Journal of *Imaging*

MDPI

Article

Zero-Shot Sketch-Based Image Retrieval Using StyleGen and Stacked Siamese Neural Networks

Venkata Rama Muni Kumar Gopu * and Madhavi Dunna

Department of Electrical, Electronics and Communication Engineering (EECE), Gitam School of Technology, Gitam Deemed to be University, Rushikonda, Visakhapatnam 530045, India; mdunna@gitam.edu
* Correspondence: mgopu@gitam.in

Abstract: Sketch-based image retrieval (SBIR) refers to a sub-class of content-based image retrieval problems where the input queries are ambiguous sketches and the retrieval repository is a database of natural images. In the zero-shot setup of SBIR, the query sketches are drawn from classes that do not match any of those that were used in model building. The SBIR task is extremely challenging as it is a cross-domain retrieval problem, unlike content-based image retrieval problems because sketches and images have a huge domain gap. In this work, we propose an elegant retrieval methodology, StyleGen, for generating fake candidate images that match the domain of the repository images, thus reducing the domain gap for retrieval tasks. The retrieval methodology makes use of a two-stage neural network architecture known as the stacked Siamese network, which is known to provide outstanding retrieval performance without losing the generalizability of the approach. Experimental studies on the image sketch datasets TU-Berlin Extended and Sketchy Extended, evaluated using the mean average precision (mAP) metric, demonstrate a marked performance improvement compared to the current state-of-the-art approaches in the domain.

Keywords: sketch-based image retrieval; SSiNN; stacked Siamese neural network; domain gap; ZS-SBIR

Citation: Gopu, V.R.M.K.; Dunna, M. Zero-Shot Sketch-Based Image Retrieval Using StyleGen and Stacked Siamese Neural Networks. *J. Imaging* **2024**, *10*, 79. https://doi.org/10.3390/jimaging10040079

Academic Editor: Nikolaos Mitianoudis

Received: 10 February 2024
Revised: 21 March 2024
Accepted: 24 March 2024
Published: 27 March 2024

1. Introduction

Sketch-based image retrieval (SBIR) [1–4] constitutes a specific subset within the wider spectrum of content-based image retrieval (CBIR) problems. Content-based image retrieval (CBIR) refers to the process of retrieving relevant images from a large database based on the contents and intent of the input query images rather than manually entered metadata or keywords. In SBIR, the system is presented with ambiguous sketches as input queries and is tasked with retrieving corresponding matches from a database composed of natural images.

The evolution of SBIR has closely paralleled advancements in image processing [5,6], computer vision, and machine learning. Originating in the late 1990s as an offshoot of CBIR, early SBIR systems focused on basic shape matching using simple feature extraction techniques [7]. The early 2000s saw the integration of more sophisticated feature descriptors like SIFT and HOG [2], enhancing the accuracy of sketch-to-image matching. The advent of deep learning, particularly with convolutional neural networks (CNNs), in the 2010s marked a significant milestone, drastically improving the ability to bridge the domain gap between sketches and natural images. Vehicle re-identification [8] and person re-identification [9] are closely related to image recognition and play a crucial role in security, surveillance, and traffic management applications. Recent developments have delved into zero-shot learning and cross-domain retrieval, pushing the boundaries of SBIR in handling diverse and unseen sketch categories. Today, SBIR [10,11] continues to evolve, integrating with emerging technologies and expanding its application scope.

