Fine-Grained Retrieval with Autoencoders

Tiziano Portenier[1], Qiyang Hu[1], Paolo Favaro[1] and Matthias Zwicker[2]
[1]*University of Bern, Bern, Switzerland*
[2]*University of Maryland, College Park, U.S.A.*

Keywords: Image Retrieval, Deep Learning, Autoencoders.

Abstract: In this paper we develop a representation for fine-grained retrieval. Given a query, we want to retrieve data items of the same class, and, in addition, rank these items according to intra-class similarity. In our training data we assume partial knowledge: class labels are available, but the intra-class attributes are not. To compensate for this knowledge gap we propose using an autoencoder, which can be trained to produce features both with and without labels. Our main hypothesis is that network architectures that incorporate an autoencoder can learn features that meaningfully cluster data based on the intra-class variability. We propose and compare different architectures to construct our features, including a Siamese autoencoder (SAE), a classifying autoencoder (CAE) and a separate classifier-autoencoder (SCA). We find that these architectures indeed improve fine-grained retrieval compared to features trained purely in a supervised fashion for classification. We perform experiments on four datasets, and observe that the SCA generally outperforms the other two. In particular, we obtain state of the art performance on fine-grained sketch retrieval.

1 INTRODUCTION

In example-based retrieval, given a query represented by an exemplar the goal is to return data items that are as similar to the exemplar as possible, usually in a list ordered by similarity to the exemplar. Similarity between exemplar and query results may be judged by whether they belong to the same object class, or by more fine-grained properties, such as whether query and result show the same instance of a class (instance-level retrieval), or by similarity in pose, color, or style of objects in images. Thanks to the availability of a large amount of labeled data, neural networks for classification and class-based retrieval can be trained very successfully in a supervised fashion (He et al., 2016). The resulting features, however, do not naturally support fine-grained retrieval, because class-based training leads to invariance against fine-grained properties such as object pose etc. To suppress invariance, one could enrich the training data with labels for fine-grained properties (such as instance-level labels), and train again in a supervised manner (Radenović et al., 2016; Gordo et al., 2016). However, this requires much more effort to prepare suitable labeled training data.

To avoid cumbersome labeling of training data, one could leverage representations obtained using autoencoder networks for fine-grained retrieval. By construction, features extracted from autoencoders retain as much information as possible about the data, including fine-grained properties. Therefore, it seems attractive to exploit such representations for fine-grained retrieval. When trained in an unsupervised manner, however, autoencoder features suffer from entanglement, which means that information about class membership and fine-grained properties may be encoded in overlapping feature dimensions. Therefore, we cannot expect good class-based retrieval performance when using autoencoder features.

Our goal in this paper is to develop a representation that supports fine-grained retrieval, but does not require supervised learning with fine-grained labels. Given a query, we want to retrieve data items of the same class, and in addition, rank the query results according to intra-class similarity (for example object pose). Our assumption is that we have training data containing class labels, but intra-class variability is not labeled and needs to be taken into account using unsupervised learning. Our main hypothesis is that we can achieve fine-grained retrieval by leveraging an autoencoder, which should learn to capture intra-class variability in an unsupervised manner.

We explore different ways to combine class-based supervised learning and an autoencoder to construct

Portenier, T., Hu, Q., Favaro, P. and Zwicker, M.
Fine-Grained Retrieval with Autoencoders.
DOI: 10.5220/0006602100850095
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 85-95
ISBN: 978-989-758-290-5

our features, and show that indeed this can improve fine-grained retrieval. As a key contribution of this paper, we propose and compare three main architectures: (1) a *siamese autoencoder (SAE)* that learns a representation that respects class membership using a contrastive loss; (2) a *classifying autoencoder (CAE)* that jointly minimizes the sum of an autoencoding and a classification loss using a single network; (3) a *separate classifier-autoencoder (SCA)* based on separate training of a supervised classification and an unsupervised autoencoding network, followed by concatenation of the respective feature vectors. In our experiments, we find that all three architectures indeed improve fine grained retrieval compared to features trained purely in a supervised fashion for classification. In addition, the SCA outperforms the other two (SAE and CAE). Intuitively, this is because the SCA leads to better disentanglement, separating invariant class properties from fine-grained variability. We exploit this to construct a feature similarity metric that is highly effective for fine-grained retrieval.

We evaluate and compare our approaches using four different datasets, MNIST hand-written digits (Lecun et al., 1998), Google street view house-numbers (Netzer et al., 2011), images rendered from ShapeNet (Chang et al., 2015), and sketches from the Sketchy database (Sangkloy et al., 2016). In general, we show that concatenating supervised and unsupervised features that were trained separately outperforms the other techniques that we explored.

In particular, we demonstrate that for sketch-based sketch retrieval we significantly outperform the state of the art on fine-grained retrieval in the Sketchy database (Sangkloy et al., 2016). This is remarkable because the previous state of the art used fine-grained labels in a supervised training setup (Sangkloy et al., 2016). In contrast, we do not use fine-grained labels for training, yet obtain improved fine-grained retrieval thanks to the inclusion of unsupervised autoencoder features in our representation.

