

Stacked Adversarial Network for Zero-Shot Sketch based Image Retrieval

Anubha Pandey¹

anubhap93@gmail.com

Ashish Mishra¹

mishra@cse.iitm.ac.in

Vinay Kumar Verma²

vkverma@iitk.ac.in

Anurag Mittal¹

amittal@cse.iitm.ac.in

Hema A. Murthy¹

hema@cse.iitm.ac.in

¹Indian Institute of Technology Madras

²Indian Institute of Technology Kanpur

Abstract

Conventional approaches to Sketch-Based Image Retrieval (SBIR) assume that the data of all the classes are available during training. The assumption may not always be practical since the data of a few classes may be unavailable, or the classes may not appear at the time of training. Zero-Shot Sketch-Based Image Retrieval (ZS-SBIR) relaxes this constraint and allows the algorithm to handle previously unseen classes during the test. This paper proposes a generative approach based on the Stacked Adversarial Network (SAN) and the advantage of Siamese Network (SN) for ZS-SBIR. While SAN generates a high-quality sample, SN learns a better distance metric compared to that of the nearest neighbor search. The capability of the generative model to synthesize image features based on the sketch reduces the SBIR problem to that of an image-to-image retrieval problem. We evaluate the efficacy of our proposed approach on TU-Berlin, and Sketchy database in both standard ZSL and generalized ZSL setting. The proposed method yields a significant improvement in standard ZSL as well as in a more challenging generalized ZSL setting (GZSL) for SBIR.

1. Introduction

The standard approaches for retrieving related information from a huge database of images are either based on a query image or query text. Retrieval of images using an image-based query is relatively easy compared to that of image retrieval using text-based queries. Text-based queries can be ambiguous, incomplete, and language-dependent. Recent research has shown that instead of text descriptions, sketches can be used as a query. It is more convenient to use sketches as queries since shapes are easy to remember than the textual description. Image retrieval using sketch-based queries is referred to as sketch-based image retrieval (SBIR). [2, 27, 3, 41].

SBIR aims to retrieve the images that belong to a class

using a set of query sketches from the same class. Freehand sketches may magnify the cross-domain discrepancy between sketches and the real-world images as they can vary significantly across persons depending upon the salient features of the image that a person wants to emphasize. In order to make retrieval robust, sketches and their corresponding images are projected to a common subspace [8, 25, 29]. The major issue with this approach is that the method fails to generalize for the test data under the unavailability of accurate sketches, and its performance on unseen classes is poor.

To address these issues recently, Dey et al.[5], Dutta et al. [7], Verma et al. [16] Pandey et al. [24], Shen et al.[31], and Yelamarthi et al.[40] proposed SBIR in Zero-Shot framework(ZS-SBIR).In ZS-SBIR, the training and testing classes are mutually exclusive. Shen et al. [31] in their proposed ZSIH approach, combined zero-shot learning and sketch-based image retrieval using a cross-modal hashing scheme. Dey et al. [5] proposed a ZS-SBIR framework that learns a common embedding space for both the sketch and image domains. [5, 7, 31] use sketch class descriptions[26] as side information along with sketch features for establishing the semantic relationship between the image feature space and sketch feature space. In contrast, Yelamarthi et al.[40] proposed two similar autoencoder-based generative models, CAAE(Conditional Adversarial Autoencoder) [19] and CVAE(Conditional Variational Autoencoder)[33] for zero-shot SBIR without using any side information. One of the major shortcomings of the ZSIH[31] and Doodle to search [5] is that they require sketch class descriptions as side information for learning semantics between the sketches and images. Due to the explosive growth of new categories, it is not practically possible to get class descriptions for every new class. We propose a generative model for SBIR in the zero-shot framework, which shows a significant improvement without using any side information among all the state-of-the-art methods for both the datasets Sketchy[29] and Berlin[8].

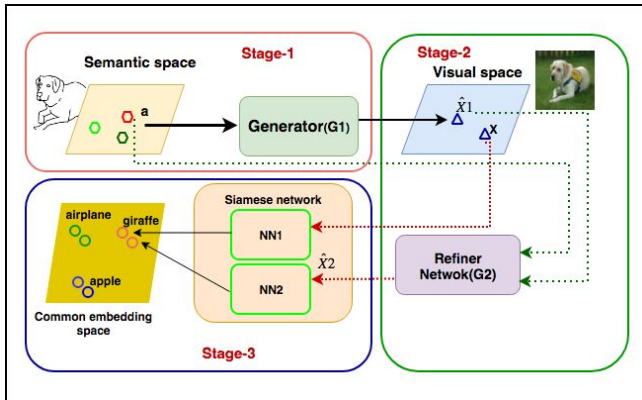


Figure 1. Overview of the proposed approach

Zero-shot learning is categorized into two settings based on the test data. One is standard zero-shot learning (ZSL), which assumes that the seen and unseen classes are mutually exclusive, and the test data comes only from the unseen classes [9, 23]. The other one is generalized zero-shot learning (GZSL), which assumes that the test data may belong to both the seen and unseen classes [1, 14, 35]. GZSL setting is more challenging as compared to the standard ZSL setting. So it is observed that most of the existing approaches are biased towards the seen classes for the GZSL setting. The prior works for ZS-SBIR, ZSIH [31], CVAE [40], Doodle to search [5], JGAN [24] and GZS-SBIR [16] have shown experiments only for standard ZSL setting, whereas our proposed model has shown competitive performances on both the ZSL and GZSL settings.

