it, and category-level similarity between a sketch and all photos in that category. Making use of both forms of supervision during training increases performance at the cost of added complexity.

We examine combinations of three different loss functions for training deep networks – Siamese loss, Triplet loss, and Classification loss. We first give an overview of these loss functions.

Siamese Network. In a Siamese network [Chopra et al. 2005; Hadsell et al. 2006] a pair of convolutional networks each accept an input. Supervision is binary – either the input pair should be similar or dissimilar. Siamese networks use a "contrastive" loss function of the form $L = l * d(S, I+) + (1 - l) * max(0, m - d(S, I-))$ where S is an embedded sketch, $I+$ is an embedded image of the same object instance, $I-$ is an embedded image of a different object instance, $d()$ is Euclidean distance, m is a margin, and $l \in \{0, 1\}$ is label with 1 for positive and 0 for negative pair. Dissimilar sketch–image pairs will be pushed apart unless their distance is already greater than the margin. Similar sketch–image pairs will be pulled together in the feature space.

Triplet Network. Triplet networks [Wang et al. 2014] are similar to Siamese networks except that the supervision is of the form "input a should be closer to input b than to input c". Triplet networks use a "ranking" loss function of the form $L = max(0, m + d(S, I+) - d(S, I-))$. This ranking loss function can express more fine-grained relationships than the Siamese loss which can only say pairs of points should be near or far.

Traditionally, while a Siamese network can be conceptualized as two networks and a Triplet network can be conceptualized as three networks there is really only a single network used repeatedly for all inputs. This makes sense when the embedding being learned is not cross-domain. For example, Hadsell et al. [2006] addresses the within-domain task of face verification where input faces are classified as same or different identity. But our inputs – sketches and photos – are dramatically different and it makes sense to train two networks with independent weights to embed the sketches and photos. This is a departure from most previous deep embedding works, even those that are cross-domain. Lin et al. [2015] find that independent weights barely improved ground-to-aerial image matching and Bell and Bala [2015] use shared weights for iconic-to-internet product photo matching.

Using shared weights also makes sense if you can convert between domains before passing inputs into the CNNs. For example, Wang et al. [2015] addresses sketch-based 3D model retrieval and renders the 3D models like sketches. An analogous approach for us would be to run edge detection on photos before using a Siamese or Triplet network with shared weights. Yu et al. [2016] take this approach. But we suspect that detected edges are not a good proxy for the image structures that humans draw, and it makes more sense to let the deep networks learn the cross-domain transformation from data.

Classification loss. A successful sketch-based image retrieval system needs to respect both the semantics (e.g. object category) and fine-grained details (e.g. pose, shape, viewpoint, and other attributes) of a user query sketch. If a user sketches a horse, then it is not satisfactory to retrieve cows, zebras, and dogs in the same pose. While the Siamese or Triplet losses encourage CNNs to be sensitive to fine-grained sketch-to-photo similarities, we can improve performance by including a *classification loss* when training our deep networks. We use the traditional "softmax" classification loss with the 125 categories of the Sketchy database and this helps ensure that retrieval results match the category of a query. The same approach was used by Bell and Bala [2015] to improve image retrieval.

Network architectures. We experiment with two deep network architectures – AlexNet [Krizhevsky et al. 2012] as implemented in Caffe [Jia et al. 2014] and the deeper GoogLeNet [Szegedy et al. 2014]. We omit the auxiliary classification loss layers of the GoogLeNet architecture since their effect was negligible in our experiments. Figure 5 visualizes our cross-domain network.

4.1 Data preparation and Pre-training

Pre-training. We first train each subnetwork for sketch and image classification. The networks independently learn weights appropriate for each domain without any initial constraints for common embedding. We start with AlexNet or GoogLeNet trained on

ImageNet. The sketch network is fine-tuned to recognize the 250 categories from Eitz 2012 [2012a]. We divide the dataset into a train/test split of 18k/2k and after fine-tuning achieve 77.29% accuracy with AlexNet and 80.85% accuracy with GoogLeNet (this represents the state of the art on Eitz 2012 to the best of our knowledge).

Cross-domain classification. Up to this point, each network is trained separately for their specific domain and the 'features' computed by these networks are not comparable (or even the same dimensionality at the highest layer). As before, we train the network with classification loss only, but switch to training using the 125 sketch categories of the Sketchy database for sketch network branch. We also collect 1000 Flickr photos for each of our 125 categories and use them to further train image branch. This means that the activations at the top of each network are comparable 125-dimensional features, where each dimension of the feature is a measure of confidence of the presence of a particular category. These 125 dimensional features are used as a "retrieval by categorization" baseline which we evaluate in Section 5.

