Sketch Based Image Retrieval

Shantanu Deshpande
Georgia Institute of Technology
sdeshpande@gatech.edu

Naman Goyal
Georgia Institute of Technology
ngoyal32@gatech.edu

Abstract

Sketch based image retrieval is a task that has been explored a lot recently as an alternative method for image retrieval. We develop this task on The Sketchy Database, where we use Siamese and Triplet network to perform sketch based image retrieval. We employ deep residual learning network as the constituent network in the Siamese and Triplet architecture and use new data augmentation techniques for the task. Recent success of deep residual network (ResNet) suggests that it performs better than the previous architectures.

1. Introduction

The paper proposes a sketch based image retrieval system which allows users to draw a sketch and the system then finds corresponding similar image from the data set. The main advantage of sketch based image retrieval as opposed to text based retrieval is that it is easier to express the orientation and pose in the query sketch to find the required image as opposed to specifying these characteristics in text. The main challenge in sketch based image retrieval system is that it requires understanding of both sketch and image domain and then do comparison. Traditional approaches have depended on hand designed features which use the gradients or edges as features which are generally invariant across both image and sketch domains but such techniques can be bettered a lot. With the advent of deep convolutional networks, there has been an improvement in image recognition tasks in both image and sketch domain. So our goal in this paper is to train a deep convolutional network to learn the cross-domain representation for sketches and images and retrieve images with same pose and orientation as the query sketch. The main contribution of our work is to use ResNet [4] as a constituent of Siamese [3] and triplet network [7] architecture. In our knowledge, this has not been performed previously.

The next section of the paper talks about some of the related work in the field of sketch based image retrieval. A brief overview of the data augmentation techniques and deep network architectures is discussed in Section 3. The results obtained and discussion of results is done in Section 4 and the conclusion and future work is presented in section 5.

2. Related Work

Cao et al.[8] proposed a new descriptor called Symmetric-aware Flip Invariant Sketch Histogram (SYM-FISH) which included three steps for extraction. The Flip Invariant Sketch Histogram (FISH) descriptor is first extracted on the input image followed by exploring the symmetric character of the image by calculating the kurtosis coefficient followed by generating the SYM-FISH by constructing a symmetry table. Saavedra et al.[5] proposed detecting mid-level patterns called learned keyshapes for describing the sketches in two steps. Firstly an offline process is used to figure out a set of keyshapes and then the LKS descriptors are generated based on the detected set of keyshapes. Eitz et al.[6] developed a new data set consisting of 31 sketches each with 40 photos ranked by similarity and proposed a new descriptors based on the bag-of-features approach. Li et al.[9] show that sketches can be used for fine-grained retrieval within object categories. They introduce a mid-level representation of sketch that along with capturing the object pose also has the ability to traverse both the sketch and image domains. Wang et al.[2] proposed a cross-domain embedding methods that train Siamese networks to learn a common feature space for Sketches and 3D models.

3. Network Architecture

3.1. Siamese Network

A Siamese architecture contains two set of copies of the function G_W which may share the same set of parameters weights W, and cost module in cases when inputs to the networks are from same domain. In case of inputs to the network belonging to different domains, each set has its different parameter weights W. The output from this architecture are given as input to a loss module which is placed on top of them. The input to the architecture consist of a pair of images (X_1, X_2) and a output label Y. The im-

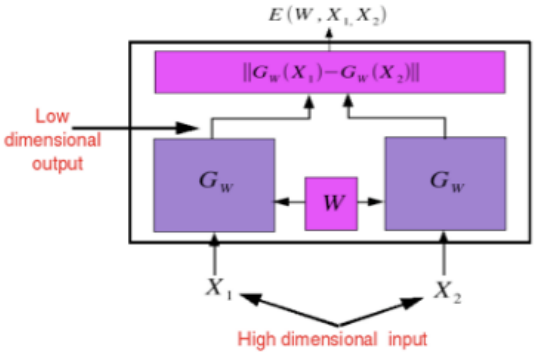

Figure 1. Siamese Network Architecture.