In this paper, we propose a multistage generative model for the sketch-based image retrieval task in a zero-shot setting. The model is inspired by the StackGAN architecture [44]. The output of the multistage model is fed to the Siamese Network (SN) [4] to learn a better embedding and reduce the Hubness problem [6]. We believe that using multiple stages of GAN, we can generate refined features that are more close to the original image feature space. Further, using Siamese Network [4], we project the generated and real image features into another space where they are more discriminative. The Siamese network uses Contrastive loss function to distinguish between the given pair of generated and real image features in the projected space. This approach helps to reduce the ZS-SBIR problem into multiple subproblems: Stage1- Projection of sketch features to image domain, Stage2- Refinement of generated image features and Stage3- Generation of more distinctive features using Siamese Network. The generative nature of the model enables the synthesis of the pseudo labeled image instances for unseen classes based on sketch features. This approach converts the zero-shot SBIR (ZS-SBIR) problem into a conventional image-to-image retrieval problem. The overview of the proposed method is shown in Figure 1. Our contribu-

tion is summarized below:

- We propose a multi-stage GAN based generative model for zero-shot setting that transforms the zero-shot sketch-based image retrieval (ZS-SBIR) problem to a conventional image-to-image retrieval problem.
- We propose to use a Maximum Mean Discrepancy (MMD) loss [11] in GAN [10] it helps to distinguish between the pairs of real and generated features of images of different classes.
- Unlike the previous approaches for ZS-SBIR [31, 40] that performs a nearest neighbor search in the image space, we use a Siamese Network based on the max-margin loss to learn a better metric for the similarity measured in the projected space, inspired by the prior work Qi et.al [27].
- Our method yields significantly better results in both the standard and generalized zero-shot setting without using any side information (e.g., word2vec based attributes of the classes [20, 26]), as compared to [40].

2. Related Work

In this section, we briefly describe the existing techniques for both SBIR and zero-shot learning. Free hand-drawn sketches fail to capture the complete information of the images; this causes a significant cross-domain gap between the sketch and the image feature space. SBIR tries to learn a shared representation for both the sketches and the images to mitigate the domain gap between the two different spaces. The traditional methods in SBIR, such as [18, 29, 36], used hand-crafted descriptors of sketches and images for retrieval. The conventional deep learn frameworks of SBIR try to project features of sketches and images into a common subspace such that the sketches and images of the same class project close to each other, while the projection of sketches and images of different classes are distant. These projected features are used in the retrieval task. Qi et.al [27] used Siamese architecture and Sangkloy et.al [29] used triplet ranking loss for coarse-grained SBIR. Liu et.al [18] proposed a semi-heterogeneous deep architecture for extracting the binary codes from the sketches and the images, which can be trained in an end-to-end fashion for the coarse-grained SBIR task.

Existing SBIR approaches [18, 27, 29, 36, 42] do not generalize in terms of learning the mapping for unseen sketches, and the corresponding classes. Similarly, state-of-the-art methods for SBIR work well for already seen classes, whereas for any new class, they fail to retrieve the same class images. The capability of zero-shot learning (ZSL) to classify an unseen class example at the test time has received significant attention [1, 9, 14, 23, 32]. ZSL

aims to recognize instances of unseen classes by a transfer of semantic information from seen to unseen classes. There are primarily two different approaches to ZSL.

The first type is embedding-based ZSL. Embedding based approaches [1, 21, 22, 23, 35, 37, 39] address this issue by learning the interaction between visual space and semantic or class attributes space. Based on the direction of the embedding function, they are divided into three sub-categories. The first one learns the embedding from visual space to semantic space. The second approach learns the embedding from the semantic space to the visual space. Both of these approaches suffer from the hubness problem [6], i.e., a small number of objects (hubs) may occur as the nearest neighbor of many categories, resulting in the diminishing of the nearest neighbor method. To address this issue, the third type of approach learns a bilinear embedding function to project both the visual features and class prototypes or semantic features into a shared latent space. This suffers from the domain-shift problem.

The second type of approach is the synthesis-based ZSL [12, 15, 22, 34, 35, 38]. These are recent generative approaches to zero-shot learning. Synthesis-Based ZSL converts the zero-shot learning problem to the traditional supervised learning problem by synthesizing pseudo-labeled data based on class-prototype or semantic description of unseen classes.

Recently [5, 7, 16, 24, 31] and [40] have proposed an approach for sketch-based image retrieval in the zero-shot framework. [31] proposed a hashing based model for ZS-SBIR, [5] has proposed to learn a joint distribution between sketch and image domain, and [40] proposed two models, one is based on conditional variational auto-encoder (CVAE). The second is based on conditional adversarial auto-encoder (CAAE) for the ZS-SBIR task. Also, [5, 7] and [31] use sketch class description as an additional information whereas [40] does not use any side information to train the model. In this paper, we propose a multi-stage conditional generative adversarial network inspired by stackGan architecture [44] followed by a Siamese network for matching. Our model does not use any additional information other than sketch features for zero-shot training similar to [40].