Fine-grained training data. As discussed in Section 3, the Sketchy database provides fine-grained correspondence between sketches and photos that can be used as positive pairs for training our cross-domain CNNs. We hold out 10% of the data for testing. The only 'jittering' we use is mirroring. Positive sketch-photo pairs are mirrored jointly because we do not want to be mirror invariant. After mirroring, the 90% of data used for training provides more than one hundred thousand positive pairs (22,500 images with *at least* 5 sketches each) and more than a billion negative pairs.

Sketch Normalization. We uniformly scale and center sketches from the Sketchy database so that the learned representation is not sensitive to the absolute location and scale of a sketch. This normalization makes our benchmark harder because it breaks the spatial correspondence between sketches and photos. But in a realistic sketch-based image retrieval scenario we assume the user wants to be invariant to location and scale and instead wants to match pose, shape, and other attributes. We release the aligned sketch-photo data, though, as it would be more appropriate training data for some scenarios (e.g. learning to render photos like sketches).

4.2 Training Cross-Domain Embeddings

We now train deep networks to embed sketches and photos into a 1024 dimensional space with Siamese and Triplet loss functions. These Siamese and Triplet losses are used simultaneously with a softmax classification loss.

Training with Siamese contrastive loss. The inputs for Siamese training are a sketch-photo pair, each passed to the corresponding subnetwork, and supervision as to whether the pair is matching or not. The contrastive loss has a free parameter, the margin, which can significantly affect the learned embedding. Since our dataset can be interpreted as having two discrete levels of similarity – categorical similarity and instance similarity – we train the Siamese network in two phases. First, we use a large margin and label all sketches and photos from the same category as matching (this is effectively categorical supervision but *not* the typical categorical softmax loss function). Next, we use a small margin and treat only instance level sketch-photo pairs as matching. E.g. a sketch of a rabbit and a photo of a rabbit would be pushed apart if the sketch was not generated from that photo. In each epoch, we train with 10 times more negative than positive pairs while maintaining a 50% ratio by repeating positive pairs accordingly. Since negative pairs are plentiful we resample new negative pairs between training epochs.

Training with Triplet ranking loss. For the triplet loss we need input tuples of the form $(S, I+, I-)$ corresponding to a sketch, a matching image, and a non-matching image. For the ranking loss, our experiments suggest it is better to sample $I+$ and $I-$ only from the sketch category (e.g. a zebra sketch with a matching zebra photo and non-matching zebra photo), because we will simultaneously use a classification loss which differentiates sketches from photos of different categories. Note that for our Triplet loss network the two image branches still share weights. As a result, both Siamese and Triplet networks will have one set of weights for the sketch domain and one set of weights for the image domain.

We train networks using Caffe [Jia et al. 2014]. Appendix A describes additional training details. Training parameters such as learning rate decay and batch size can be found in the supplemental material. Figure 6 visualizes the embedding learned by this final Triplet network. The non-linear projection from 1024D to 2D is performed using t-SNE [van der Maaten and Hinton 2008].

4.3 Matching Sketches and Images

Our network consists of two subnetworks responsible for mapping the input sketch and photo into a shared 1024 dimensional feature space.[3] We can precompute features for large sketch or image databases and rapidly search for matches when given a query.

5 Quantitative Evaluation

We evaluate our learned representation on two sketch-based image retrieval benchmarks – the fine-grained benchmark of Li et al. [2014] and the held out test set of the Sketchy database.

The Li et al. [Li et al. 2014] benchmark evaluates *intra-category* sketch retrieval for a known category, e.g. given a sheep sketch rank a set of sheep photos according to similarity. Retrieval results are scored based on how often the $K = 5$ top retrieved images share discrete, hand-coded attributes of viewpoint, zoom, part configuration, and body shape. Higher scores are better. We evaluate on the 10 common categories between our datasets. This benchmark assumes category-specific models but our Triplet network is trained to simultaneously embed and recognize 125 object categories. Nonetheless it slightly outperforms Li et al.'s category-specific deformable part models on average, although their method is better on three categories. Both methods outperform a spatial pyramid (SP) baseline.

Table 1: *Sketch retrieval benchmark of Li et al. [2014]*

	ours	Li et al.	SP
airplane	**27.2**	22	20.33
bicycle	**21.5**	11.67	13.83
car	15.8	**18.83**	14.5
cat	**13.8**	12.17	7.67
chair	**21.7**	20	20.33
cow	**19.8**	19.67	14
dog	**21**	9.5	6.83
horse	23.2	**31.67**	7.33
motorbike	13	**22.5**	9
sheep	**21**	17.67	5
average	**19.8**	18.57	11.88

We also evaluate several deep networks and baselines for sketch-based retrieval on our held out test set of 6312 query sketches and 1250 photos spanning 125 categories. For each sketch query there

[3] The networks also output 125 dimensions corresponding to classification confidences, but we discard these for retrieval applications.

