Cousin Network Guided Sketch Recognition via Latent Attribute Warehouse

Kaihao Zhang,[1,3] **Wenhan Luo,**[2] ✉ **Lin Ma,**[2] **Hongdong Li**[1,3]
[1]Australian National University [3]Australian Centre for Robotic Vision
{kaihao.zhang, hongdong.li}@anu.edu.au quad {whluo.china, forest.linma}@gmail.com

Abstract

We study the problem of sketch image recognition. This problem is plagued with two major challenges: 1) sketch images are often scarce in contrast to the abundance of natural images, rendering the training task difficult, and 2) the significant domain gap between sketch image and its natural image counterpart makes the task of bridging the two domains challenging. In order to overcome these challenges, in this paper we propose to transfer the knowledge of a network learned from natural images to a sketch network - a new deep net architecture which we term as *cousin network*. This network guides a sketch-recognition network to extract more relevant features that are close to those of natural images, via adversarial training. Moreover, to enhance the transfer ability of the classification model, a sketch-to-image attribute warehouse is constructed to approximate the transformation between the sketch domain and the real image domain. Extensive experiments conducted on the TU-Berlin dataset show that the proposed model is able to efficiently distill knowledge from natural images and achieves superior performance than the current state of the art.

Introduction

Automatic hand-drawn sketch recognition is an important task in computer vision, and has found many real-world applications. For example, sketch can be used as a convenient user query input for content based image retrieval. This becomes a popular user interface thanks to the wide use of digital pen. However, sketch recognition remains a very challenging problem, due to multi-fold reasons: (1) Sketches often exhibit high abstraction and intra-class variance. Different person may draw the same object very differently. (2) The amount of available sketch data is very scarce, and can be expensive to obtain. Therefore it is unable to meet the requirement of training data needed by most deep neural networks.

In this paper, we notice that the amount of available natural images is enormous and keeps growing. We intend to bridge the domain of natural images with that of sketches by the idea of transfer learning, to boost the performance of sketch recognition. However, as show in Fig. 1, compared with natural images, sketches are typically absent of many visual attributes like color, textures *etc*, which makes the domain transfer task difficult. To address this issue, we propose to exploit models trained solely on natural images (ImageNet) to guide the model aimed for sketch classification. This is achieved by using *Adversarial Learning*. In order to cope with the loss of discriminative power due to the absence of visual attributes in sketch images, we develop a new mechanism called the "attribute warehouse". Specifically, we adopt a neural network of ResNet structure as our basis, which learns feature representation for classification in different levels from the ImageNet dataset. With the natural images from the TU-Berlin-extend (Zhang et al. 2016) and TU-Berlin (Eitz, Hays, and Alexa 2012) datasets, this pre-trained network is fine-tuned to adapt to the categories of sketches, *i.e.*, from 1000 classes of natural images to 250 classes of sketch images. This fine-tuned network is termed as a *cousin network* to guide the target network of sketch classification to learn the latent representation commonly shared by both real images and sketch images. The target network is then initialized as the cousin network. A two-

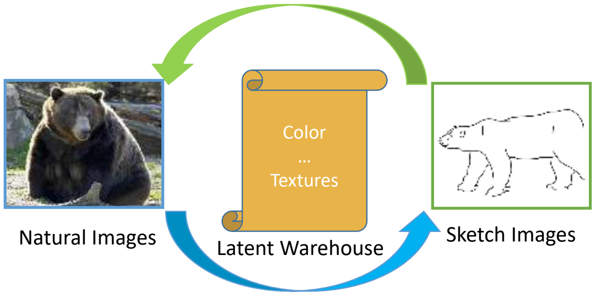


Figure 1: The relationship between natural images and sketches. Compared with natural images, sketches are absent of many visual attributes. These different attributes increase the differences of features captured from two kinds of images, thus it is difficult for models trained from natural images to recognize sketch images. Meanwhile, it is also difficult for sketch recognition model to learn under the guidance of model pre-trained on natural image datasets. To address this issue, we propose a Latent Sketch-to-Image Warehouse Network, which can learn the domain gap between two kinds of images.

stream structure is derived for training the target network specifically for sketch classification. As shown in Fig. 2, the parallel cousin network and the target network are provided with natural images and sketch images, respectively. Fixing the cousin network, the target network is updated with three terms of losses. Firstly, we enforce that the high-level features from these two streams are close in the latent feature space. This loss guides the target network to learn from the cousin network directly. Secondly, the features from these two streams are discriminated by a discriminator network in an adversarial manner. Different from the discriminator in a traditional generative adversarial network which takes samples in the visual space as input, we conduct discrimination between the abstract features from higher levels. By doing so, the target network is taught implicitly. Both these terms are helpful to inherit knowledge from the network trained on natural images. Thirdly, a cross-entropy loss is minimized for the sketch classification task.