3. Proposed Approach

3.1. Zero-Shot SBIR (ZS-SBIR)

In the zero-shot setting, we partition the dataset into two mutually exclusive sets based on sketch classes: Seen classes(S) and Unseen classes(U) i.e., $S \cap U = \phi$. The train data belongs to the Seen Classes(S). In the Standard ZSL setting, test data belongs to the Unseen Classes(U), and in the Generalized ZSL setting, test data belongs to both the Seen and Unseen Classes. The objective of zero-shot learn-

ing is to train a model that generalizes well for unseen class sketches as well. The mathematical formulation and notations of the ZS-SBIR are given below:

Let $A = \{(\mathbf{x}_i^s, \mathbf{x}_i^{im}, \mathbf{y}_i) | \mathbf{y}_i \in \mathcal{Y}\}$ be the triplet of sketch, image, and the class label. Here \mathcal{Y} is the set of all class labels. We partition the class labels in the data into \mathcal{Y}_{train} and \mathcal{Y}_{test} for the train and test respectively. Let $A_{tr} = \{\mathbf{x}_i^s, \mathbf{x}_i^{im}, \mathbf{y}_i | \mathbf{y}_i \in Y_{train}\}$ and $A_{te} = \{\mathbf{x}_i^s, \mathbf{x}_i^{im}, \mathbf{y}_i | \mathbf{y}_i \in Y_{test}\}$ be the partition of A into train and test sets. We denote sketch feature \mathbf{x}^s with \mathbf{c} and image feature \mathbf{x}^{im} with \mathbf{x} through out this paper for convenience. Another assumption for the ZS-SBIR is $A_{tr} \cap A_{te} = \phi$.

The overall architecture of the proposed system consists of three stages, as described below:

3.2. Stage-1

The first module consists of a Conditional Generative Adversarial Network (CGAN). It takes sketch features and a random vector from the unit Gaussian distribution as input and generates the corresponding class image features. The main task of this module is to generate the image features, conditioned on the same class sketch feature. We call it a generator module.

This module is composed of a generator $G_1 : C \times Z \rightarrow X$ parameterized by θ_{G_1} , a discriminator $D_1 : X \rightarrow [0, 1]$ parameterized by θ_{D_1} and a regressor $R_1 : X \rightarrow C$ parameterized by θ_{R_1} . Where C is a set of conditional attributes (sketch features) and Z is a set of random vectors sampled from a unit Gaussian. Generator G_1 takes as input a sketch feature \mathbf{c} and random vector \mathbf{z} which is sampled from $\mathcal{N}(0, 1)$ and generates the image feature $\hat{\mathbf{X}}_1$ of the same class as that of the sketch. Discriminator D_1 takes input as real image feature \mathbf{X} or generated image feature $\hat{\mathbf{X}}_1$ and attempts to distinguish between real features, and synthesized features. Regressor R_1 acts as a regularizer for the generator G_1 , where it tries to reconstruct the original sketch feature from the generated image feature $\hat{\mathbf{X}}_1$. Regressor R_1 helps the generator G_1 to generate more discriminative and realistic image features. The loss functions used are:

$$L_{rec}(\theta_{G_1}) = \|\mathbf{X} - G_1(\mathbf{c}, \mathbf{z}; \theta_{G_1})\|_2 \quad (1)$$

$$L_{adv}(\theta_{G_1}, \theta_{D_1}) = \log(D_1(\mathbf{X}; \theta_{D_1})) - \log(1 - D_1(G_1(\mathbf{c}, \mathbf{z}; \theta_{G_1}))) \quad (2)$$

$$L_{reg}(\theta_{G_1}, \theta_{R_1}) = \|\mathbf{c} - R_1(G_1(\mathbf{c}, \mathbf{z}; \theta_{G_1}); \theta_{R_1})\| \quad (3)$$

Here L_{rec} is the reconstruction loss, L_{adv} is the adversarial loss and L_{reg} is the regularizer loss. The overall GAN loss is given as :

$$L_{GAN_1}(\theta_{G_1}, \theta_{D_1}, \theta_{R_1}) = L_{rec} + \alpha * L_{adv} + \beta * L_{reg} \quad (4)$$

Here α and β are hyper-parameters. In the proposed approach, instead of pure adversarial loss (Equation 2), we

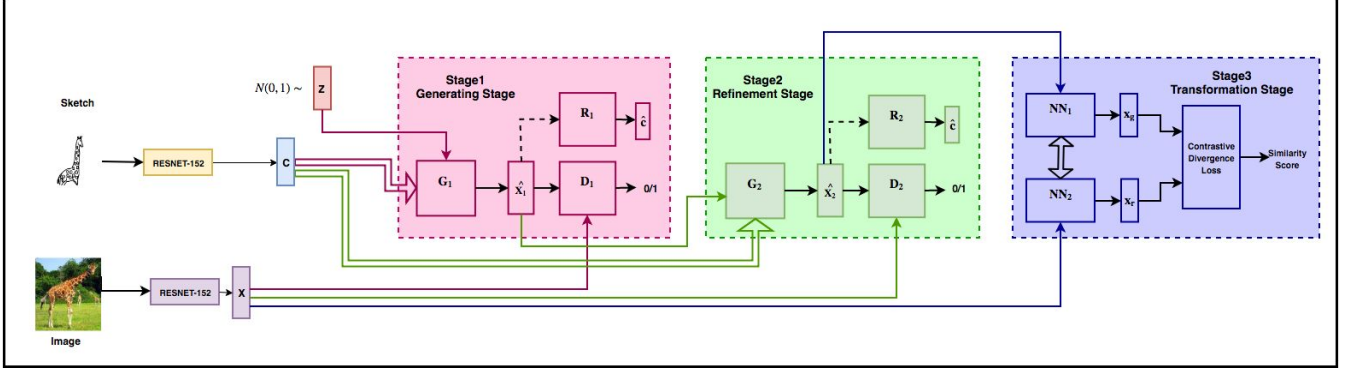


Figure 2. The Training pipeline of our proposed model. The features for images \mathbf{x} and sketches \mathbf{c} are extracted using same pretrained ResNet-152 on ImageNet-1000 dataset.

include the supervised mean square error loss (Equation 1). Empirically we have found that the joint loss given in Equation 4 shows better results than the adversarial loss.