2 RELATED WORK

Image Retrieval. While our main application in this paper is focusing on sketch data, our problem is related to image retrieval in general. Traditionally, descriptors for image retrieval are based on local features, for example by aggregating local gradient-based features and by building bag-of-visual-words (BOV), Fisher kernel (Perronnin et al., 2010), or VLAD (vector of locally aggregated descriptors) (Jgou et al., 2010) representations. More recently, the success of deep convolutional neural networks for image classification has inspired holistic image representations based on these techniques targeted at image retrieval. Babenko et al. (Babenko et al., 2014) were among the first to leverage activations in convolutional neural networks (CNNs) as features for image retrieval, demonstrating competitive results compared with traditional hand-crafted features. Paulin et al. learn patch-based features using CNNs and aggregate them using VLAD for image retrieval (Paulin et al., 2015).

Babenko et al. (Babenko and Lempitsky, 2015) make the interesting observation that a global descriptor constructed from local CNN features by sum pooling aggregation, without high-dimensional embedding, outperforms aggregation using more sophisticated techniques such as Fisher vectors and VLAD. Similarly, Tolias et al. (Tolias et al., 2016) propose to build a feature based on a regional maximum activation of convolutions (R-MAC). They show that their representation is significantly more suitable for fine-grained retrieval tasks, such as particular object retrieval, compared to previous work based on CNN features (Babenko et al., 2014; Babenko and Lempitsky, 2015). They also develop a re-ranking approach using approximate object localization and query expansion, and they show that with these additional steps, their technique also outperforms the previous state of the art based on hand crafted features (Mikulik et al., 2013) on standard benchmarks (Philbin et al., 2007; Philbin et al., 2008).

Instead of relying on hand crafted aggregation strategies, like in R-MAC (Tolias et al., 2016), it seems attractive to learn parameters of feature aggregation in an end-to-end manner. Arandjelovic et al. (Arandjelovic et al., 2016) propose a network architecture that includes an aggregation layer inspired by VLAD (Jgou et al., 2010) that can be trained by backpropagation. They report state of the art results on place recognition benchmarks. Gordo et al. (Gordo et al., 2016) build on R-MAC, but include a region proposal network that is trained in an end to end manner, instead of using a fixed grid of regions. Radenovic et al. (Radenović et al., 2016) use a representation also based on maximum activation of convolutions (MAC), but instead of regionally aggregating, they propose to fine tune the network using hard positive and hard negative examples. Both Gordo and Radenovic et al. achieve excellent results, although Gordo (Gordo et al., 2016) reports the highest mean average precision scores on standard benchmarks for image retrieval.

Sketch Retrieval. A main difference between these image retrieval techniques and our approach is that

our goal is to construct a feature representation suitable for fine grained retrieval in a partly unsupervised manner, that is, when fine-grained labels are not available. In addition, our main application is sketch retrieval, rather than image retrieval. Deep learning has been used for sketch classification (Yu et al., 2016b) or retrieval (Su et al., 2015), outperforming classical techniques based on bags-of-visual-words (Eitz et al., 2012) or Fisher vectors (Schneider and Tuytelaars, 2014) by a large margin. One could leverage features extracted from classification networks also for fine-grained retrieval, but we show that our approach outperforms this strategy.

Recently, Sangkloy et al. (Sangkloy et al., 2016) constructed a database (called Sketchy) with sketch-photo pairs that provide fine-grained instance level labels. They leverage this data to learn a joint embedding for sketches and images using a triplet loss. Their approach is very similar to concurrent work by Yu et al. (Yu et al., 2016a), who collected a similar database and also learned a joint embedding with CNNs and a triplet loss. Both Sankloy and Yu et al. show state of the art results for instance-level sketch-based image retrieval and sketch-based sketch retrieval. In contrast, we are not using instance-level labels for training, and we focus on sketch retrieval. We use the fine-grained labels in the Sketchy database only to evaluate the fine-grained retrieval performance, and we show that our approach improves fine-grained sketch retrieval, even though we do not use fine-grained labels for training.

3 NETWORK ARCHITECTURES FOR FINE-GRAINED RETRIEVAL USING AUTOENCODERS

In this section we propose our approach to learn features that supports fine-grained example-based retrieval. Our assumption is that the training data contains semantic object category labels, but no fine-grained labels that encode intra-class variability such as object pose, style, or color. To learn a representation that includes both class-level semantics and fine-grained properties, we propose three network architectures as shown in Figure 1, and all three leverage autoencoders to capture intra-class variability in a partly unsupervised manner. We are calling these architectures Siamese autoencoders (SAE), classifying autoencoders (CAE), and separate classifier-autoencoders (SCA). The motivation behind choosing these three architectures is as follows: comparing the SAE and CAE allows us to evaluate the suitability of contrastive versus classification loss, and the CAE explores separate versus joint training (as in SAE and CAE) of the classifier and the autoencoder. We report on our evaluation in Section 4.

3.1 Siamese Autoencoder (SAE)

Siamese networks with contrastive loss functions have been widely used to learn representations that support classification, retrieval, and cross-domain embeddings in a common feature space (see Wang et al. (Wang et al., 2015) for an example). If the loss is purely driven by class membership, however, the representations are pushed to become invariant to intra-class variability, which is not desirable for fine-grained retrieval. Therefore, we propose Siamese autoencoders (SAE), which extend the Siamese architecture with a pair of decoders as shown in Figure 1(a). Intuitively, this should force the learned representation to retain intra-class variability, while still separating different classes. To the best of our knowledge this is novel in the context of image representations, although a similar architecture has been introduced to learn speaker-specific representations for speaker recognition (Chen and Salman, 2011).