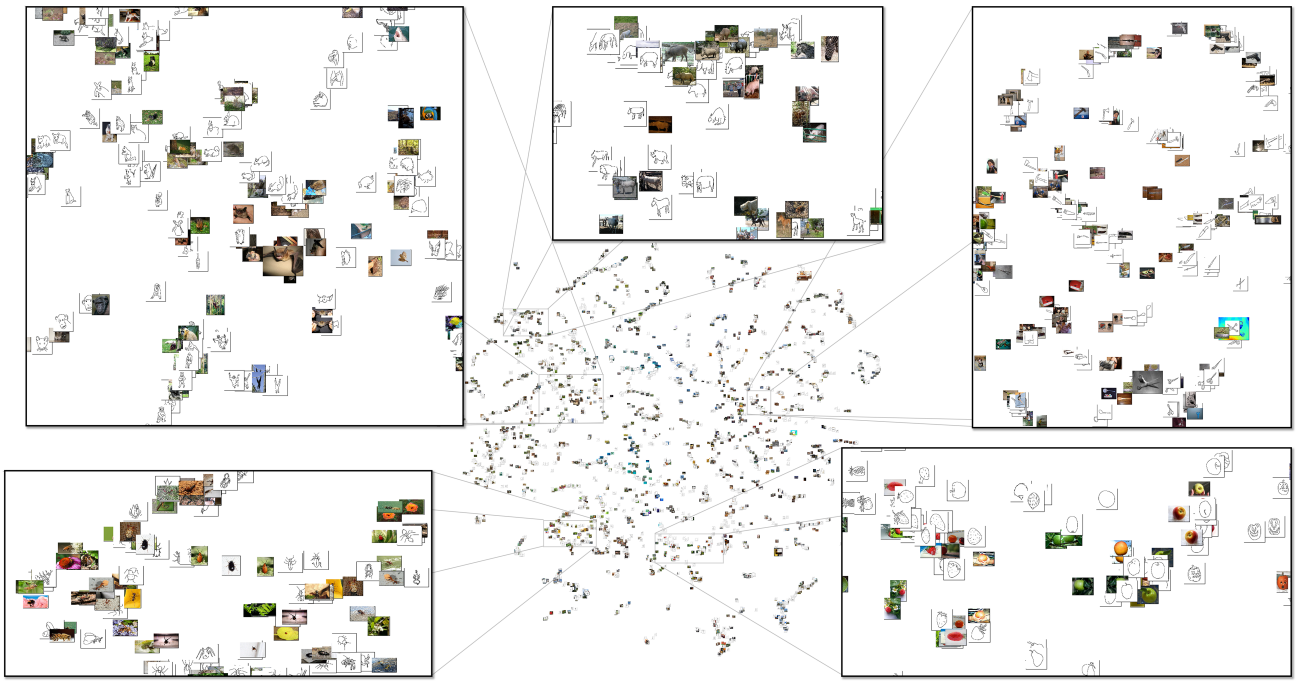

Figure 6: *t-SNE visualization of the Sketchy database test set of 1250 images and sketches. The images and sketches are embedded into a common feature space with our best performing Triplet network. Note that nearby sketches and photos share not just the same category but also similar shape and pose. Categories also seem to have arranged themselves by semantic similarity even though no such supervision was provided – insects such as bees, scorpions, ants are grouped (lower left), as are small animals (top left), grazing animals (top center), hand-held devices (top right), and fruits (bottom right).*

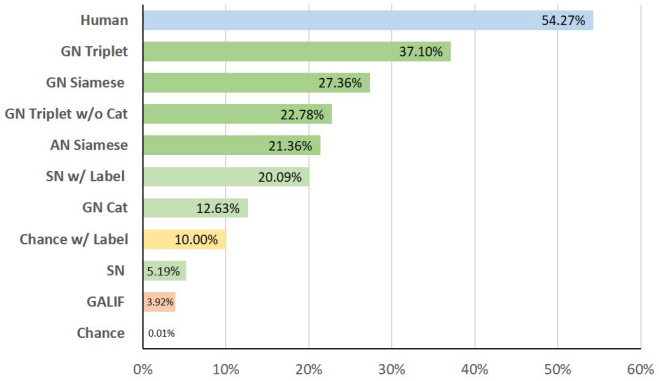

Figure 7: *Average recall at $K = 1$ on the Sketchy database test set. This is the percentage of time that the highest ranking (smallest distance) retrieved image is the single matching photo among 1250 test images. See main text for explanation of algorithm variants. Green bars correspond to methods and features proposed in this paper.*