3.3. Stage-2

This module uses an architecture similar to the Stage-1, but the task is to refine the features generated in Stage-1. The StackGAN architecture [44] inspires the combination of Stage-1 and Stage-2, where the first GAN learns to generate high-level features, and the second GAN learns to generate low-level features. Because of the multiple stage refinement, StackGAN generates more realistic images as compared to a single GAN. The Generator G_2 takes the generated feature $\hat{\mathbf{X}}_1$ from the Stage-1 and its corresponding attribute \mathbf{c} as input, and generates the refined feature $\hat{\mathbf{X}}_2$. The Discriminator D_2 takes the real image features \mathbf{X} and the generated image features $\hat{\mathbf{X}}_2$ as input and classifies them as synthetic or real. The Regressor R_2 acts as a regularizer by reconstructing the original attribute using the generated features $\hat{\mathbf{X}}_2$. This regularization step helps G_2 to generate more discriminative features that are close to that of the actual image. The loss functions used are:

$$L_{rec}(\theta_{G_2}) = \|\mathbf{X} - G_2(\mathbf{c}, \hat{\mathbf{X}}_1; \theta_{G_2})\| \quad (5)$$

$$L_{adv}(\theta_{G_2}, \theta_{D_2}) = \log(D_2(\mathbf{X}; \theta_{D_2})) - \log(1 - D_2(G_2(\mathbf{c}, \hat{\mathbf{X}}_1; \theta_{G_2}))) \quad (6)$$

$$L_{reg}(\theta_{G_2}, \theta_{R_2}) = \|\mathbf{c} - R_2(G_2(\mathbf{c}, \hat{\mathbf{X}}_1; \theta_{G_2}); \theta_{R_2})\| \quad (7)$$

We further add a Maximum Mean Discrepancy loss(MMD)[11] in the generator G_2 . The MMD loss is a kernel-based distance function between pairs of synthesized and real samples. Using MMD loss, we project both the synthesized and real image features in a high dimensional space using a kernel function and try to preserve the property of the image class. MMD loss also acts as a regularizer

for generator G_2 to generate more discriminative and similar features to the original class image features. We compute MMD loss between generated image features $\hat{\mathbf{X}}_2$, and real image features \mathbf{X} . Assume \mathbf{x} is real image feature and $\hat{\mathbf{x}}$ is the generated image feature. The overall MMD loss for all N training samples is defined as :

$$L_{img}^{mmd}(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{j=1}^{j=N} \sum_{j'=1}^{j'=N} k(\mathbf{x}_j, \mathbf{x}_{j'}) - 2 \sum_{j=1}^{j=N} \sum_{i=1}^{i=N} k(\mathbf{x}_j, \hat{\mathbf{x}}_i) + \sum_{i=1}^{i=N} \sum_{i'=1}^{i'=N} k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i'}) \quad (8)$$

Here, we use a linear combination of multiple RBF kernels ($k(\mathbf{x}, \hat{\mathbf{x}})$) that is defined as :

$$k(\mathbf{x}, \hat{\mathbf{x}}) = \sum_n \eta_n \exp\left(\frac{-\|\mathbf{x} - \hat{\mathbf{x}}\|^2}{2\sigma_n}\right) \quad (9)$$

where σ_n is the standard deviation and η_n is the weight factor for n^{th} RBF kernel.

The overall GAN loss for stage-2 is defined as:

$$L_{GAN_2}(\theta_{G_2}, \theta_{D_2}, \theta_{R_2}) = L_{rec} + \alpha * L_{adv} + \beta * L_{reg} + \gamma * L_{img}^{mmd} \quad (10)$$

Here α , β and γ are hyper-parameters. The architecture of this stage is similar to that of Stage-1, the only difference is that the generator G_2 takes input as $\mathbf{c}, \hat{\mathbf{X}}_1$ i.e. the original attribute and the reconstructed sample.

3.4. Stage-3

This stage learns the joint embedding space between the generated image features from Stage-2 and the real image features based on class labels. This module consists of a Siamese Network which projects the real image and the synthesized image into a common subspace. The projection is

made in such a way that the same class images are close to each other while the different class images are separated by a margin. Our ultimate goal is to generate image features based on the sketch such that the distribution of generated features should follow the same distribution as the real image features. However, this may not be true always since the domain shift may occur between the synthesized samples and the original samples. To reduce the domain shift, we project the data into a common space. In this module one network takes generated image features $\hat{\mathbf{X}}_2$ as input and second network takes real image features \mathbf{X} as input and tries to learn a projection such that if the generated image feature $\hat{\mathbf{X}}_2$ and real image feature belongs to the same class, the similarity metric should be maximum. Otherwise, the similarity metric should be small. The loss function used in this module is as follows:

First, we define true labels \mathbf{Y}_t

$$\mathbf{Y}_t = \begin{cases} 0 & \text{if } \hat{\mathbf{X}}_2 \text{ and } \mathbf{X} \text{ belongs to the different class;} \\ 1 & \text{if } \hat{\mathbf{X}}_2 \text{ and } \mathbf{X} \text{ belongs to the same class.} \end{cases}$$

False labels \mathbf{Y}_f is defined as : $\mathbf{Y}_f = 1 - \mathbf{Y}_t$

$$\begin{aligned} \mathbf{O}_g &= NN_1(\hat{\mathbf{X}}_2, \theta_N) \\ \mathbf{O}_r &= NN_2(\mathbf{X}, \theta_N) \\ d &= \|\mathbf{O}_r - \mathbf{O}_g\|_2 \end{aligned}$$

$$L_C = \mathbf{Y}_t * d + \mathbf{Y}_f * (\max[(m - d), 0])^2 \quad (11)$$

Here NN_1 and NN_2 are the neural networks from the Siamese Network with shared weight. θ_N is the parameter of the Siamese Network and m is the margin hyperparameter. The projected features \mathbf{O}_r and \mathbf{O}_g correspond to real image features and features generated from G_2 in stage-2 respectively. \mathbf{O}_r and \mathbf{O}_g are used for image retrieval task. d is the Euclidean distance between \mathbf{O}_r and \mathbf{O}_g .