To bridge the domain gap between sketches and real images, we learn a module of transformation from sketch to image to compensate the lost distinctive cues leading to a performance degradation. This module learns a latent attribute warehouse from sketch to real image, which transforms sketch to real image crossing a latent space. With the latent attributes, the target network learns more discriminative features which make the sketches to be conceptually closer to real images, further improving the performance of sketch classification. Note that this latent attribute warehouse is learned in low levels. After that, a fine-tuning step considering the recognition task is carried out. During inference, the target network is chained with the latent attribute warehouse. An input sketch image is forwarded through the learned transformation module in the low levels in the network, approaching the samples located in the latent space shared by sketch and real images. A classification layer outputs the prediction of class labels.

We have conducted extensive experiments on the public dataset of TU-Berlin. Both ablation study and comparison with other state-of-the-art methods verify the effectiveness of the guidance from cousin network and the power from the latent attribute warehouse.

Related Work

Sketch recognition/retrieval (Eitz et al. 2011) has been studied for years. Roughly, existing approaches can be classified by whether it employs deep learning or not. Classical (non-deep-learning) sketch recognition methods often use handcrafted features. For example, local features along with the bag-of-words model (Sivic and Zisserman 2003) are used to represent sketch images and both SVM and the nearest-neighbor classification are applied (Eitz, Hays, and Alexa 2012). SVM is a popular choice of classifier (Li, Song, and Gong 2013). Schneider *et al.* (Schneider and Tuytelaars 2014) use Fisher vector to encode sketch images for recognition. Li *et al.* (Li et al. 2015) use multi-kernel learning to learn useful feature representation from various local features. To alleviate annotation cost in sketch recognition, active learning is introduced in (Yanık and Sezgin 2015). Cao *et al.* (Cao et al. 2013) propose a novel descriptor named Symmetric-aware Flip Invariant Sketch Histogram (SYM-FISH) to describe sketches inspired by the shape context.

Recent years have witnessed the success of deep learning in various tasks of computer vision, such as object detection (Ren et al. 2015), image recognition (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2015; He et al. 2016), object tracking (Luo et al. 2018), facial recognition (Sun, Wang, and Tang 2013; Zhang et al. 2015; 2017) and multimedia analysis (Simonyan and Zisserman 2014; Ledig et al. 2017; Xiong et al. 2018; Zhang et al. 2019). For sketch recognition, a variant of Siamese CNN is proposed in (Wang, Kang, and Li 2015) to match sketch and photo without special process of sketch images. Different network structures are designed according to statistics from sketch images rather than natural images in (Yu et al. 2015; 2017). With two novel techniques of data augmentation, performance is improved. Yu *et al.* (Yu et al. 2016) address the task of image retrieval given a sketch image. They propose a model of triplet ranking along with data augmentation. Specific problem of forensic sketch recognition is dealt in (Ouyang et al. 2016). To handle the sketch-photo modality gap due to the forgetting process, a model is trained to reverse the forgetting process and it is proved to recover facial details. Inspired by the idea of Hashing in natural image retrieval, deep sketch hashing is proposed in (Liu et al. 2017) to encode sketch images with a semi-heterogeneous deep architecture. It handles the gap between natural image and sketch image well. For 3D cases, Xie *et al.* (Xie et al. 2017) and Dai *et al.* (Dai et al. 2017) propose deep models to address sketch-based 3D shape retrieval. In this paper, we build a latent attribute warehouse for this purpose. A similar idea of employing natural image for sketch recognition is proposed in (Zhang et al. 2016). Specifically, this method divides sketch classification into three steps. Firstly, a CNN is trained to output top-5 predictions given a sketch image. Then natural images corresponding to the returned top sketches are paired with the given sketch and input to a fine network trained with positive and negative pairs of sketch and natural images. Finally, the predictions from a fine network are fused to give final results based on metric SVM. Though this method achieves the state-of-the-art results, it has two problems: 1) multiple networks are used to make final decision, which poses a large amount of computation cost, 2) its performance heavily depends on the top-5 predictions in the first step. In order to alleviate their problems, we build a cousin network guided learning method, which uses an end-to-end model in the testing stage and obtains better performance.

Cousin Network Guided Feature Learning

In this section, we firstly introduce feature learning by transferring knowledge from natural images. Then, a two-stream architecture of the sketch recognition network will be developed.

Guided Feature Learning

To train deep networks, it is often necessary to have a large amount of training data. Unfortunately, for the task of free-hand sketch recognition there is not sufficient training data