is a single matching photo (the photo that prompted the creation of that sketch), and we measure performance by recall @ K. For a particular sketch query, recall @ K is 1 if the corresponding photo is within the top K retrieved results and 0 otherwise. We average over all queries to produce Figures 7 and 8. Figure 7 focuses on the most challenging $K = 1$ case. It shows how often the top retrieval result is the single matching photo in the 1250 photo test set. Figure 8 plots recall @ K for $K = 1$ to 10.

We estimate human performance at $K = 1$ by having participants manually find the matching photo for a query sketch from our test set. To ease the task we sort the photos into categories so that participants need not browse 1,250 photos, but participants are not told the ground truth category. After 772 trials, the human participants select the correct photograph 54% of the time. This is not an "upper bound", though – there was large variance in accuracy between participants with some achieving greater than 70% $K = 1$ recall. We include this human baseline in Figure 7.

We compare the following retrieval methods:

GN Triplet. This is our top-performing model, GoogLeNet trained with Triplet and classification loss.

GN Siamese. GoogLeNet trained with Siamese and classification loss.

GN Triplet w/o Cat. GoogLeNet trained exclusively with the Triplet loss.

AN Siamese. AlexNet trained with Siamese and classification loss.

GN Cat. GoogLeNet trained exclusively with the classification loss on the Sketchy database. The sketch and photo networks are trained independently, but as they predict the same 125 categories their final layer outputs are comparable. GN Cat is meant to represent a "retrieval by categorization" method.

Chance. Photos are retrieved in random order.

Chance w/ Label. This represents a hypothetical algorithm which can achieve perfect category recognition but ranks the results within that category randomly. This is a strong baseline for our test set – with 125 categories and 10 images per category, this baseline will

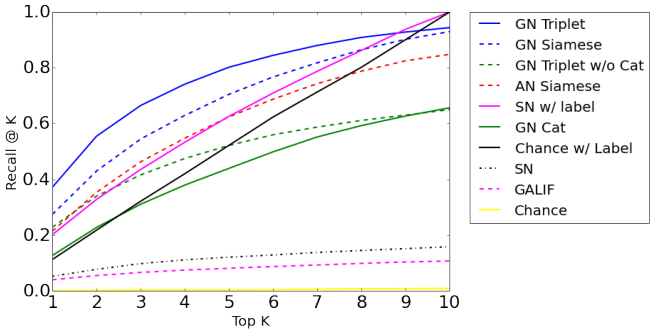

Figure 8: *Evaluation on our test set. We measure recall at K – whether or not the network can retrieve target image within Top K nearest neighbors. See main text for explanation of algorithm variants.*

always find the matching photo within the top 10 out of 1250 retrieved photos.

GALIF. Gabor local line based feature [Eitz et al. 2012b]. This feature describes sketches using a bank of Gabor filters. It was used for sketch-based shape retrieval by rendering 3D objects as sketches. We apply it to sketch-based image retrieval by performing Canny edge detection on the photos before computing the feature.

SN. For another "retrieval by categorization" baseline, we fine-tune GoogLeNet with the 250 category Eitz 2012 dataset and then use the network as a feature extractor. Similar to GALIF approach, we first use edge detection on photos then extract features from both sketch and edge images. We use the 1024 dimensional penultimate network layer activations as the feature.

SN w/ Label. Same as SN, except that we assume category recognition is perfect as in the Chance w/ Label baseline. The SN representation outperforms that baseline because it can still sort results within class. Even though this method has an oracle telling it ground truth sketch and photo category, it still performs worse than networks trained on fine-grained sketch-photo pairs for small values of K.

Our benchmark leads us to conclude that the deeper GoogLeNet significantly outperforms AlexNet for our task. Likewise, Triplet loss significantly outperforms Siamese loss. Perhaps most surprising is the effect of combining classification loss with Triplet or Siamese losses – incorporating classification loss significantly improves fine-grained sketch retrieval even though it is the least successful loss function on its own. Classification loss alone leads to $K = 1$ recall of 12.6%, and Triplet loss alone leads to $K = 1$ recall of 22.8% but together the recall improves to 37.1%. This means that one third of the time the single correct match out of 1250 test photos is the first retrieval result. The correct match is within the top 8 results 90% of the time.