Image retrieval methodology

During the test, we have sketches features of unseen classes. We aim to retrieve the same class images as sketches from an image database. Following are the steps involved in retrieving real images using sketches:

- We pass the sketch features as the conditional variable \mathbf{c} and a random vector \mathbf{Z} to the trained generator G_1 which generates the corresponding image features $\hat{\mathbf{X}}_1$.
- The generated features $\hat{\mathbf{X}}_1$ along with its sketch features are passed to the trained generator G_2 which generates refined features $\hat{\mathbf{X}}_2$.
- Using trained Siamese network projected features \mathbf{O}_f and \mathbf{O}_r are obtained corresponding to the generated features $\hat{\mathbf{X}}_2$ and real image features \mathbf{X} respectively.

- The real images are ranked according to the Euclidean distance $d(\mathbf{O}_f, \mathbf{O}_r)$ for retrieval.

4. Experiments Setting and Results

4.1. Dataset and Visual Feature

We evaluate our proposed model on two widely used datasets for the task of ZS-SBIR: Sketchy [29] and TU-Berlin [8], along with the additional images provided by the [18]. Both the datasets are a collection of sketches and corresponding real images from several different categories.

The visual features for images and sketches are extracted using ResNet-152 [13] network pre-trained on ImageNet-1000 dataset. No fine-tuning was performed. We forward pass the images and sketches in the pre-trained ResNet-152 model and extract 2048-dimensional features from the last fully connected layer. Visual features for the sketch is used as conditioning attributes for our proposed generative model.

4.1.1 Sketchy Dataset(Extended)

The Sketchy dataset [29] contains sketch-image pairs from 125 different categories. Initially, there were 100 images from each category in the dataset. Hand-drawn sketches corresponding to the objects in these 12500 images were collected, resulting in 75471 sketches. Later [18] introduced 60502 more real images from all 125 classes resulting in a total of 73,002 images. We use a test-train split similar to [40] for the Sketchy dataset that contains 104 classes in the train set, and 21 classes in the test set. The split proposed by [40] ensures that none of the classes in the test set are present in the Imagenet-1000 classes. To form the sketch-image pair for training, we randomly select images and sketches from the same class and pair them. We make 1000 such pairs from each class to form the training set.

4.1.2 TU Berlin Dataset(Extended)

TU Berlin [8] (extended) contains 250 different categories of sketches and images. It is a collection of 20000 sketches and 204489 images extended by [18, 43]. We randomly select 30 classes for the test set and the remaining 220 classes for training. The dataset has some classes with large samples and some with only a few. To reduce the bias during training, we sample an equal number of sketches and images from each category. Following [31], during the test, we select only those classes with more than 400 samples. To form the image-sketch pairs for training, we follow the same strategy as the Sketchy dataset.

4.2. Implementation details

Our proposed network has following of 3 stages-
Stage-1: Stage1 consists of a Generator, a Discriminator,

Type	Method	Sketchy Dataset		TU Berlin Dataset	
		Precision@200	mAP@200	Precision@200	mAP@200
SBIR	Baseline	0.176	0.099	0.139	0.083
	Siamese-1 [4]	0.243	0.134	0.127	0.061
	Siamese-2 [27]	0.251	0.149	0.133	0.067
	Fine-Grained Triplet [29]	0.155	0.081	0.086	0.050
	Coarse-Grained Triplet [30]	0.169	0.083	0.128	0.057
ZSL-SBIR	Direct Regression	0.066	0.022	0.117	0.062
	ESZSL[28]	0.187	0.117	0.131	0.072
	DAP [17]	0.078	0.071	0.075	0.067
	SAE [14]	0.238	0.136	0.152	0.084
	CAAE [40]	0.240	0.146	0.159	0.094
	CVAE [40]	0.269	0.159	0.182	0.109
	SAN(Ours)	0.322	0.236	0.218	0.141
GZSL-SBIR	CAAE [40]	0.186	0.124	0.162	0.0912
	CVAE[40]	0.202	0.134	0.177	0.0985
	SAN(Ours)	0.304	0.227	0.203	0.124

Table 1. Precision@200 and mAP@200 results on the traditional SBIR and ZSL method in the ZS-SBIR setup. Note that for a fair comparison, we reproduce the results using the same ResNet-152 features for all the baselines. [40] proposed two models CAAE and CVAE.