A Siamese network (Chopra et al., 2005) consists of a pair of networks with shared weights. Training is performed by feeding triplets $(x_i, x_j, l_{i,j})$ that contain a pair of images (x_i, x_j) and a binary label $l_{i,j} \in \{0,1\}$ that is zero if x_i and x_j have the same class label and one otherwise. Siamese networks can be trained by minimizing a contrastive loss,

$$\begin{aligned}\mathcal{L}_{\text{con}}(x_i, x_j, l_{i,j}) =& (1 - l_{i,j}) d(z_i, z_j) \\ &+ l_{i,j} \max(0, m - d(z_i, z_j)),\end{aligned} \tag{1}$$

where $d(x,y)$ is the Euclidean distance between x and y, z is the network output for image x, and m is a user-defined margin. The network acts as an encoder E that produces a latent representation $z = E(x)$ of x. In this representation, input images from the same class are pulled together, and images from different classes are pushed further apart than the margin m.

Our Siamese autoencoder adds a pair of Siamese decoders at the end of the network, see Figure 1(a), to capture fine-grained properties. Each decoder D tries to reconstruct the original image from the latent representation z such that $D(z)$ is as similar to the input x as possible, forcing the representation to retain intra-class variability. The autoencoder can be trained by minimizing a reconstruction loss, for example L_2,

$$\mathcal{L}_{\text{rec}}(x) = ||x - \tilde{x}||_2^2, \tag{2}$$

where $\tilde{x} = D(E(x))$. Hence, we train the SAE using the following loss function consisting of a weighted

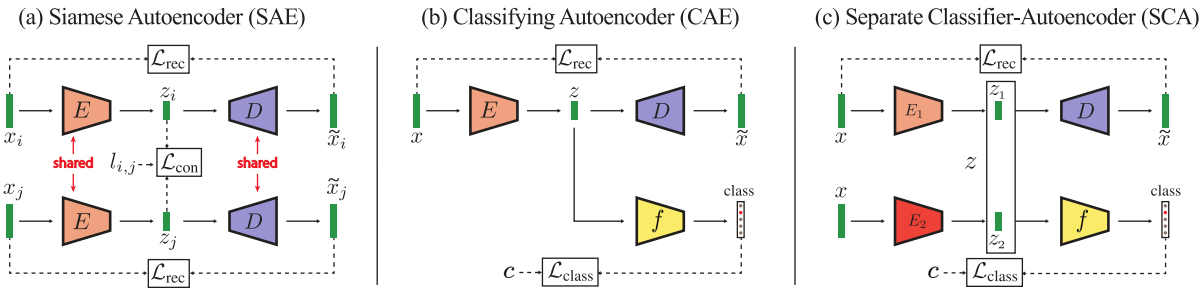

Figure 1: We propose and evaluate three architectures for fine-grained retrieval in a partly unsupervised manner (no fine-grained labels) by leveraging autoencoders: a Siamese autoencoder (SAE), classifying autoencoder (CAE), and separate classifier-autoencoder (SCA).

sum of contrastive and reconstruction loss,

$$\mathcal{L}_{\text{sae}}(x_i,x_j,l_{i,j}) = \gamma\mathcal{L}_{\text{con}}(x_i,x_j,l_{i,j}) + (1-\gamma)\frac{\mathcal{L}_{\text{rec}}(x_i)+\mathcal{L}_{\text{rec}}(x_j)}{2}, \quad (3)$$

where $\gamma \in [0,1]$ is used to balance contrastive vs. reconstruction loss.

3.2 Classifying Autoencoder (CAE)

We observe in practice that training using a contrastive loss can be unstable and requires very careful initialization. To alleviate these issues, Siamese networks can be regularized by including a softmax classification loss (Bui et al., 2016; Sangkloy et al., 2016). To determine whether the contrastive loss provides any advantage for fine-grained retrieval at all, we propose a classifying autoencoder (CAE) that only uses a classification network instead of a Siamese setup to learn the labeled class-level semantics. We again include a decoder $D(z)$ and reconstruction loss to capture intra-class variability, as shown in Figure 1(b). We achieve this by adding a single fully-connected layer f with linear activation function to the output of the encoder z. This maps the latent representation z to a probability distribution over the object categories $f(z)$. Note that we feed z as input to the decoder and not $f(z)$. We train the CAE by minimizing the following loss function,

$$\mathcal{L}_{\text{cae}}(x,c) = \gamma\mathcal{L}_{\text{class}}(x,c) + (1-\gamma)\mathcal{L}_{\text{rec}}(x), \quad (4)$$

where c is the category label of image x, $\mathcal{L}_{\text{class}}$ is the traditional classification loss, i.e. cross entropy with softmax, and γ is used to balance classification vs. reconstruction loss. Note that one can easily combine SAE and CAE, and we evaluate this option as well.

3.3 Separate Classifier-Autoencoder (SCA)

A disadvantage of the two previous architectures is that they lead to detrimental competition between the classification and reconstruction losses, for example when two images belong to different classes, but exhibit similar fine-grained properties. In theory, with enough training data this issue should resolve itself. But in practice, we found that we can train the networks to be good at classification or fine-grained retrieval, but not both. Moreover, the weighting parameter γ depends on the training data and has to be fixed at training time, and finding γ such that the learned representation provides a desired trade-off is tedious.

To mitigate these issues, we propose a representation based on a separate classifier-autoencoder (SCA), which consists of two separately trained networks. Hence, the two loss functions do not directly compete. The first networks is an autoencoder that learns a representation z_1 to encode intra-class variability. The second network is trained in a supervised fashion using a classification loss to learn a representation z_2 that captures the semantics in the data. We concatenate these two vectors to form the final representation $z = (z_1, z_2)$. See Figure 1(c) for a visualization of the proposed architecture.