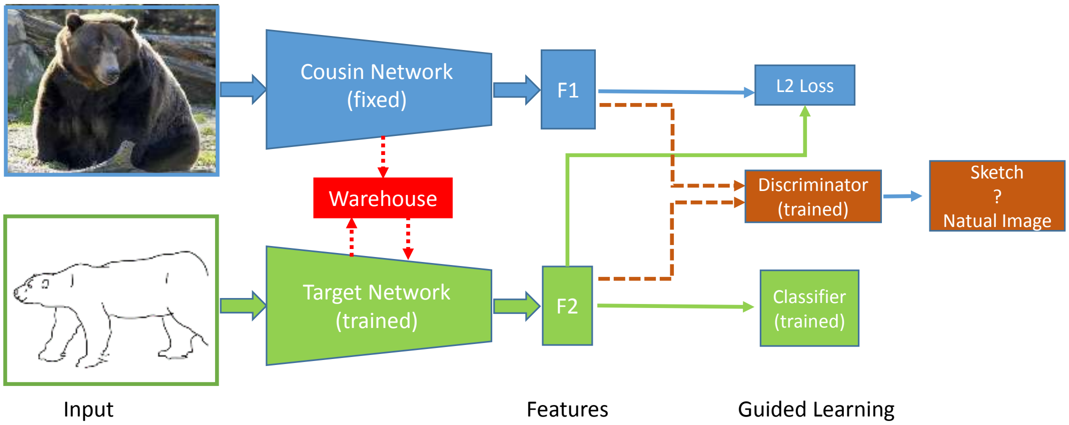


Figure 2: The proposed Cousin Network Guided Sketch Classification Network (CNG-SCN). Our model consists of a cousin network (top), a target network (bottom), a discriminator, a classifier, and a warehouse module. The cousin network is pre-trained on the ImageNet dataset and has a strong power of learning features of natural images. We fine-tune it and fix it in advance. The target network, the discriminator, and the classifier are trained jointly in our CNG-SCN architecture. The features of natural and sketch images extracted from the two networks are fed into the discriminator, while the classifier only takes as input the features from the target network. All modules except warehouse network are introduced in Section: Cousin Network Guided Feature Learning. The latent attribute warehouse is learned via the nonlinear relationship between sketch and real images, which is discussed in Section: Latent Sketch-to-Image Warehouse.

available. In contrast, natural/real images are relatively easy to obtain (such as ImageNet). It is natural to transfer knowledge learned from the natural image domain to the target domain, *i.e.*, sketches.

Deep convolutional neural networks are strong at extracting multi-scale feature representation from low-level edges to high-level abstractions. The learned feature representation is effective for the task of classification. High-level abstractions like shapes are similar in both domains despite of the absence of color and texture. Thus, to take the advantage of the well learned features, we employ a model trained with sufficient natural images to guide the training of the target network for sketch recognition.

Specifically, in our method, a neural network of ResNet-18 structure (He et al. 2016) trained on the dataset of ImageNet is adopted. As this model is originally trained on the 1000 classes of objects and the class number for sketch recognition is 250, this model is fine-tuned for the adaption of sketch recognition. The additional data in the dataset of TU-Berlin-extend is suitable for the adaption. Compared with the dataset of TU-Berlin, each class of sketch is extended with real images of the same category from the ImageNet dataset. We replace the classification layer of 1000 classes with a new 250-class layer and use the additional real images of the same categories to fine tune the original ResNet-18 network. This fine-tuned network is termed as a *Cousin Network* to guide the following training of the target network.

With the availability of the cousin network, a two-stream architecture is composed of the cousin network and the target network as shown in Fig. 2. Natural images and sketch images are input to the cousin network and the target network individually and features are extracted. To accomplish the guidance, we enforce these two instances of feature to be close in terms of not only a predefined distance metric in feature space but also a learned metric by adversarial learning. The former predefined distance metric is accomplished as a layer of MSE loss. The latter metric is learned as adversarial learning by discriminating feature instances from real images and sketch images. Adversarial learning has been popular since the seminal work GAN by (Goodfellow et al. 2014). The discriminator in conventional GAN is trained to distinguish samples in visual space. On the contrary, in this work we train a discriminator to tell feature instances of sketch images from those of natural images, until it is fooled. The insight behind is, discrimination between sketch image and real image is easy in the visual space, thus training a discriminator by doing so is not helpful in sketch classification. While the discrimination between high-level abstract features from sketch image and real image is difficult, so training such a discriminator towards fooling it is beneficial in teaching the target network to learn powerful feature representation.

Network Architecture

The network architecture is shown in Fig. 2. In general, it is a two-stream structure, one stream (cousin network) for real images while the other one (target network) for sketch images. As mentioned above, the cousin network is a finetuned ResNet-18 structure. The target network is initialized as the cousin network trained in the two-stream structure with the cousin network being fixed. There are three output branches of this two-stream structure. 1) The first branch takes feature instances from the two streams and computes the distance between them. The cousin network guides the target network to learn similar features in the high dimension feature