Figure 10 shows examples of sketch-based image retrieval. For each sketch query we show the top 10 retrieval results (smallest distance in the Triplet GoogLeNet feature space). We search a collection of 255,828 Flickr images. 125,000 of these photos come from downloading 1,000 Flickr photos with tag searches based on the 125 Sketchy database categories. 130,828 more images are downloaded randomly. The query sketches come from the Eitz 2012 database. The retrieved photos generally depict objects of the desired type in similar poses.

Evaluation of pre-training. We quantify the effect of pre-training by incrementally adding pre-training with ImageNet, Eitz 2012, and Flickr. All experiments conclude with 160 thousand iterations of training on the Sketchy database with triplet and classification loss and the GoogLeNet architecture. No pre-training (random initial weights) leads to recall at $K = 1$ of 2%. Pre-training with ImageNet leads to 28%. ImageNet and Eitz 2012 leads to 30%. ImageNet and Flickr leads to 33%. Using all three for pre-training leads to 36% recall at $K = 1$. This is slightly less than our best reported model because of simplified training during these experiments.

Training on MS COCO object boundaries. We claim that human object sketches are not particularly similar to object boundaries or silhouettes. To test this, we train a network using object boundaries from MS COCO [Lin et al. 2014] rendered like our sketches. We omit MS COCO instances that are partially occluded by other objects or truncated by the image boundary. We train and evaluate on the 13 object categories that overlap between the Sketchy and COCO databases and have at least 500 object instances. The MS COCO "sketches" are centered and scaled in the same manner as the Sketchy database and used to train networks with only ImageNet pre-training. For these 13 categories, the network trained on the Sketchy database performs significantly better – 35% vs 11% $K = 1$ recall. This suggests that faithful object boundaries are not a good proxy for human-drawn object sketches.

6 Visualization: Average Objects

If our learned representation allows us to retrieve semantically similar objects with similar poses it could enable an object synthesis or data exploration tool in the spirit of AverageExplorer [Zhu et al. 2014]. AverageExplorer operates on constrained image sets, e.g. image search results for "kids with Santa" or "a brown tabby cat" but we aim to create average images from a diverse collection of 255,828 Flickr images with no user intervention beyond the initial query sketch.

The primary challenge is that our retrieval results (Figure 10) contain objects at different scales and locations, even when the pose and object attributes are fairly similar. This means that a naive average of the top retrieved photos is very blurry. In Section 3 we speculated that our sketch-photo training pairs implicitly encode information about which image structures are salient. We use the approach of Zeiler and Fergus [2014] to localize "salient" regions as learned by our deep network. The approach is conceptually simple – we grey out image regions and if that leads to a large change in the final network layer activations then we must have removed a salient structure. Figure 9a shows that this does indeed tend to localize objects in photos. Figure 9b shows an average image computed by centering and rescaling top retrieval results such that the salient regions overlap. Such averages are sharper than unaligned averages but still blurry.

To improve the averages further we use FlowWeb [Zhou et al. 2015] to find correspondences among the retrieved photos *and* the query sketch. Aligning the sketch to the detected edges of the photos works slightly better than using the photos directly. Figure 9b shows averages computed after the retrieved photos have been transformed according to FlowWeb correspondences. The FlowWeb algorithm was generally unable to find correspondences if we skipped the salience-based alignment and centering. We think that these average images demonstrate the ability of our learned representation to find cross-domain matches and hints at the possibility of new sketch-based image synthesis techniques.

7 Limitations and Future Opportunities

Limitations. Our data collection and training assumes that there are three discrete levels of similarity between sketches and photos –