and a Regressor Network. We use a series of fully connected (FC) layers in all these networks and apply ReLU after each layer except the last layer. A 300-dimensional noise vector \mathbf{z} , concatenated with a 2048-dimensional conditioning variable \mathbf{c} , is fed into the generator G_1 . The conditioning variables \mathbf{c} is a 2048-dimension features of sketches, obtained from ResNet-152 [13]. G_1 passes the input features through a series of 4 FC layers having 1024, 512, 1024, 2048 neurons respectively, and outputs 2048-dimensional feature vector $\hat{\mathbf{X}}_1$ of the corresponding real image. Discriminator module D_1 tries to distinguish between the features of real images \mathbf{X} , and features generated $\hat{\mathbf{X}}_1$ from G_1 . It takes 2048 dimension feature vectors and passes through a series of 3 FC layers having 1024, 512, and 128 neurons, respectively. It outputs the probability of the features being real. Regressor Network R_1 takes features generated from G_1 and tries to regenerate the features of the conditioning variable \mathbf{c} . It passes the input through a series of 4 FC layers having 1024, 512, 1024, and 2048 neurons, respectively. The output of the network is 2048-dimensional feature vector $\hat{\mathbf{c}}$. We train our network using Adam Optimizer on L_{GAN_1} loss (Equation 4) with learning rate = 0.00001, batch size = 50 keeping hyperparameters $\alpha = 0.01$ and $\beta = 0.0001$. We tune the α and β hyper-parameters via a grid search from 10^{-6} to 10^3 . While training, we first train the discriminator separately for two epochs and then train the entire network end-to-end for L_{GAN_1} loss. We observe that the validation performance saturates after 30 epochs.

Stage-2: The network architecture of this stage is the same as that of Stage-1. The generator G_2 , of this stage, takes the output of G_1 concatenated with a conditioning variable \mathbf{c} and outputs more refined features $\hat{\mathbf{X}}_2$ closer to the real image features than the previous stage. This network is also trained using Adam Optimizer on L_{GAN_2} loss (Equation

10) with learning rate = 0.00001, batch size = 50 keeping hyperparameters $\alpha = 0.01$, $\beta = 0.0001$ and $\gamma = 0.01$. We tune the α , β and γ hyper-parameters via a grid search from 10^{-6} to 10^3 . The training is done in a similar way, as described above for Stage1. We observe that the validation performance saturates after 35 epochs.

Stage-3: This stage uses Siamese Network to find the similarity between the features generated in stage-2, namely, $\hat{\mathbf{X}}_2$ and the features of real image \mathbf{X} . It uses two similar neural networks NN_1 and NN_2 with shared weights to process both the input features. NN_1 and NN_2 has an input FC layer with 1024 neurons followed by a ReLU layer and an output FC layer with two neurons. We minimize the contrastive divergence loss between \mathbf{X}_g and \mathbf{X}_r features obtained by passing input features $\hat{\mathbf{X}}_2$ and \mathbf{X} through NN_1 and NN_2 respectively. We train the network using Adam optimizer on the contrastive divergence loss L_C (Equation 11) setting hyperparameter $\mathbf{m} = 5$ with learning rate 0.01 and batch size 32. We tune the hyper-parameter \mathbf{m} via a grid search from 1 to 100. We train the network for 20 epochs and observe that the validation performance saturates after 15 epochs.

4.3. Comparison with existing methods

We compare our proposed model with the existing state-of-the-art of SBIR, ZSL baselines, and recently proposed ZS-SBIR approaches.

4.3.1 Comparison with SBIR baseline

The baseline models of SBIR includes Siamese-1[4], Siamese-2[27], Fine-Grain Triplet(FGT)[29] and Coarse-grained triplet(CGT)[30]. All the models were built according to the description in the original paper and trained under

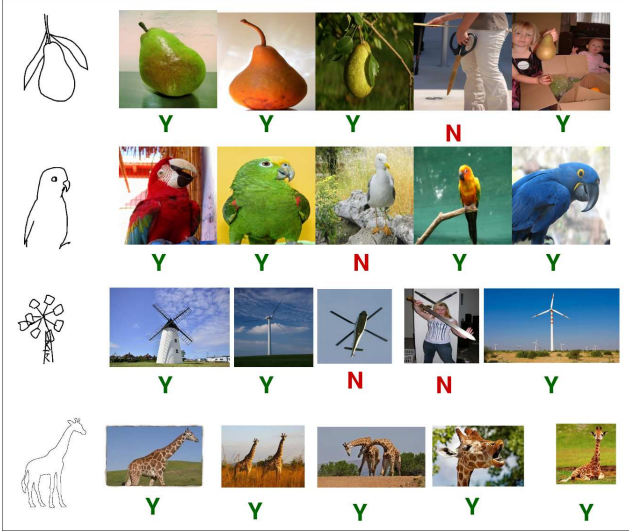


Figure 3. Top 5 Retrieval results of our proposed model. Here we can see that a retrieved object fails when the sketch outline is very close to the image outline. N indicates false-positive retrieval results.

the zero-shot setting. We use the same seen-unseen splits of categories for all the experiments for a fair comparison. A baseline also added for comparison. We take a ResNet-152 network pre-trained on ImageNet-1K as the baseline. The score for a given sketch-image pair is given by the cosine similarity between their ResNet-152 features.

4.3.2 Comparison with ZSL baseline

We select a set of state-of-art zero-shot learning approaches as the benchmark and implement the same for the sketch-based image retrieval task. The selected ZSL algorithms involves Direct Regression, ESZSL[28], DAP[17], and SAE[14]. The Semantic Autoencoder (SAE) proposes an autoencoder framework to encourage the re-constructibility of the sketch vector from the generated image vector. ESZSL[28] learns a bilinear compatibility matrix between images and attribute vectors in the context of zero-shot classification. We adapt the model to the ZS-SBIR task by mapping the sketch features to the image features using labeled training data from the seen classes. In Direct-Regression, the ZS-SBIR task is formulated as a simple regression problem where each image feature vector is predicted from the sketch features. This is similar to the direct attribute prediction method that is a widely used baseline for zero-shot image classification.