The SCA leads to a representation that better disentangles class information from fine-grained properties compared to SAE and CAE. Since the second part of our representation encodes only semantic information, we can design a similarity metric that enables the user to choose the tradeoff between semantic and fine-grained information during retrieval time. Using the cosine similarity as distance metric, we propose a weighted dot product for retrieval,

$$sim(z_i,z_j) = \alpha\frac{z_{i_2}\cdot z_{j_2}}{||z_{i_2}||\,||z_{j_2}||} + (1-\alpha)\frac{z_{i_1}\cdot z_{j_1}}{||z_{i_1}||\,||z_{j_1}||}, \quad (5)$$

where α is a user-defined parameter for the tradeoff between semantics and intra-class variability. One drawback of SCA is that the learned representation is prone to redundancy, since the autoencoder will encode some semantic information in z_1. Our experiments, however, show that in practice the benefits of separating the two loss functions outweigh this issue.

4 EXPERIMENTS

In this section, we present extensive evaluations of our method on different datasets. First, we quantitatively evaluate our approach on sketch images from the Sketchy database and compare to previous state of the art on this dataset. Second, we show a quantitative comparison of the proposed method to two baselines on images rendered from ShapeNet. Third, we show qualitative results on MNIST and SVHN. We use Tensorflow (Abadi et al., 2016) with the Adam optimizer (Kingma and Ba, 2014) in all our experiments.

4.1 Sketch-based Sketch Retrieval

To provide a quantitative evaluation of our proposed architectures, we consider the problem of sketch-based sketch retrieval and train using sketch images from the Sketchy database (Sangkloy et al., 2016). This dataset is attractive for our evaluation because it includes both class-level and fine-grained annotations. The data consists of sketch-photo pairs of 125 categories, collected using crowd sourcing: for each photo, participants were asked to sketch the object with a pose similar to that of the object in the photo. The database contains 12,500 photos and 75,481 sketches with a resolution of 256×256, and each photo has at least 5 associated sketches that show the same object with similar pose. This information can be leveraged to design a benchmark for fine-grained sketch retrieval: we consider a retrieved sketch relevant if it stems from the same photo as the query sketch, which implies that the result is of the same category and has the same pose as the query. We use the test set proposed by Sankloy et al. (Sangkloy et al., 2016) for evaluation (7,063 sketches) and the remaining 68,418 sketches for training.

Inspired by Sankloy et al. (Sangkloy et al., 2016), we use GoogLeNet (Szegedy et al., 2015) as encoder, initialized with weights pre-trained on ImageNet (Russakovsky et al., 2015). We use the activations of the last pooling layer, a 1024-dimensional vector, as latent representation z. The decoder consists of 11 layers: a fully-connected layer to map z to a 980-dimensional vector that is reshaped to form a $7 \times 7 \times 20$ tensor, followed by 10 layers of transposed convolutions, sometimes called deconvolutions, with 3×3 kernels. We use ReLU activations for all but the output layer, and the hyperbolic tangent activation on the output layer. The input to the network is of size 224×224 and we randomly crop and flip input sketches for data augmentation.

In our experiment, we observed that training a SAE on this type of data using contrastive loss as defined in Equation 1 leads to unstable training and the network often diverges. As proposed by Bui et al. (Bui et al., 2016), we obtained more stable training by adding a classification term to our loss, which results in a combination of our SAE and CAE architectures. The classification term is weighted 20 times lower than the contrastive term, which is enough to achieve stable training. We report results for $\gamma = 0.05$ and $\gamma = 0.0005$. Since the encoder is pre-trained on ImageNet and the decoder is trained from scratch, we start training with $\gamma = 0.0001$ and continuously increase γ during training to the final value.

In addition to SAE, we also train a CAE by minimizing the loss as defined in Equation 4. We start training with $\gamma = 0.01$ and increase γ during training to $\gamma = 0.5$. Note the different magnitude of γ compared to SAE: classification loss and reconstruction loss have comparable magnitudes on our training data, whereas the contrastive loss is about two orders of magnitude higher than the other two.

Our third approach is to train a SCA as introduced in Section 3.3. We use the same architecture as before but add an additional fully-connected layer with ReLU at the end of both encoders (in the separate classification and autoencoder branches) to reduce the dimensionality to 512. After training, we concatenate the two feature vectors to form a 1024-dimensional embedding (the same as used in our SAE and CAE architecture), and we perform retrieval using the distance metric in Equation 5 with $\alpha = 0.3$.

We compare the performance of our SAE, CAE, and SCA features to four baselines: (1) GoogLeNet trained solely on classification, (2) a fully unsupervised autoencoder, (3) the sketch branch from the Sketchy network (Sangkloy et al., 2016), and (4) R-MAC (Tolias et al., 2016). Note that (3) was trained by leveraging the fine-grained sketch-photo relations in a supervised manner via a triplet loss. In contrast, our SAE, CAE, and SCA architectures only use object categories and learn the fine-grained similarities in an unsupervised manner using the autoencoder. For R-MAC, we use the activations from the last convolutional layer of (1), which is a $7 \times 7 \times 1024$ tensor, to construct the R-MAC features. In addition to the fine-grained retrieval benchmark described above, we use the sketches from the Sketchy database to define a second, semantic only retrieval benchmark. In this benchmark, we consider retrieval results relevant if they are of the same category as the query, regardless of the pose. For all methods, we report mean average precision (mAP) on both benchmarks in Table 1. On the fine-grained benchmark, retrieval with R-MAC features (which were designed to facilitate fine-grained re-