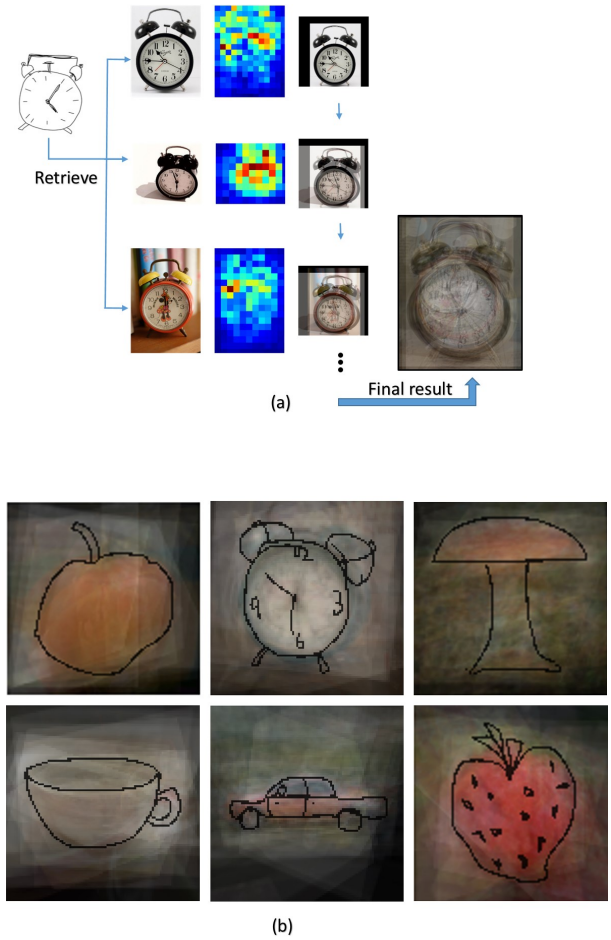

Figure 9: *In (a) we align top retrieval results such that the salient regions overlap and this produces a cleaner average image. (b) shows examples of average images from our top 20 results. Fine-grained alignment with the overlaid sketch is based on corresponding points from Flowweb [Zhou et al. 2015].*

the same instance, the same category, or completely dissimilar. But sketch-photo similarity as perceived by a human would of course be more continuous. It is not economical to collect all pairs of perceptual distances between 75,471 sketches and 12,500 photos and it doesn't appear necessary to learn a useful representation. Still, it would probably help to have more annotations, especially within categories where it seems unsatisfactory to assume that there is only one valid match for any sketch query – there could be several photos that match a sketch reasonably well. The Triplet ranking loss naturally accommodates such finer-grained supervision. This limitation is addressed by "Sketch Me that Shoe" [Yu et al. 2016], but their database is considerably smaller.

Possible applications of the Sketchy Database

Previous portrait synthesis methods [Berger et al. 2013; Limpaecher et al. 2013] used sketch-photo pairs to aid the generative photo to sketch process. Our database could be used as training data for generating sketches of photographic objects. In the other direction, previous methods synthesized images from sketches [Chen et al. 2009; Eitz et al. 2011b] and the Sketchy database could be used to improve the sketch-based retrieval at the core of these methods or to directly learn deep generative image models [Dosovitskiy et al. 2014]. The database could also enable the study of the humans artists analogous to "Where do people draw lines" [Cole et al. 2008] which studied drawings of 3D shapes. Finally, while we were motivated to collect the Sketchy database to provide training data for deep networks, it can also serve as an image-retrieval benchmark for a field that has a shortage of large scale benchmarks.

Acknowledgments

The "Badly Drawn Bunnies" subtitle was inspired by an article by Rebecca Boyle discussing "How Do Humans Sketch Objects?" [Eitz et al. 2012a]. Patsorn Sangkloy is supported by a Royal Thai Government Scholarship. James Hays is supported by NSF CAREER Award 1149853.

References

ANTOL, S., ZITNICK, C. L., AND PARIKH, D. 2014. Zero-Shot Learning via Visual Abstraction. In *ECCV*.

BANSAL, A., KOWDLE, A., PARIKH, D., GALLAGHER, A., AND ZITNICK, L. 2013. Which edges matter? In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, 578–585.

BELL, S., AND BALA, K. 2015. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph. 34*, 4 (July).

BERGER, I., SHAMIR, A., MAHLER, M., CARTER, E., AND HODGINS, J. 2013. Style and abstraction in portrait sketching. *ACM Trans. Graph. 32*, 4 (July), 55:1–55:12.

BRADY, T. F., KONKLE, T., ALVAREZ, G. A., AND OLIVA, A. 2008. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences 105*, 38, 14325–14329.

BRADY, T. F., KONKLE, T., GILL, J., OLIVA, A., AND ALVAREZ, G. A. 2013. Visual long-term memory has the same limit on fidelity as visual working memory. *Psychological Science 24*, 6.

CAO, Y., WANG, C., ZHANG, L., AND ZHANG, L. 2011. Edgel index for large-scale sketch-based image search. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 761–768.

CAO, X., ZHANG, H., LIU, S., GUO, X., AND LIN, L. 2013. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 313–320.

CHEN, T., MING CHENG, M., TAN, P., SHAMIR, A., AND MIN HU, S. 2009. Sketch2photo: internet image montage. *ACM SIGGRAPH Asia*.

CHEN, T., TAN, P., MA, L.-Q., CHENG, M.-M., SHAMIR, A., AND HU, S.-M. 2013. Poseshop: Human image database construction and personalized content synthesis. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (May), 824–837.