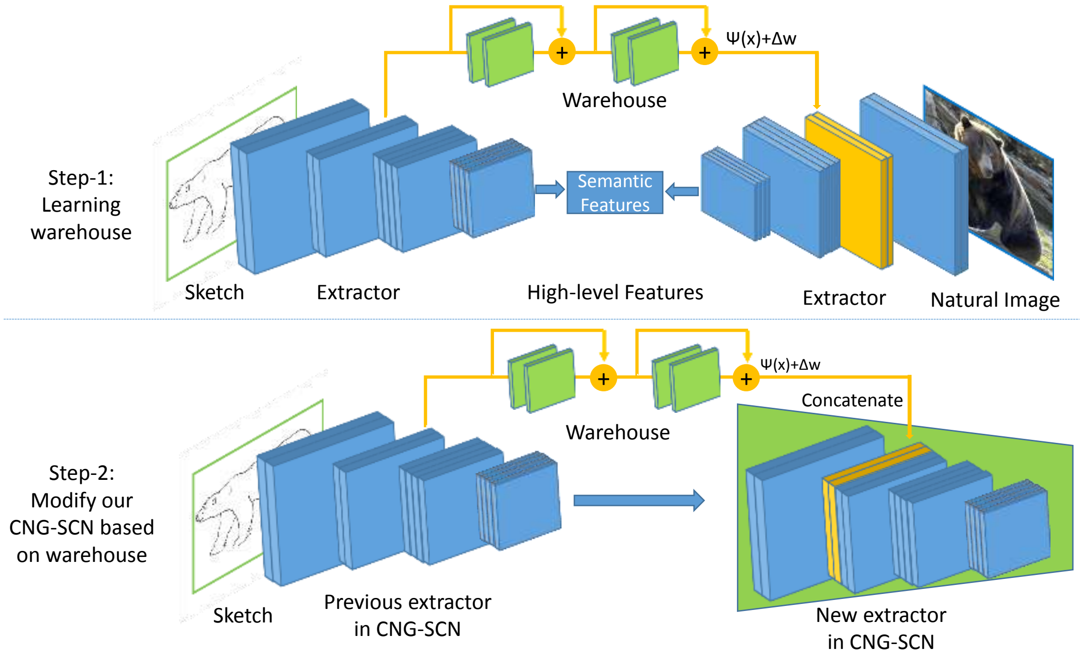


Figure 3: The illustration of learning the latent Sketch-to-Image warehouse. The feature extractors are components of CGN-SCN. The warehouse networks are stacks of convolutional layers and all of them do not change the size of input. In the first step, we train the warehouse network based on the feature maps of sketches and natural images. Then we integrate the warehouse network into the CNG-SCN to improve the performance in the second step.

space. 2) The second branch is a discriminator which tries to distinguish feature of the cousin network from that of the target network. This discriminator consists of several convolutional layers, fully-connected layers and a binary classification layer. 3) The third one carries out the classification task taking the feature from the target network as input. It is composed of a few fully-connected layers and a soft-max layer for multi-class classification.

Loss Function. The loss corresponding to the first branch is the MSE between features from sketch and real image. It is formulated as,

$$\mathcal{L}_{dis} = \left\| T\left(\mathbf{z}\right) - \tilde{T}\left(\mathbf{x}\right) \right\|_2^2 , \tag{1}$$

where **z** and **x** are real image and sketch image, T and $\tilde{T}$ are the feature extraction of the cousin network and the target network respectively.

The loss term of the discriminator D is,

$$\mathcal{L}_{adv} = \min_{\tilde{T}} \max_{D} \mathbb{E}\left[logD\left(\tilde{T}\left(\mathbf{x}\right)\right)\right] + \mathbb{E}\left[log\left(1 - D\left(T\left(\mathbf{z}\right)\right)\right)\right] . \tag{2}$$

The loss term of the classification branch is the cross entropy as follows,

$$\mathcal{L}_{cla} = \sum P\left(\mathbf{x}\right) logQ\left(\mathbf{x}\right) , \tag{3}$$

where $P\left(\cdot\right)$ and $Q\left(\cdot\right)$ are prediction and ground truth respectively.

The overall loss is a weighted summation of the above loss terms with weights α and β,

$$\mathcal{L} = \mathcal{L}_{dis} + \alpha \cdot \mathcal{L}_{adv} + \beta \cdot \mathcal{L}_{cla} . \tag{4}$$

Learning and Optimization. The three branches along with the target network are tuned jointly with the loss function above. We use the stochastic gradient descent method for optimization.

Latent Sketch-to-Image Warehouse

Knowledge Transfer

The lost attributes like color and textures lead to ambiguity in recognizing sketch images compared with natural image recognition. Though we have guided the target network to learn similar high-level features while the domain gap is still the obstacle in transferring as there lacks straightforward connection between these two domains. We here propose to learn the connection between sketch and image as a transformation. The argument is that, direct learning the transformation can compensate the lost attributes, termed as *latent attribute warehouse*, which are discriminative for sketch recognition.

To learn such a transformative attribute warehouse, we link the target network to the cousin network with module networks in low layers. Our warehouse does not include high level features, because they are too semantic (Hariharan et al. 2015). The module network in each level is of the similar structure. The warehouse path is expected to learn the latent attributes. Along with the short-cut path for identical transfer, the sketch domain $\mathbf{S}$ is transformed to the real-image domain $\mathbf{I}$.