Table 1: mAP for both fine-grained and semantic only retrieval benchmark on the Sketchy test set.

method	fine-grained	semantic only
Autoencoder	0.2370	0.0405
Classification Network	0.2334	**0.6290**
Sketchy Network (Sangkloy et al., 2016)	0.2867	0.5125
R-MAC (Tolias et al., 2016)	0.2809	0.4171
SAE $\gamma = 0.01$ (ours)	0.1313	0.3843
SAE $\gamma = 0.0005$ (ours)	0.2837	0.2396
CAE $\gamma = 0.5$ (ours)	0.4654	0.2652
SCA $\alpha = 0.3$ (ours)	**0.4946**	0.5303

trieval) increases mAP by almost 5% compared to the classification baseline that serves as input to construct the R-MAC features. This is roughly on par with Sketchy (Sangkloy et al., 2016). Our SCA feature performs best on the fine-grained benchmark, closely followed by CAE. SAE performs worse than CAE, which is surprising, since using a Siamese architecture and a triplet loss has been proposed for image retrieval (Wang et al., 2014). Yet in our experiment, we observe that the CAE architecture, which combines classification and an autoencoder, is more effective for fine-grained retrieval. As a key contribution of our work, SCA obtains an mAP score more than 20% higher than the previous state of the art (Sangkloy et al., 2016), and CAE is still 17% better, even though we do not use the fine-grained labels for training our models.

On the semantic only benchmark, the classification network (unsurprisingly) performs best, followed by SCA, which still beats Sketchy. Note that in contrast to SCA, semantic retrieval performance decreases drastically for both SAE and CAE, compared to the classification baseline. We believe this is because (1) the learned embedding is too entangled, and (2) finding an optimal γ is not feasible, since it has to be fixed at training time. Remarkably, R-MAC also decreases the semantic only retrieval performance by more than 20%, compared to the classification baseline. It seems that SAE, CAE, and R-MAC introduce a strong tradeoff between fine-grained and semantic only retrieval, which is undesirable.

Figure 2 plots the mAP for SCA for different values of α on both benchmarks. Although there is still a tradeoff between fine-grained and semantic retrieval, we can obtain good retrieval performance on both benchmarks for a wide range of α values. Intuitively, this is because SCA better enforces the separation of class-level and fine-grained information in the features. Moreover, SCA enables the user to choose the

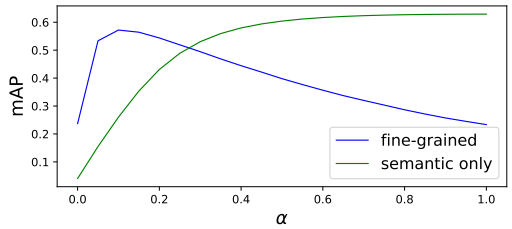

Figure 2: SCA mAP versus α for both fine-grained and semantic only retrieval on the Sketchy test set.

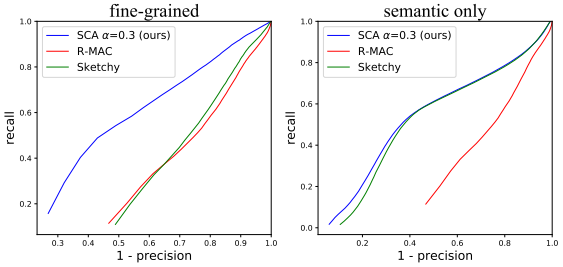

Figure 3: Precision-recall curves on the Sketchy fine-grained and semantic only retrieval benchmarks, averaged over all queries.

tradeoff by setting the parameter α at retrieval time. Figure 3 shows precision-recall curves on both benchmarks for SCA, our best performing method, in comparison to R-MAC and Sketchy. Even though SCA and Sketchy perform very similar on the semantic only benchmark in terms of mAP, the difference is significant for small recalls, which is a useful property in practice. Note that R-MAC performs miserably on the semantic only benchmark, which is surprising because the input features for R-MAC perform superior. Finally, we show some qualitative retrieval results in Figure 4. Note that our method retrieves sketches at the top that match both object category and object pose.

4.2 ShapeNet

In this experiment, we evaluate fine-grained retrieval of rendered 3D objects according to viewpoint. Given a rendered object, the goal is to retrieve images of other objects of the same class, seen from the same viewpoint. We obtained an image dataset by rendering objects from 11 ShapeNet categories: *airplane, bed, bench, bus, car, chair, guitar, piano, table, train,* and *boat*. For each object, we render diffuse RGB images of resolution 256x256 from 32 discrete viewpoints, 8 azimuth and 4 elevation angles. This results in a dataset of 902,336 images and we split the objects to form a training set of 812,103 images and a test set of 90,233 images. We train a SAE, CAE, and SCA as described in Section 3.3 using the same net-