CHOPRA, S., HADSELL, R., AND LECUN, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 539–546.

COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ,

S. 2008. Where do people draw lines? *ACM Transactions on Graphics (Proc. SIGGRAPH) 27*, 3 (Aug.).

DEL BIMBO, A., AND PALA, P. 1997. Visual image retrieval by elastic matching of user sketches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 19*, 2 (Feb), 121–132.

DOSOVITSKIY, A., SPRINGENBERG, J. T., AND BROX, T. 2014. Learning to generate chairs with convolutional neural networks. *CoRR abs/1411.5928*.

EITZ, M., HILDEBRAND, K., BOUBEKEUR, T., AND ALEXA, M. 2010. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics 34*, 5, 482–498.

EITZ, M., HILDEBRAND, K., BOUBEKEUR, T., AND ALEXA, M. 2011. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Transactions on Visualization and Computer Graphics 17*, 11, 1624–1636.

EITZ, M., RICHTER, R., HILDEBRAND, K., BOUBEKEUR, T., AND ALEXA, M. 2011. Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications*.

EITZ, M., HAYS, J., AND ALEXA, M. 2012. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4, 44:1–44:10.

EITZ, M., RICHTER, R., BOUBEKEUR, T., HILDEBRAND, K., AND ALEXA, M. 2012. Sketch-based shape retrieval. *ACM Transactions on Graphics (Proceedings SIGGRAPH) 31*, 4, 31:1–31:10.

EVERINGHAM, M., GOOL, L., WILLIAMS, C. K., WINN, J., AND ZISSERMAN, A. 2010. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision 88*, 2 (June), 303–338.

FELZENSZWALB, P. F., GIRSHICK, R. B., MCALLESTER, D., AND RAMANAN, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 9 (Sept.), 1627–1645.

GRILL-SPECTOR, K., AND KANWISHER, N. 2005. Visual recognition: as soon as you see it, you know what it is. *Psychological Science 16*, 2, 152–160.

HADSELL, R., CHOPRA, S., AND LECUN, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 1735–1742.

HAN, X., LEUNG, T., JIA, Y., SUKTHANKAR, R., AND BERG, A. 2015. Matchnet: Unifying feature and metric learning for patch-based matching. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 3279–3286.

HU, R., AND COLLOMOSSE, J. 2013. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding 117*, 7, 790–806.

JACOBS, C. E., FINKELSTEIN, A., AND SALESIN, D. H. 1995. Fast multiresolution image querying. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, ACM, SIGGRAPH '95, 277–286.

JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., AND DARRELL, T. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

JUN, X., AARON, H., WILMOT, L., AND HOLGER, W. 2014. Portraitsketch: Face sketching assistance for novices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ACM.

KATO, T., KURITA, T., OTSU, N., AND HIRATA, K. 1992. A sketch retrieval method for full color image database-query by visual example. In *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, 530–533.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *26th Annual Conference on Neural Information Processing Systems (NIPS)*, 1106–1114.

LEE, D., AND CHUN, M. M. What are the units of visual short-term memory, objects or spatial locations? *Perception & Psychophysics 63*, 2, 253–257.

LI, Y., HOSPEDALES, T. M., SONG, Y.-Z., AND GONG, S. 2014. Fine-grained sketch-based image retrieval by matching deformable part models. In *British Machine Vision Conference (BMVC)*.

LI, Y., SU, H., QI, C. R., FISH, N., COHEN-OR, D., AND GUIBAS, L. J. 2015. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph. 34*, 6 (Oct.), 234:1–234:12.

LIMPAECHER, A., FELTMAN, N., TREUILLE, A., AND COHEN, M. 2013. Real-time drawing assistance through crowdsourcing. *ACM Trans. Graph. 32*, 4 (July), 54:1–54:8.

LIN, T., MAIRE, M., BELONGIE, S. J., BOURDEV, L. D., GIRSHICK, R. B., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P., AND ZITNICK, C. L. 2014. Microsoft COCO: common objects in context. *CoRR abs/1405.0312*.

LIN, T.-Y., CUI, Y., BELONGIE, S., AND HAYS, J. 2015. Learning deep representations for ground-to-aerial geolocalization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

MAINELLI, T., CHAU, M., REITH, R., AND SHIRER, M., 2015. Idc worldwide quarterly smart connected device tracker. http://www.idc.com/getdoc.jsp?containerId=prUS25500515, March 20, 2015.

MARTIN, D., FOWLKES, C., TAL, D., AND MALIK, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 416–423.