ages are given as input to the functions, which gives two outputs G(X_1) and G(X_2). The cost module takes those output G(X_1) and G(X_2) as input and then generates some form of distance $D_W(G_W(X_1), G_W(X_2))$. The loss function uses both, the generated D_W and label Y to produce the scalar loss value L_S. The parameter W to the network is updated using stochastic gradient descent rule. The gradients are computed by back-propagating the loss value, the cost value, and the two sets of G_W. The combined gradient is the addition of the contributions from the sets of G_W. In our case, the output label is generally binary and the input pair to the networks are either similar or dissimilar. Siamese networks use a contrastive loss function given by the equation

$$L = Y(I) \times d(S, I)^2 + (1 - Y(I)) \times max(0, md(S, I))^2$$

where S is an embedded sketch, I is an embedded image of the object instance, which may be same or different, d() signifies the Euclidean distance, and m is a margin. The contrastive loss function ensures that dissimilar sketchimage pairs are pushed away from each other and similar sketchimage pairs are pulled nearer to each other in feature space.

3.2. Triplet Network

A Triplet network which is inspired by above Siamese network is made up of 3 sets of the feed-forward network which may share the same parameters. When the network is fed with three samples, it gives as outputs two intermediate values, which correspond to the L2 distance values between the representation of two of its inputs from the representation of the third. In our case if the inputs to the Triplet networks are given as x, x+ and x, then the loss is expressed in the form input x should be closer to input x+ than to input x-. Triplet networks use ranking loss which is given by the equation

$$L = max(0, m + d(S, I+)d(S, I))$$

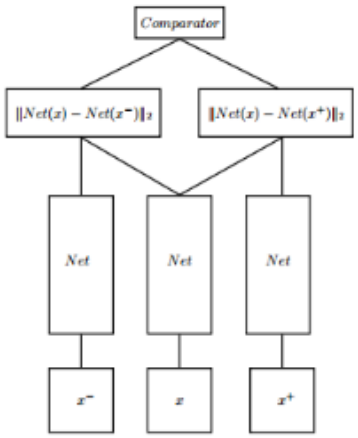

Figure 2. Triplet Network Architecture.

This ranking loss function is more expressive in terms of the fine-grained relationships than the Siamese loss.

3.3. Deep residual network

Residual Networks which are recently introduced by [4] modify a plain convolution network by inserting shortcut connection between input and output of every layer which converts the plain convnet into residual network. The identity shortcuts connections mentioned above can be added directly when the dimensions of the input and output for a layer are same. When there is a increase in the dimensions then there are two different ways of inserting connections between input and output. In first method the shortcut connection correspond to identity mapping and extra zero entries are padded for increasing the dimensions. The second method of adding connections is to match the dimensions which can be done by 11 convolutions. The basis of residual learning is that it consider some underlying mapping $H(x)$ which is to be fit by some of the stacked layers, with x being the inputs to the first of these layers. If we consider that multiple nonlinear layers can asymptotically approximate complicated functions, then the consideration is same as the consideration that they can asymptotically approximate the residual functions, which is, $H(x) - x$. In this case, the layers are made to explicitly approximate a residual function $F(x) := H(x)x$. The modifies the original function to $F(x) + x$. Now, thought it is possible to asymptotically approximate both the above forms to the the required functions as hypothesized, it might be easier to learn the latter form.