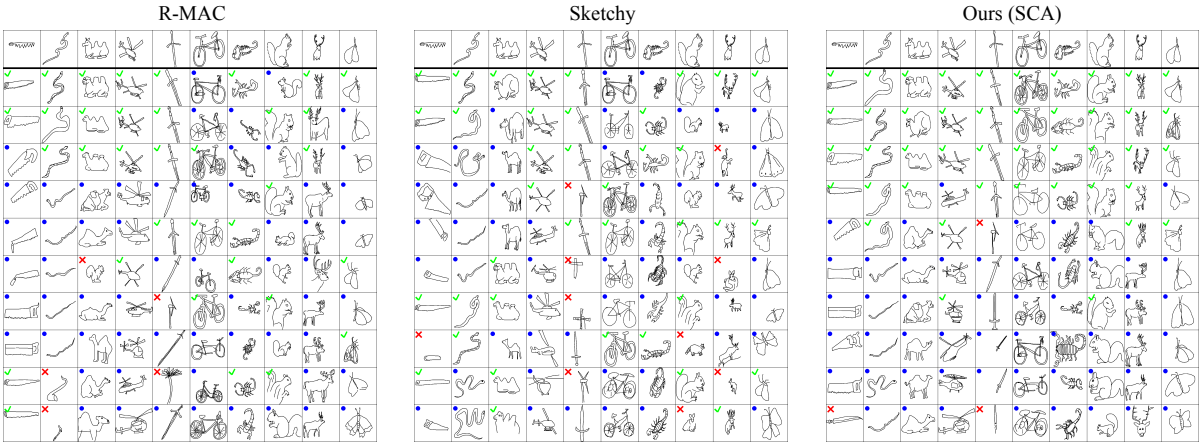

Figure 4: Qualitative retrieval results on the Sketchy test set. The query images are on the top row and we show the top-10 retrieval results beneath. Retrieval results that do not match the query class label are marked with a red cross. Blue circles mark retrieval results that match the query class label but do not stem from the same photo, thus they potentially do not match the query pose. Results that stem from the same photo as the query sketch are marked with a green tick. These are the results that we consider relevant for fine-grained retrieval, since they are guaranteed to match both query category label and pose. We use $\alpha = 0.3$ for SCA.

Table 2: mAP for both fine-grained and semantic retrieval benchmark on the ShapeNet test set.

method	fine-grained	semantic only
Autoencoder	0.1738	0.2138
Classification Network	0.1890	**0.6827**
SAE $\gamma = 0.0005$ (ours)	0.2910	0.6578
SAE $\gamma = 0.01$ (ours)	0.2854	0.6686
CAE $\gamma = 0.5$ (ours)	0.2120	**0.6827**
CAE $\gamma = 0.3$ (ours)	0.2545	0.6746
CAE $\gamma = 0.1$ (ours)	0.3081	0.6552
SCA $\alpha = 0.1$ (ours)	**0.3310**	0.6443

work architectures as in the Sketchy experiments, and the fine-grained viewpoint labels are not used for training. An autoencoder and a classification network are trained as baselines for comparison. All encoders follow a GoogLeNet architecture and are initialized with weights pre-trained on ImageNet. Unfortunately, we are not able to compare RMAC performance on this experiment. Computing RMAC on the Sketchy dataset took more than one week using the publicly available implementation, and since our ShapeNet dataset is 10 times larger, applying RMAC to this dataset is not feasible.

Our test set serves as a fine-grained retrieval benchmark, where we consider results as relevant only if they match both object category and viewpoint with the query image. In addition, we also evaluate on a semantic only retrieval benchmark by considering results relevant if they have the same object category as the query, regardless of the pose. Table 2 shows mAP for all networks on both benchmarks. The classification network and the autoencoder perform very similar on fine-grained retrieval. SAE, CAE, and SCA all increase fine-grained retrieval performance significantly, and SCA outperforms both SAE and CAE. Note that even though the classification network performs best on semantic only retrieval, SAE, CAE, and SCA performance is only insignificantly lower. We show qualitative results in Figure 5. Note that SCA retrieves objects that match both query category and viewpoint, whereas the classification network is more invariant to pose and the autoencoder often retrieves the wrong categories. Our ShapeNet benchmark contains objects that are almost rotationally symmetric, such as buses or guitars. In these cases, SCA often finds images of objects that are 180° rotated, which is reasonable for nearly symmetric objects. However, our benchmark considers these results as irrelevant (that is, wrong pose), which may explain why the mAP for fine-grained retrieval (Table 2) is lower than in the Sketchy benchmark (Table 1).

4.3 MNIST

For a qualitative evaluation of our method, we train a SCA on hand-written digits from the MNIST (LeCun and Cortes, 2010) dataset. Here we demonstrate that SCA enables retrieval of digits with similar handwriting style, where sensitivity to style is learned in an unsupervised manner.

The official training set of 60,000 images is used for training and we evaluate using the official test set of 10,000 examples. The two encoders E_1 and E_2 take 28×28 grayscale images as input and consist of four