The goal of the warehouse is to discover the latent difference between $\mathbf{I}$ and $\mathbf{S}$ and learn the transformation from $\mathbf{S}$ to $\mathbf{I}$. The transformation process can be described as

$$\mathbf{z} = \Phi\left(\mathbf{x}; \mathbf{w}\right) \,, \tag{5}$$

where $\mathbf{x}$ and $\mathbf{z}$ are the instances from $\mathbf{S}$ and $\mathbf{I}$, respectively, $\mathbf{w}$ is the latent sketch-to-image warehouse, Φ is the approximation function that controls the transfer from $\mathbf{S}$ to $\mathbf{I}$.

Due to the high diversity of attributes between $\mathbf{S}$ and $\mathbf{I}$, it is difficult for Φ to conduct the transfer directly. In order to learn the warehouse, we try to learn the latent warehouse via a deep structure, and therefore Φ can be represented as

$$\Psi(\mathbf{z}) = \hat{\Phi}\left(\Psi(\mathbf{x}); \mathbf{w}\right) \,, \tag{6}$$

where $\hat{\Phi}$ denotes the learning process via deep module. If $\Psi(\mathbf{z})$ and $\Psi(\mathbf{x})$ learn sufficiently semantic knowledge, then the transfer can be represented as linear shifting in the latent space (Li et al. 2017), and thus the above equation can be reformulated as

$$\Psi\left(\mathbf{z}\right) = \Psi\left(\mathbf{x}\right) + \Delta\mathbf{w} \,, \tag{7}$$

where $\Delta\mathbf{w}$ denotes the sketch-to-image warehouse in latent space.

Learning Warehouse

The overall procedure in Fig. 3 to learn the latent warehouse includes two steps. In the first step, we extract features of sketches and photo images and use a CNN network to learn the lost information of sketches relatively to photos. In the second step, we add the lost information contained in the warehouse into the extractor of the network in the previous section to derive a new feature extractor.

As shown in Fig. 3, feature maps of sketch are passed to the short-cut path and the warehouse path individually. Output feature maps are concatenated and forwarded to the following convolutional layers for the purpose of deriving the same number of feature maps comparable with those in the cousin network.

The process of learning warehouse $\Delta\mathbf{w}$ focuses on attribute factors such as color and appearance. A straight method is to learn the average difference between $\mathbf{S}$ and $\mathbf{I}$, which is represented as

$$\Delta\mathbf{w} = \frac{1}{m}\sum_{i=1}^{m} \Psi\left(\mathbf{z}_i\right) - \frac{1}{n}\sum_{j=1}^{n} \Psi\left(\mathbf{x}_j\right) \tag{8}$$

where m and n are the numbers of input samples from $\mathbf{I}$ and $\mathbf{S}$, respectively. We can use this equation to learn the warehouse if, 1) there are sufficient training samples to alleviate the issue of intra-class variation, or 2) samples from $\mathbf{S}$ and $\mathbf{I}$ are of similar attributes (ignorable intra-class variation) even the data is not sufficient. However, the training samples are hardly sufficient and we cannot make sure samples from $\mathbf{S}$ and $\mathbf{I}$ have similar attributes even they come from an identical class.

Generating Training Sets. In order to satisfy the second assumption above, we search K-nearest neighbors of each training sample to construct the training set. Specially, for an input sketch $\mathbf{x}$, the latent attribute warehouse $\Delta\mathbf{w}$ is learned as

$$\Delta\mathbf{w}\left(\mathbf{x}\right) = \frac{1}{K}\sum_{\mathbf{z}_i \in N_{\mathbf{I}}^{K}(\mathbf{x})} \Psi\left(\mathbf{z}_i\right) - \frac{1}{K}\sum_{\mathbf{x}_j \in N_{\mathbf{S}}^{K}(\mathbf{x})} \Psi\left(\mathbf{x}_j\right) \tag{9}$$

where $N_{\mathbf{I}}^{K}(\mathbf{x})$ and $N_{\mathbf{S}}^{K}(\mathbf{x})$ denote the K-nearest neighbors in $\mathbf{I}$ and $\mathbf{S}$, respectively. To search the K-nearest neighbors, instances are forwarded in a pre-trained ResNet which is trained on both photos and sketch images.

Note that, transformation networks in different levels are tuned individually. After they are tuned, we disconnect the link from the transformation network to the cousin network, and redirect it to the subsequent layers following the current layer in the target network. This network is finetuned regarding the sole task of sketch classification. This finetune step will enhance the connections across layers on different levels of depth and the result network is used as the classifier for evaluation.

Network Design. In order to learn the latent attribute warehouse, we stack three groups of CNN network to capture the non-linearity transfer. Each group stacks Conv-BN-ReLU-Conv-BN-ReLU-Conv layers. In addition, a residual layer is adopted between the input and output of each group.