4. Dataset

We use Sketch Database for the task. Sketchy is a dataset that was recently collected sketch dataset from images. It has rich pose and orientation information since the human

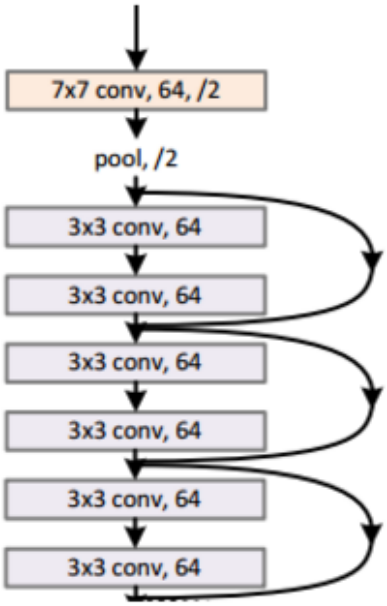

Figure 3. Residual Network Architecture.

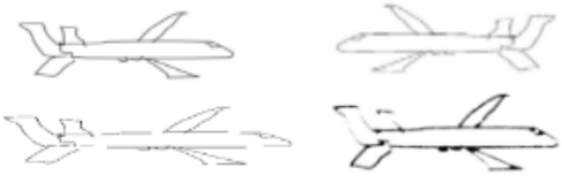

Figure 4. Data Augmentation. From top to bottom, left to right - Original, Mirrored, Stretched, Wiggle Noise added Sketches.

sketch makers were instructed to keep the same pose and orientation as shown in the image. The dataset consists of 74,425 sketches of 12,500 objects under 125 categories.

4.1. Data Augmentation

The following data augmentation techniques were used to generate more training sketches. CNNs are highly sensitive to input mirroring and the sketch images are horizontally mirrored to increase the training data set. Stretching is done to increase the horizontal width of the image keeping the pixels constant. Wiggle noise is added to the sketches to give the wiggling to the line strokes in the sketches.

5. Evaluation

5.1. Training Methodology

We do stepwise training for all our architectures. We initialize our weights for both image and sketches net with classification net trained on ImageNet [1]. We then pretrain our skech net on Eitz [6] dataset. To pretrain the image

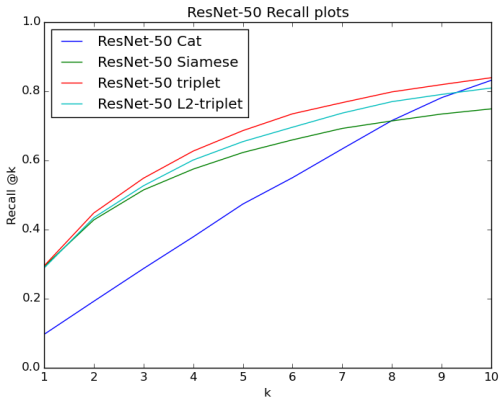

Figure 5. Recall plots for ResNet-50 architecture

Method	Recall@1	Recall@10
ResNet-50 Cat	10.98	82.56
GoogleNet-50 Cat	12.45	65.54
ResNet-50 Siamese	29.33	76.78
GoogleNet-50 Siamese	27.36	92.05
ResNet-50 Triplet	29.50	82.34
GoogleNet-50 Triplet	37.05	93.04

Table 1. Recall@1 and Recall@10 for our method (ResNet-50) and previous method

dataset, we downloaded 125,000 total images from flickr for 125 categories. Finally we train our net on embedding loss.

We use SGD as our training method. The learning rate is $1e-5$, weight delay 0.0002, momentum 0.9 and batch size 10 for triplet network and 15 for Siamese network. We trained the network for $250,000$ iterations.

5.2. Results

Figure 5 gives the recall plots for ResNet-50 architecture for networks trained on various different losses. As we can see the embedding loss gets improvement in 1-Recall to the network trained purely on categorization. This emphasizes embedding loss is useful in such cross domain task.

Table 1 shows the results of ResNet-50 along with GoogleNet architecture. As we can see the 1-Recall is better for ResNet-50 than GoogleNet in case of Siamese network. This shows that the higher depth of ResNet architecture is useful here also. Although the same results are not replicated for Triplet network. We suppose that this might be due to higher number of parameters in ResNet architecture it requires more positive training pairs.