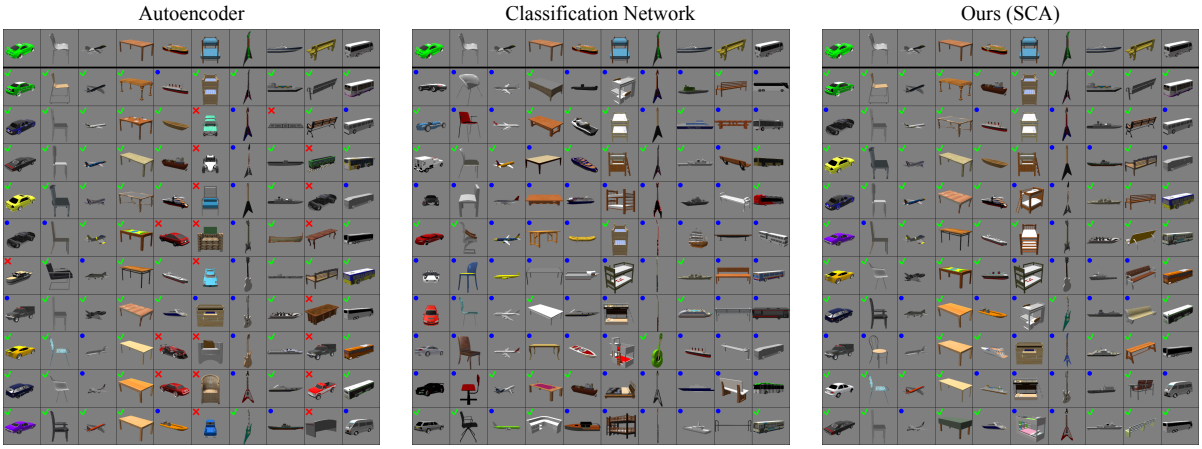

Figure 5: Qualitative retrieval results on the ShapeNet test set. The query images are on the top row and we show the top-10 retrieval results beneath. Retrieval results that do not match the query class label are marked with a red cross. Blue circles indicate retrieval results that match the query category but do not match the query pose. Results that we consider relevant for fine-grained retrieval are marked with a green tick. We use $\alpha = 0.1$ for SCA.

convolutional layers with 3×3 kernels, followed by ReLU activations. The outputs of the last convolutional layers are $4 \times 4 \times 256$ tensors that are mapped to a 16-dimensional latent representations z_1 and z_2 using a single fully-connected layer with ReLU activation. Finally, we add an additional fully-connected layer with linear activation f behind E_2. We employ batch normalization (Ioffe and Szegedy, 2015) for all convolutional layers but not for the fully-connected layers. The decoder is the reverse of the encoders: we start with a fully-connected layer to increase the dimensionality of the latent representation to 4096-dimensional and reshape to form a $4 \times 4 \times 256$ tensor. This tensor is mapped back to the input image space using four layers of transposed convolutions. We employ batch normalization and ReLU activations for all but the output layer. The output layer uses hyperbolic tangent activation.

To compare the effect of the proposed SCA on the learned embedding, we trained two baseline networks: a classification network and a fully unsupervised autoencoder. Both baselines use the same architecture as our SCA, with the exception of the latent representation being 32-dimensional.

Figure 6 shows t-SNE (Maaten and Hinton, 2008) embeddings for digits "one" and "seven" from the MNIST test set, where we plot the digits at the position of their t-SNE coordinates. The visualization shows that digits are embedded depending on writing style when training the SCA. For example, straight digits "one" are mapped to the upper right region of the cluster and more italic styles are mapped to the lower left region. This also holds for digits "seven", for example all digits featuring a cross are embedded close together (lower left inset). The autoencoder also embeds digits according to writing style, but it does not separate the different classes as well as SCA and the classification network, it actually maps italic digits and straight digits to two completely distinct clusters. In contrast, the embedding learned by the classification network is completely invariant to writing style, as shown in the insets.

Figure 7 shows some retrieval examples on the MNIST test set for all three networks. Training a classification network leads to retrieval results with the same class label as the query, but arbitrary writing style. In contrast, training an autoencoder yields retrieval results with similar writing styles but often wrong class labels. Note that the proposed SCA learns an embedding where neighboring samples are similar in both class label and writing style, which enables fine-grained retrieval without any supervision on writing styles.

4.4 SVHN

Similar to the previous experiment, we also train a SCA on the SVHN (Netzer et al., 2011) dataset. Training is performed without any data augmentation using the official training set consisting of 73,257 examples, and we evaluate our networks using the test set of 26,032 images. The SCA takes $32 \times 32 \times 3$ RGB images as input and follows the exact same architecture as for the MNIST experiments presented above.

Again, we train two baseline networks for comparison: a classification network and an autoencoder. Figure 8 shows retrieval examples on the SVHN test set for all three networks. We can observe the same behavior as for MNIST: the embedding learned by the

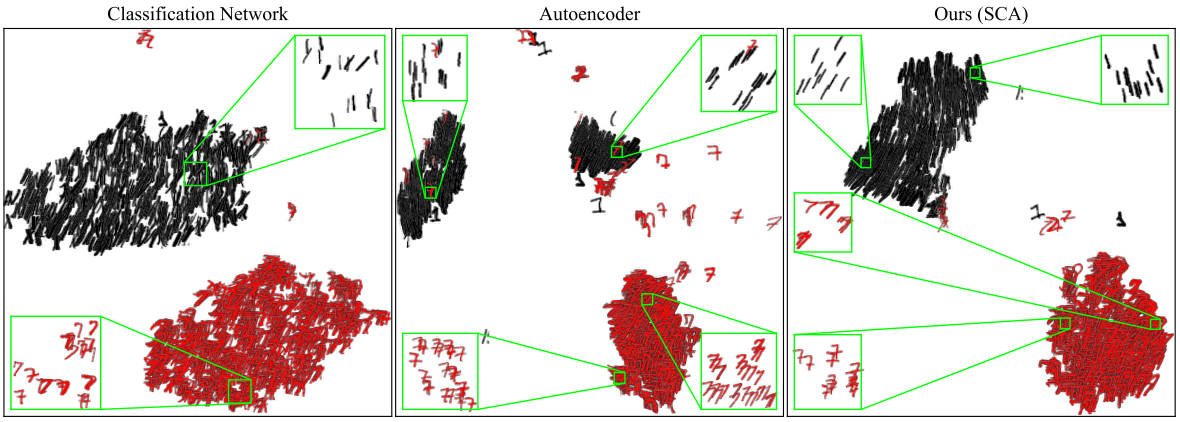

Figure 6: Learned t-SNE embeddings for digits “one” (black) and “seven” (red) from MNIST test set. We plot the digits at their t-SNE coordinates.