Experiments

Datasets & Evaluation Metrics

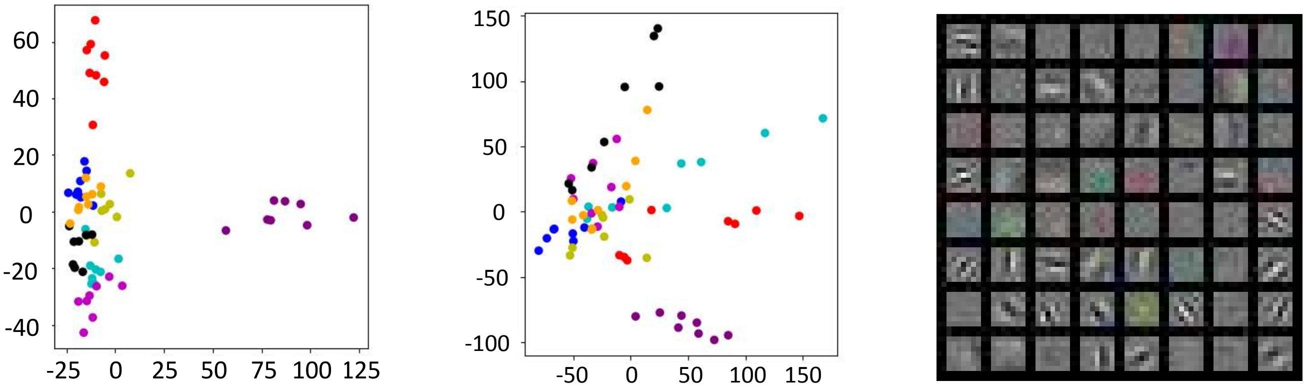
TU-Berlin Dataset. This dataset is proposed in (Eitz, Hays, and Alexa 2012) for sketch recognition, which contains 250 classes of objects. Each class has 80 sketches.

TU-Berlin-extend Dataset. The TU-Berlin dataset is extended by adding real images (Zhang et al. 2016). Images from ImageNet of the corresponding classes are employed as extension. There are 764 natural images extended for each category on average, and 191,067 natural images for all categories.

Evaluation Metrics. Following (Eitz, Hays, and Alexa 2012), we set up 8 kinds of training-testing protocols. For each protocol, a number of t sketch instances in each category is used for training and the rest sketches in the category are used for testing. Values of t are 16, 24, 32, 40, 48, 56, 64 and 72 for these protocols. We use mean Average Precision (mAP) throughout our experiments.

Implementation Details

In the training stage, the input sketch and images are both resized to 256×256. We crop $224 \times 224 \times 3$ patches from the resized images and flip them horizontally at random. All weights are initialized as a Gaussian distribution (mean=0 and standard deviation = 0.02). Momentum is set at 0.9. α and β are set to 0.3. The **whole training procedure** is as follows. Firstly, The CNG-SCN model is trained without warehouse module. Then we fix the above pre-trained model and train the warehouse individually. After that, the CNG-SCN model is combined with warehouse module and fine-turned



(a) Features

(b) Features

(c) Filters

Figure 4: (a) and (b) show the two-dimensional sketch features extracted from eight categories, which are outputs of the last convolutional layers of the CNG-SCN and the ResNet model, respectively. It is not difficult to observe that, samples are separated more clearly by CNG-SCN than the plain ResNet model. (c) is the visualization of filters in the first layer of the proposed CNG-SCN.

to obtain a new model. We combine the two kinds of CNG-SCN models to make final prediction.

The Effectiveness of Cousin Network Guided Learning

The CNG-SCN has the advantage of learning better sketch feature representation aided by the cousin network than direct training a single CNN. To demonstrate the effectiveness of this model, we conduct experiments with the last protocol, *i.e.*, 72 training samples in each class. Table 2 shows the accuracies of different models. Our CNG-SCN model improves the ResNet by varying degrees in terms of different rank-*n* accuracies, thanks to the guided feature learning from the cousin network. It proves that the improvement is due to our contribution of distilling knowledge from natural images, rather than the capacity of ResNet. It is worthy to note that, the model of *ResNet (mixed)* means utilizing the mixture data of both sketch images and real images, and its performance is even worse than the model *ResNet* trained with sole sketch images. It suggests that, naive using natural images with sketch images cannot guarantee a more powerful model.

We also carry out an ablation study to investigate how effective of features learned in the ResNet and the CGN-SCN. Fig. 4(a) and 4(b) show the features learned with CNG-SCN and baseline ResNet. Features of eight classes are extracted by the last convolutional layer. We conduct PCA to reduce the dimension to 2 and plot them in the figures. These figures suggest that, clusters corresponding to the baseline ResNet overlap with each other significantly, making it difficult to distinguish different sketch images. While feature samples are better separated by the CNG-SCN. To show what the CNG-SCN has learned, we visualize the filters in the low-level layer, shown in Fig. 4(c). Evident edge and color patterns can be observed in this figure.

The Effectiveness of Sketch-to-Image Warehouse

To verify the effectiveness of the learned attribute warehouse, we further conduct comparison between CGN-SCN and CGN-SCN plus the attribute warehouse. Table 2 shows the advantage of latent warehouse for sketch classification. Our CNG-SCN model plus warehouse can further improve the ResNet across different rank-k accuracy metrics, showing its additional benefit due to that the gap between natural image domain and sketch domain is bridged by the learned attribute warehouse.