4.3.3 Comparison with ZS-SBIR

Recently ZSIH [31], CVAE [40] and Doodle to search [5] methods are proposed for ZS-SBIR. Both these methods [5, 31] use side information(word vector [26] for sketch

classes) along with sketch features to train the model. [40] proposed two generative models, CVAE [40] and CAAE [40] that use only sketches features as a condition to synthesize image features(without using any side information). CAVE[40] and CAAE[40] have performed experiments only on the sketchy dataset in a new split of seen and unseen classes, whereas ZSIH [31] and Doodle to search [5] have shown experiments on both the Berlin and Sketchy datasets. However, all these methods have shown experiments only in the standard zero-shot setting(ZSL). So, for a fair comparison, we compare our proposed model with CVAE[40] and CAAE[40].

4.4. Results and Analysis

From Table 1, we observe that all the SBIR and ZSL baselines are not able to generalize well for unseen class sketches. The reason for their failure is that these methods have been trained in a supervised setting and hence have not used any transfer learning techniques for unseen classes.

For a fair comparison we reproduce the results of CVAE[40] and CAAE[40] for ResNet-152 features on Sketchy(on realistic split) and Berlin(on random split) dataset. We perform experiments in both standard and generalized ZSL settings. We observe that in Standard ZSL setting, our model outperforms CVAE by 5.3%, 7.7%, and 3.6%, 3.2% absolute improvement in precision@200 and mAP@200 in Sketchy and Berlin dataset respectively. For GZSL, we randomly sampled 10% examples per class from seen classes and included with unseen class examples to create test data for our proposed model. Our model outperforms CVAE by 10.2%, 9.3%, and 2.6%, 2.5% absolute improvement in precision@200 and mAP@200 in Sketchy and Berlin dataset respectively. We observe that our model without using any side information outperforms the Doodle to search[5] in the Berlin dataset that uses the sketch class description as side information to train the model.

Figure 3 shows the top-5 retrieval results of our model for sketches of unseen classes. The retrieved images show that our proposed approach is robust for unseen classes, and it learns a better mapping from sketch space to image space.

5. Ablation Analysis

In this section, we show some ablation studies to prove the plausibility of our proposed model. Tables 2 and 3 clearly show the significance of each stage in our proposed model.

Ablation with multi-stage GAN

Our model generates more robust features with two stages $G_1 + G_2$ for unseen classes based on sketches, the improvement of performance in $G_1 + G_2$ over G_1 justifies our claim. With $G_1 + G_2$ there is 2.1%, 1.8% and 1.3%, 1.6% absolute performance improvement in precision@200 and

Berlin Dataset			
Type	Method	Precision@200	mAP@200
G_1	SAN-SBIR(our)	0.170	0.101
$G_1 + G_2$	SAN-SBIR(our)	0.191	0.119
Improvement	With G_2	2.1%	1.8%
$G_1 + G_2$	SAN-SBIR(our)	0.191	0.119
$G_1 + G_2 + \text{MMD}$	SAN-SBIR(our)	0.204	0.128
Improvement	With MMD	1.3%	0.9%
$G_1 + G_2 + \text{MMD}$	SAN-SBIR(our)	0.204	0.128
$G_1 + G_2 + \text{MMD} + P$	SAN-SBIR(our)	0.218	0.141
Improvement	With P	1.4%	1.3%

Table 2. Precision@200 and mAP@200 results of our proposed approach on ZS-SBIR setup for Berlin Dataset. G_1 , G_2 , P, MMD corresponds to Stage-1, Stage-2, Stage-3 and maximum mean discrepancy respectively.

Sketchy Dataset			
Type	Method	Precision@200	mAP@200
G_1	SAN-SBIR(our)	0.284	0.189
$G_1 + G_2$	SAN-SBIR(our)	0.297	0.205
Improvement	With G_2	1.3%	1.6%
$G_1 + G_2$	SAN-SBIR(our)	0.297	0.205
$G_1 + G_2 + \text{MMD}$	SAN-SBIR(our)	0.314	0.218
Improvement	With MMD	1.7%	1.3%
$G_1 + G_2 + \text{MMD}$	SAN-SBIR(our)	0.314	0.218
$G_1 + G_2 + \text{MMD} + P$	SAN-SBIR(our)	0.322	0.236
Improvement	With P	0.8%	1.8%

Table 3. Precision@200 and mAP@200 results of our proposed approach on ZS-SBIR setup for Sketchy dataset. G_1 , G_2 , P, MMD correspond to Stage-1, Stage-2, Stage-3 and maximum mean discrepancy respectively.

mAP@200 for Berlin and Sketchy datasets respectively as compared to only G_1 .

Effect of MMD Loss

Our ablation shows that adding MMD loss in the Generator of the second stage (G_2) has boosted the model performance. The MMD loss enforces the model to maximize the margin between generated samples of a different class, therefore increases the robustness of the retrieval task. We found an absolute improvement of 1.3%, 0.9% and 1.7%, 1.3% in precision@200, and mAP@200 as compare to without using MMD for Berlin and Sketchy datasets respectively.