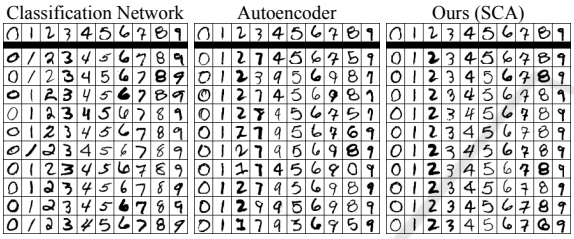

Figure 7: Qualitative retrieval results on the MNIST test set. The query images are on the top row and we show the top-10 retrieval results beneath. We use $\alpha = 0.2$ for SCA.

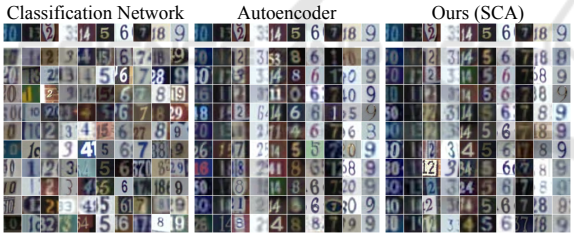

Figure 8: Qualitative retrieval results on the SVHN test set. The query images are on the top row and we show the top-10 retrieval results beneath. We use $\alpha = 0.4$ for SCA.

SCA enables retrieval of samples that are similar in both content and style, whereas the embedding learned by the classification network is invariant to style and the autoencoder does not learn much semantics. Note that for this dataset the learned style is mainly determined by the color of the digits, the background colors and textures, and adjacent distractor digits.

5 CONCLUSIONS

In this paper we have developed features that support fine-grained retrieval in a partly unsupervised manner, without requiring fine-grained labels. We proposed three different architectures leveraging autoencoders for this purpose: a Siamese autoencoder (SAE), a classifying autoencoder (CAE), and a separate classifier-autoencoder (SCA). We found that, despite its simplicity, the SCA architecture performs best in practice. The SCA avoids using a contrastive loss, which can be unstable to train. In addition, it retains fine-grained information by including an autoencoder. Finally, it better separates semantic class-level information from fine-grained properties compared to the CAE approach. This avoids detrimental competition between the classification and reconstruction loss during training. We quantitatively evaluate our approach and show that it leads to a significant improvement over the state of the art in a fine-grained sketch retrieval benchmark. In addition, it reliably retrieves correct object poses in a benchmark with images rendered from ShapeNet. We further demonstrate fine-grained retrieval of hand-written digits and images of house numbers based on style, without requiring style annotations.

As a disadvantage of our approach, various fine-grained properties (color, style, viewpoint, etc.) of complex data will be entangled in the autoencoding features, and the user cannot control which of these properties should be considered (ir)relevant for retrieval. In the future, we would like to investigate interactive techniques that allow the user to intuitively control the retrieval criteria on the fly and easily obtain desired results.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine

learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307.

Babenko, A. and Lempitsky, V. (2015). Aggregating local deep features for image retrieval. In *The IEEE International Conference on Computer Vision (ICCV)*.

Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014). *Neural Codes for Image Retrieval*, pages 584–599. Springer International Publishing, Cham.

Bui, T., Ribeiro, L., Ponti, M., and Collomosse, J. (2016). Generalisation and sharing in triplet convnets for sketch based visual search. *arXiv preprint arXiv:1611.05301*.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chen, K. and Salman, A. (2011). Extracting speaker-specific information with a regularized siamese deep network. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 298–306. Curran Associates, Inc.

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*, volume 1, pages 539–546. IEEE.

Eitz, M., Hays, J., and Alexa, M. (2012). How do humans sketch objects? *ACM Trans. Graph.*, 31(4):44:1–44:10.

Gordo, A., Almazán, J., Revaud, J., and Larlus, D. (2016). Deep image retrieval: Learning global representations for image search. In *ECCV*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Jgou, H., Douze, M., Schmid, C., and Prez, P. (2010). Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Mikulik, A., Perdoch, M., Chum, O., and Matas, J. (2013). Learning vocabularies over a fine quantization. *International Journal of Computer Vision*, 103(1):163–175.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5.

Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., and Schmid, C. (2015). Local convolutional features with unsupervised training for image retrieval. In *The IEEE International Conference on Computer Vision (ICCV)*.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). *Improving the Fisher Kernel for Large-Scale Image Classification*, pages 143–156. Springer Berlin Heidelberg, Berlin, Heidelberg.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Radenović, F., Tolias, G., and Chum, O. (2016). *CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples*, pages 3–20. Springer International Publishing, Cham.

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.

Sangkloy, P., Burnell, N., Ham, C., and Hays, J. (2016). The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*.

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

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 945–953, Washington, DC, USA. IEEE Computer Society.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Tolias, G., Sicre, R., and Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activati-

ons. In *International Conference on Learning Representations*.

Wang, F., Kang, L., and Li, Y. (2015). Sketch-based 3d shape retrieval using convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1883.

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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393.

Yu, Q., Liu, F., Song, Y. Z., Xiang, T., Hospedales, T. M., and Loy, C. C. (2016a). Sketch me that shoe. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 799–807.

Yu, Q., Yang, Y., Liu, F., Song, Y.-Z., Xiang, T., and Hospedales, T. M. (2016b). Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision*, pages 1–15.