We are also curious about what the warehouse has learned, thus we analyze the feature maps resulted from the network input. The final 512-dimensional feature maps of the proposed warehouse-based model and non-warehouse model are visualized and shown in Fig. 5, respectively. In order to alleviate the effect of absolute values like Fig. 4, features are normalized to a range of 0∼1.0. We calculate difference of pairs and light correspond locations if their values are larger than a fixed threshold value. Thus the more similar features of sketch images correspond to fewer light positions. These figures show features of identical class captured by model with warehouse network have more common activated neurons than the features captured by model without warehouse network. This is because our model with warehouse network concatenates the features with latent attribute information, which is beneficial in learning features implied in sketch images and thus improves the performance.

Comparison with State-of-The-Art Methods

We compare the proposed method with the state-of-the-art methods in Table 1. The results of other methods are reported by (Zhang et al. 2016) . (Eitz, Hays, and Alexa 2012) and (Schneider and Tuytelaars 2014) are two typical methods based on hand-crafted features. Popular deep convolutional neural networks like NIN (Lin, Chen, and Yan 2013), AlexNet (Krizhevsky, Sutskever, and Hinton 2012), VGGNet (Simonyan and Zisserman 2015), GoogLeNet

Table 1: The rank-1 accuracies [%] of different models on the TU-Berlin sketch dataset. The best results are shown in bold, which also applies in the following tables.

Model	16	24	32	40	48	56	64	72
Eitz *et al.* (Eitz, Hays, and Alexa 2012)	41	44	46	50	51	54	55	55
FisherVector (Girshick 2015)	52	56	59	62	65	66	67	68
NIN (Lin, Chen, and Yan 2013)	61.90	65.50	68.05	70.61	71.50	72.02	73.82	74.40
AlexNet (Krizhevsky, Sutskever, and Hinton 2012)	62.4	67.6	68.12	69.86	71.65	72.62	74.02	75.02
VGGNet (Simonyan and Zisserman 2015)	60.65	63.05	65.54	67.34	69.54	73.83	75.17	76.53
GoogLeNet (Szegedy et al. 2015)	59.61	62.45	67.48	69.19	70.50	71.50	72.40	75.25
ResNet (He et al. 2016)	57.84	62.51	66.62	69.82	71.65	72.65	74.83	74.46
SketchNet (Zhang et al. 2016)	64.37	66.20	71.19	69.57	73.62	73.43	76.50	77.41
CNG-SCN	64.88	68.67	72.12	73.89	75.27	76.32	77.01	78.71
CNG-SCN (warehouse)	**66.43**	**69.77**	**73.54**	**74.60**	**76.36**	**76.79**	**78.36**	**80.10**

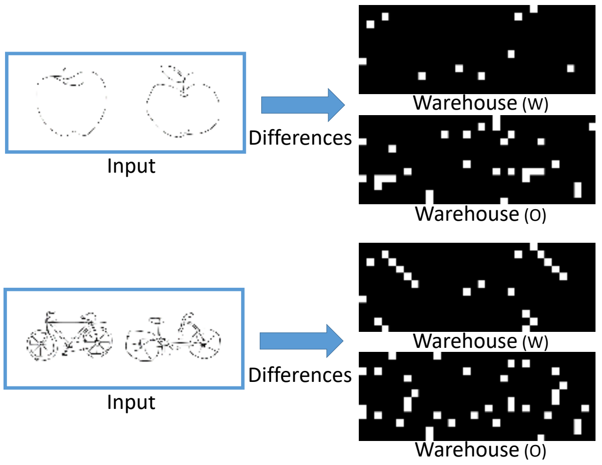


Figure 5: The differences of the captured 512-dimensional sketch features between pairs of sketch images. The left images show test pairs from TU-Berlin dataset. All pairs are of the same class. We calculate the difference of the last hidden layer features, which are shown in the right, and light the corresponding locations if they are larger than one threshold value. Thus the fewer light positions, the more similar their features are. The first and third rows in the right are features learned based on model with warehouse network, while the second and fourth rows are features learned based on model without warehouse network. For the sake of convenient illustration, we rearrange the features as 12 x 32.

(Szegedy et al. 2015) and ResNet (He et al. 2016) are employed for sketch classification in (Zhang et al. 2016). We also compare with these networks. SketchNet (Zhang et al. 2016) is a recent model utilizing web image for sketch classification, and achieves the state-of-the-art performance. Note that, the SketchNet we compare with is without the metric learning module. This module is learned by traditional metric SVM after the network is trained, not in an end-to-end manner. It can be optionally added to any network once it is ready. Reader may refer (Zhang et al. 2016) for more details. Thus, for fair comparison we compare with SketchNet without metric to focus on effectiveness of sole networks.

Table 2: The rank-1, rank-2, rank-3, rank-4, and rank-5 accuracies [%] of different models on TU-Berlin.