Ablation with Siamese Network

Hubness may occur on applying the nearest neighbor search on generated features for the task of image retrieval that may degrade the performance of our model. [6] shows that the probability of becoming a hub node is high if we compute the KNN in the original space, whereas if we compute it in a mapped space, the hubness problem reduces as compared to previous one. We address this issue in stage-3 (transformation stage) of our model. In this stage, the features generated by stage2 and the real image features are

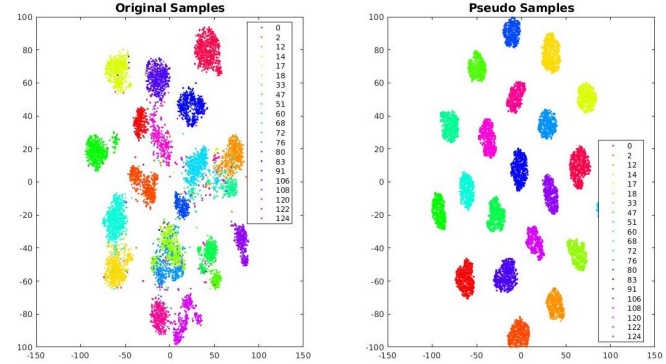


Figure 4. tSNE-Visualization of the original and synthesized samples. We can see the generated samples follow the same distribution as the original one, and the projected features for unseen classes are discriminative and class-wise well separated.

projected to a common space by similarity using a Siamese Network. The projection is made such that the features of the same class are close, while a significant margin separates the features of different classes. This approach provides more class-wise discriminative features. Tables 2 and 3 do establish that the inclusion of stage-3 does improve performance significantly. Including stage-3, the absolute performance of our model improves by 1.4%, 1.3% and 0.8%, 1.8% in precision@200 and mAP@200 as compare to 2 stage model ($G_1 + G_2 + \text{MMD}$) for Berlin and Sketchy datasets respectively. Figure 4 shows the tSNE visualization of original features and synthesized features in projected space, and we can observe that the projected features are well class-wise separated.

6. Conclusion

In this paper, we propose to use a multi-stage GAN based framework called SAN to solve the SBIR problem in a zero-shot setting. The proposed approach uses SAN to synthesize refined image samples from the sketch features and hence reduces the SBIR problem to an image-to-image retrieval problem. The proposed method is based on a multi-stage GAN to synthesize refined samples. The nearest neighbor search technique for the SBIR task suffers from the hubness [6] problem. To address this issue, we project the data to another space using Siamese Network, where hubness has its minimal effect. In the ablation study, we found that all the proposed components (Stage-1, Stage-2, Stage-3) have a significant contribution to improving the performance of the ZS-SBIR task. We perform an extensive experiment on Sketchy and TU-Berlin datasets for the ZS-SBIR in both ZSL and GZSL settings. Our proposed approach shows the state-of-the-art result without using any additional information to train the model.

References

- [1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015.
- [2] X. Cao, H. Zhang, S. Liu, X. Guo, and L. Lin. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *ICCV*, pages 313–320, 2013.
- [3] Y. Cao, C. Wang, L. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. *IEEE*, 2011.
- [4] S. Chopra, R. Hadsell, and Y. LeCun. 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*, 2005.
- [5] S. Dey, P. Riba, A. Dutta, J. Lladós, and Y.-Z. Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [6] G. Dinu, A. Lazaridou, and M. Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.
- [7] A. Dutta and Z. Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5089–5098, 2019.
- [8] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graph.*, 31:44–1, 2012.
- [9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.
- [10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.
- [11] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.
- [12] Y. Guo, G. Ding, J. Han, and Y. Gao. Synthesizing samples for zero-shot learning. *IJCAI*, 2017.
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [14] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.
- [15] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, June 2018.
- [16] V. Kumar Verma, A. Mishra, A. Mishra, and P. Rai. Generative model for zero-shot sketch-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [17] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 36(3):453–465, 2014.
- [18] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, pages 2862–2871, 2017.
- [19] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *ICLR*, 2016.
- [20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [21] A. Mishra, M. Reddy, A. Mittal, and H. A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. *arXiv preprint arXiv:1709.00663*, 2017.
- [22] A. Mishra, V. K. Verma, M. S. K. Reddy, A. Subramaniam, P. Rai, and A. Mittal. A generative approach to zero-shot and few-shot action recognition. *WACV*, pages 372–380, 2018.
- [23] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.
- [24] A. Pandey, A. Mishra, V. Kumar Verma, and A. Mittal. Adversarial joint-distribution learning for novel class sketch-based image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [25] S. Parui and A. Mittal. Similarity-invariant sketch-based image retrieval in large databases. In *ECCV*, pages 398–414, 2014.
- [26] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
- [27] Y. Qi, Y.-Z. Song, H. Zhang, and J. Liu. Sketch-based image retrieval via siamese convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2460–2464. *IEEE*, 2016.
- [28] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015.
- [29] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35:119, 2016.
- [30] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.
- [31] Y. Shen, L. Liu, F. Shen, and L. Shao. Zero-shot sketch-image hashing. In *CVPR*, June 2018.
- [32] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.
- [33] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.
- [34] V. K. Verma, D. Brahma, and P. Rai. A meta-learning framework for generalized zero-shot learning. *arXiv preprint arXiv:1909.04344*, 2019.
- [35] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *ECML-PKDD*, pages 792–808, 2017.
- [36] F. Wang, L. Kang, and Y. Li. Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR*, abs/1504.03504, 2015.
- [37] W. Wang, Y. Pu, V. K. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin. Zero-shot learning via class-conditioned deep generative models. *AAAI*, 2018.

- [38] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. 2018.
- [39] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *IJCV*, 123(3):309–333, 2017.
- [40] S. K. Yelamathi, S. K. Reddy, A. Mishra, and A. Mittal. A zero-shot framework for sketch-based image retrieval. *ECCV*, 2018.
- [41] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C.-C. Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.
- [42] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision*, 122(3):411–425, 2017.
- [43] H. Zhang, S. Liu, C. Zhang, W. Ren, R. Wang, and X. Cao. Sketchnet: Sketch classification with web images. In *CVPR*, pages 1105–1113, 2016.
- [44] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017.