NIEUWENSTEIN, M., AND WYBLE, B. 2014. Beyond a mask and against the bottleneck: Retroactive dual-task interference during working memory consolidation of a masked visual target. *Journal of Experimental Psychology: General 143*, 1409–1427.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., BERG, A. C., AND FEI-FEI, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3, 211–252.

SAAVEDRA, J. M., AND BARRIOS, J. M. 2015. Sketch based image retrieval using learned keyshapes (lks). In *Proceedings of the British Machine Vision Conference (BMVC)*, 164.1–164.11.

SCHNEIDER, R. G., AND TUYTELAARS, T. 2014. Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graph. 33*, 6 (Nov.), 174:1–174:9.

SCLAROFF, S. 1997. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition 30*, 4, 627 – 641.

SHRIVASTAVA, A., MALISIEWICZ, T., GUPTA, A., AND EFROS, A. A. 2011. Data-driven visual similarity for cross-domain image matching. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 154.

SMEULDERS, A., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 12 (Dec), 1349–1380.

SU, H., MAJI, S., KALOGERAKIS, E., AND LEARNED-MILLER, E. G. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2014. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

TAIGMAN, Y., YANG, M., RANZATO, M., AND WOLF, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 1701–1708.

VAN DER MAATEN, L., AND HINTON, G. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research 9*, 3 (Nov.), 2579–2605.

WANG, J., SONG, Y., LEUNG, T., ROSENBERG, C., WANG, J., PHILBIN, J., CHEN, B., AND WU, Y. 2014. Learning fine-grained image similarity with deep ranking. *CoRR abs/1404.4661*.

WANG, F., KANG, L., , AND LI, Y. 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

XIAO, J., EHINGER, K. A., HAYS, J., TORRALBA, A., AND OLIVA, A. 2014. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 1–20.

YU, Q., YANG, Y., SONG, Y.-Z., XIANG, T., AND HOSPEDALES, T. 2015. Sketch-a-net that beats humans. In *British Machine Vision Conference (BMVC)*.

YU, Q., LIU, F., SONG, Y., XIANG, T., HOSPEDALES, T., AND LOY, C. C. 2016. Sketch me that shoe. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

ZEILER, M. D., AND FERGUS, R. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*. Springer, 818–833.

ZHOU, T., JAE LEE, Y., YU, S. X., AND EFROS, A. A. 2015. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

ZHU, J.-Y., LEE, Y. J., AND EFROS, A. A. 2014. Averageexplorer: Interactive exploration and alignment of visual data collections. *ACM Transactions on Graphics (SIGGRAPH 2014) 33*, 4.

A Miscellaneous training details

Undoing mirror invariance. The original ImageNet training and the sketch and photo subnetwork pre-training have the unfortunate side effect of making the networks largely mirror invariant. This is to be expected when using a classification loss – mirroring should not affect the predicted category. Although mirror invariance may be desirable for some sketch-based retrieval applications such as product search, we want more pose sensitivity. Building positive training pairs out of similarly oriented sketch-photo pairs was not enough to overcome the pre-training. By using sketch-photo pairs where only one of the domains has been mirrored as *negative* training pairs we are able to make the networks "forget" mirror invariance while maintaining high classification accuracy.

Closing the Triplet cross-domain gap. Unlike the Siamese network, the loss function for the Triplet network only requires positive pairs to be *closer* than negative pairs, but not necessarily *close* in an absolute sense. Since each subnetwork has its own weights, there is no guarantee that the feature for sketch and image would lie in the same space as long as the right match is at the closest distance among all other matches from the same domain. To encourage the network to close this gap, we further fine-tune the triplet network with an additional loss term based on the distance between the positive sketch and photo pair. The loss becomes: $L = c_1 * D(S, I+) + c_2 * max(0, m + D(S, I+) - D(S, I-)$ where c_1 and c_2 control the relative influence of the absolute distance loss and the Triplet loss. It is possible to avoid this complication by using only the Siamese contrastive loss, but the Triplet network performs better on our sketch-based image retrieval benchmark.

Feature Dimension. In our Triplet GoogLeNet experiments, we fix the output dimension to 1024 when comparing features. However, we can reduce the output dimensionality by adding an additional fully connected layer with fewer hidden units before the loss layer. We can compute 64-dimensional features with little decrease in retrieval accuracy. See the supplemental material for more details.

Figure 10: *Qualitative retrieval results. Queries are sampled from the Sketchy database test set. Matching photos are retrieved from a held out collection of 450 thousand Flickr photos, of which 50k are retrieved by querying categories in the Sketchy database and 400k are random. There is no ground truth, but clearly incorrect retrieval results are highlighted in red.*