Model	rank-1	rank-2	rank-3	rank-4	rank-5
ResNet	74.46	84.29	88.03	90.47	91.97
ResNet (mixed)	74.11	83.54	87.53	90.22	91.92
CNG-SCN	78.71	86.43	89.88	92.17	93.22
CNG-SCN (warehouse)	**80.10**	**87.08**	**90.42**	**92.32**	**93.57**

We report the rank-1 accuracies with regard to all the 8 training-testing protocols. Results show that, 1) deep models consistently outperform methods based on hand-crafted features, which verifies the learning ability by deep models, 2) the proposed CNG-SCN outperforms other methods and the model with warehouse further improves the performance. We believe the improvement originates from the careful design of network to transfer knowledge from natural image domain to sketch domain. We also note that, the testing in SketchNet is composed of two steps, obtaining top-5 prediction by the first network and forwarding five pairs of sketch and real images in the second network, while the testing step in our method is in an end-to-end fashion.

Conclusion

We have proposed a deep network for sketch recognition by transferring knowledge from a cousin network trained on natural images. The cousin network guides the target network to extract powerful features for recognition in an adversarial manner. A latent attribute warehouse is developed to improve the transfer ability from sketch to natural image domain, boosting significantly the performance. Experiment results demonstrate the effectiveness of the proposed model.

Acknowledgment

Kaihao Zhang's PhD scholarship is funded by Australian National University. Hongdong Li is CI (Chief Investigator) on Australia Centre of Excellence for Robotic Vision (CE14) funded by Australia Research Council. This work is also supported by 2017 Tencent Rhino-Bird Elite Graduate Program.

References

Cao, X.; Zhang, H.; Liu, S.; Guo, X.; and Lin, L. 2013. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *ICCV*.

Dai, G.; Xie, J.; Zhu, F.; and Fang, Y. 2017. Deep correlated metric learning for sketch-based 3d shape retrieval. In *AAAI*.

Eitz, M.; Hildebrand, K.; Boubekeur, T.; and Alexa, M. 2011. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*.

Eitz, M.; Hays, J.; and Alexa, M. 2012. How do humans sketch objects? *TOG*.

Girshick, R. 2015. Fast R-CNN. In *ICCV*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Hariharan, B.; Arbeláez, P.; Girshick, R.; and Malik, J. 2015. Hypercolumns for object segmentation and fine-grained localization. *TPAMI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A. P.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

Li, Y.; Hospedales, T. M.; Song, Y.-Z.; and Gong, S. 2015. Free-hand sketch recognition by multi-kernel feature learning. *CVIU*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. In *NIPS*.

Li, Y.; Song, Y.-Z.; and Gong, S. 2013. Sketch recognition by ensemble matching of structured features. In *BMVC*.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. In *ICLR*.

Liu, L.; Shen, F.; Shen, Y.; Liu, X.; and Shao, L. 2017. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*.

Luo, W.; Sun, P.; Zhong, F.; Liu, W.; Zhang, T.; and Wang, Y. 2018. End-to-end active object tracking via reinforcement learning. In *ICML*.

Ouyang, S.; Hospedales, T. M.; Song, Y.-Z.; and Li, X. 2016. Forgetmenot: Memory-aware forensic facial sketch matching. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Schneider, R. G., and Tuytelaars, T. 2014. Sketch classification and classification-driven analysis using fisher vectors. *TOG*.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Sivic, J., and Zisserman, A. 2003. Video google: A text retrieval approach to object matching in videos. In *ICCV*.

Sun, Y.; Wang, X.; and Tang, X. 2013. Deep convolutional network cascade for facial point detection. In *CVPR*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.; et al. 2015. Going deeper with convolutions. In *CVPR*.

Wang, F.; Kang, L.; and Li, Y. 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In *CVPR*.

Xie, J.; Dai, G.; Zhu, F.; and Fang, Y. 2017. Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval. In *CVPR*.

Xiong, W.; Luo, W.; Ma, L.; Liu, W.; and Luo, J. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*.

Yanık, E., and Sezgin, T. M. 2015. Active learning for sketch recognition. *Computers & Graphics*.

Yu, Q.; Yang, Y.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. 2015. Sketch-a-net that beats humans. In *BMVC*.

Yu, Q.; Liu, F.; Song, Y.-Z.; Xiang, T.; Hospedales, T. M.; and Loy, C.-C. 2016. Sketch me that shoe. In *CVPR*.

Yu, Q.; Yang, Y.; Liu, F.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2017. Sketch-a-net: A deep neural network that beats humans. *IJCV*.

Zhang, K.; Huang, Y.; Song, C.; Wu, H.; and Wang, L. 2015. Kinship verification with deep convolutional neural networks. In *BMVC*.

Zhang, H.; Liu, S.; Zhang, C.; Ren, W.; Wang, R.; and Cao, X. 2016. Sketchnet: Sketch classification with web images. In *CVPR*.

Zhang, K.; Huang, Y.; Du, Y.; and Wang, L. 2017. Facial expression recognition based on deep evolutional spatial-temporal networks. *TIP*.

Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Liu, W.; and Li, H. 2019. Adversarial spatio-temporal learning for video deblurring. *TIP*.