5.3. Error Analysis

The top 1 per category recall for some of the categories is shown in figure 6 which gives some idea about the per-

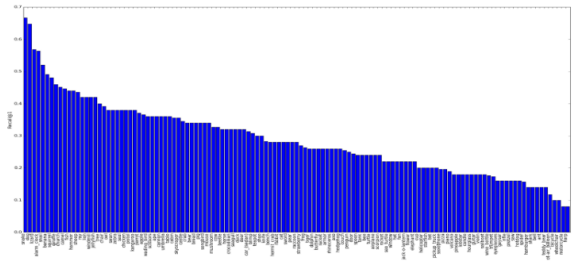

Figure 6. Per category recall.

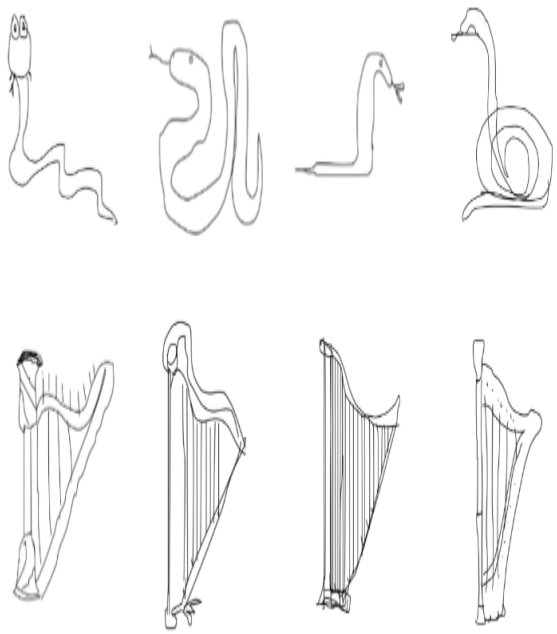

Figure 7. Random sketches of snakes and harps.

formance of the network architecture. It can be seen from figure 6 that for categories snake, dog the per category recall is very high whereas for categories like harp, wheelchair the per category recall is very low. This results are expected as sketches of snakes or dogs inherently have different pose and orientation information but sketches of harp generally will have same orientation. Figure 7 shows randomly sampled sketches of snakes and harps which shows that sketches of snakes have more variety in terms of pose and orientation whereas the harp sketches are monotonous in terms of pose and orientation. Due to this though a sketch query of harp may return an image of harp which may match the pose and orientation of the sketch but may not correspond to the image assigned to the sketch in the data set.

6. Conclusion

We have given a new architecture for sketch based image retrieval task. We showed that ResNet architectures inside a Siamese/Triplet network performs good. As previous work have shown we got better results than GoogleNet architecture on Siamese network. However we couldn't get similar improvements on Triplet network. We believe that this is due to higher number of parameters that needs more positive pairs to be trained. We think that the task of sketch based image retrieval should be posed as similarity prediction task for future uses because different photos can be of similar pose and orientations. Similarity rating collection for pairs of sketches in photos in Sketchy Database would be the future work on this task.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[2] W. Fang, K. Le, and L. Yi. Sketch-based 3d shape retrieval using convolutional neural networks. *arXiv preprint arXiv:1504.03504*, 2015.

[3] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[5] S. Jose and B. Juan. Sketch based image retrieval using learned keyshapes (lks). *Proceedings of the British Machine Vision Conference*, 2015.

[6] E. Mathias, H. Kristian, B. Tamy, and A. Marc. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Transactions on Visualization and Computer Graphics*, 2011.

[7] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[8] C. Xiaochun, Z. Hua, L. Si, G. Xiaojie, and L. Liang. Symfish: A symmetry-aware flip invariant sketch histogram shape descriptor. *IEEE International Conference on Computer Vision*, 2013.

[9] L. Yi, H. Timothy, S. Yi-Zhe, and G. Shaogang. Fine-grained sketch-based image retrieval by matching deformable part models. *Proceedings of the British Machine Vision Conference*, 2014.