Image retrieval using sketches as query input is gaining more practical significance, particularly with the rise of touch-based devices and the increasing demand for accessibility

features. A practical application of SBIR involves utilizing free-hand sketches as search queries, particularly when textual search methods are impractical or when language barriers are present. Highlighting the versatility of SBIR techniques, the following are some key applications:

- Law enforcement and forensic art [12]: In law enforcement, SBIR can help match sketches of suspects or missing persons with photographic databases.
- E-commerce and online retail [13]: SBIR can be used in online shopping platforms to allow users to sketch an item they wish to purchase. This is particularly useful when shoppers are unsure of the technical name of the item but can draw it.
- Digital art and graphic design [14]: Artists and designers can use SBIR to find reference images based on a rough sketch. This is useful in creative processes where visualizing an idea is easier through drawing than describing it in words.
- Education and research [15]: In educational settings, SBIR can assist students and researchers in finding scientific diagrams or historical images based on hand-drawn sketches. This can be particularly useful in fields like archaeology, history, or biology.
- Medical imaging [11]: SBIR can be used in medical diagnostics by allowing doctors to sketch symptoms or conditions and retrieve similar medical images or case studies. This could be particularly useful in dermatology or radiology.
- Cultural heritage and museums [16]: Museums and cultural institutions can use SBIR to help visitors connect with artworks or artifacts. Visitors could sketch an artifact or art piece they are interested in and receive information about similar items in the museum's collection.
- Architecture and interior design [17]: Architects and interior designers can use SBIR to find building designs, interior decor ideas, or furniture based on sketches. This can streamline the process of translating conceptual sketches into concrete plans or finding matching furniture and decorations.

SBIR presents a significant challenge because of the substantial domain gap between the sketch input and image database, thereby transforming it into a cross-domain image retrieval issue. Sketches are notably abstract and devoid of key features typically leveraged in traditional content-based image retrieval, such as shape features, color attributes, texture, and structural properties. This absence of features further compounds the complexity of SBIR, rendering it a more intricate task compared to standard CBIR [18,19] activities. Delving a bit deeper into specific challenges and limitations faced by SBIR, Different people have different sketching styles, and even the same person may sketch differently at different times. This variability can lead to inconsistencies in how objects are represented, making it difficult for SBIR systems to accurately match sketches with images. There is a significant domain gap between the high-dimensional data of photographs and the low-dimensional, abstract nature of sketches. Sketches can be symbolic or abstract [20], not always representing real-world objects accurately or realistically. This abstraction poses a challenge in matching these sketches with real images, especially when the sketches represent conceptual ideas rather than concrete objects. Sketches may not always maintain consistent scale or orientation relative to the actual objects they represent. An SBIR system needs to be robust to such variations, which adds complexity to the retrieval process. For SBIR systems to be practical, especially in commercial applications, they need to offer real-time or near-real-time performance. Processing sketches and searching through large image databases efficiently are computationally demanding tasks that require optimized algorithms and hardware. Understanding the semantic meaning of a sketch [21] (what object or concept it represents) is a complex task. This is particularly challenging when sketches are vague or when the same sketch could represent multiple objects. An SBIR system trained on one dataset may not perform well on another due to differences in image types, sketch styles, and object categories. Ensuring that these systems generalize well across different datasets is a significant challenge. In realistic scenarios, as databases expand to include new image categories, an SBIR system may lack prior information about these novel classes. The task of retrieving such images, which were not represented in

the original system design, is referred to as zero-shot sketch-based image retrieval (ZS-SBIR) [22–24]. This task introduces an additional layer of complexity due to the inherent knowledge gap associated with zero-shot samples, posing an even greater challenge to the system. The fundamental challenge in zero-shot learning [25] is the absence of training examples for unseen classes. The model must infer knowledge about these classes from the data it has, which can be difficult if the unseen classes are significantly different from the seen ones. The model must generalize from the seen classes to the unseen ones. This requires an understanding of underlying patterns and features that are common across classes, which is a complex task, especially when the unseen classes diverge significantly from the seen ones. Bridging the semantic gap between low-level features extracted from data and high-level class concepts is challenging. The model must understand and utilize abstract, semantic relationships without having direct examples of those relationships. Models trained on a specific set of classes may develop biases toward those classes, leading to poor performance when encountering new classes. This domain shift is a significant hurdle in ensuring that the model performs well on both seen and unseen classes. Models need to be specific enough to accurately categorize seen classes but also general enough to adapt to unseen classes. Finding this balance is challenging, as overfitting to the seen classes can reduce the model's ability to generalize. As the number of classes increases, the computational complexity can grow significantly. Ensuring that the model scales efficiently and maintains performance with a growing number of classes is a challenge.
Our contributions:

1. Propose a novel approach for the ZS-SBIR problem through the segregation of domain gap reduction and image-retrieval stages.
2. Propose the mathematical formulation of the StyleGen approach, illustrating various loss functions involved in training an effective model for domain gap reduction between sketches and images.
3. Presents the neural network architectures for the StyleGen model, which comprises generator and discriminator blocks.
4. Provides an adaption of the latest SSiNN [26] architecture for image retrieval to maximize the overall system's retrieval performance.
5. Presents the datasets used in the experimental study and the performance metrics used for the evaluation and comparison with the existing approaches.
6. Presents a comprehensive presentation of experimental results, illustrating the effectiveness of our approach in ZS-SBIR scenarios.

2. Related Work

In this section, we will provide a brief overview of the research literature related to SBIR (sketch-based image retrieval), ZSL (zero-shot learning), and ZS-SBIR.

2.1. Sketch-Based Image Retrieval

SBIR approaches primarily focus on addressing the challenges associated with the domain gap between sketches and images. They aim to develop techniques that can effectively capture and represent the visual features of sketches and images, devise matching algorithms to measure their similarity, and design retrieval strategies to accurately retrieve relevant images based on user-provided sketches.

Ming Zh et al. [27] proposed a gradually focused bilinear attention model to improve fine-grained image retrieval based on sketches by accurately highlighting representative local positions and using weighted bilinear coding for more discriminative feature representations. Deep network architectures for sketch-based image retrieval (SBIR) employing convolutional neural networks (CNNs) with multi-stage regression are proposed and evaluated in the research by Bui et al. [28]. The authors investigate their networks' capacity to generalize across multiple object categories using minimal training data, as well as methodologies for weight sharing, preprocessing, data augmentation, and dimensionality reduction. They describe a hybrid multi-stage training network for improving performance

by combining contrastive and triplet networks. The work by Zhou et al. [29] proposed a deep learning approach that dealt with sketch data scarcity by incorporating a method for sketch augmentation that generates additional sketches from existing data by removing, adjusting, and rotating strokes. The proposed approach utilizes a multi-domain learning technique that employs a couple of Siamese CNNs that pair 2D shape images and sketches in conjunction with a joint Bayesian metric to maximize inter-class similarity and minimize intra-class similarity. The study by Niteesh et al. [30] proposed image preprocessing and deep learning-based methods to fix sketch-based image retrieval (SBIR) systems that do not have enough semantic knowledge. The Canny edge detection method makes a binary image of the edges of the natural image. CNN models that are built on ImageNet pull out deep features. Rocchio's method is used to provide relevant feedback for gap identification.

2.2. Zero-Shot Learning

Zero-shot learning is a method that aims to recognize unseen categories by using a shared visual-semantic function. The paper by Xian et al. [31] addresses the need for a unified benchmark in zero-shot learning and proposes a new benchmark by defining evaluation protocols and data splits. It emphasizes the importance of comparable and reliable results in the field. The study by Li et al. [32] presents a zero-shot learning strategy that simultaneously learns visual prototypes and maintains semantic consistency across visual and semantic domains, yielding much better outcomes.

2.3. Zero-Shot Sketch-Based Image Retrieval

ZS-SBIR refers to the task of retrieving relevant images from a database using a sketch as a query when there are no examples of that specific class in the training set. The GTZSR framework [33] employs a graph transformer to maintain the semantic space class topology while transmitting the visual space's class context graph. It attempts to narrow the domain gap between image and sketch features by minimizing the Wasserstein distance between them. The ACNet framework [23] employed a two-module approach in which a retrieval module directs the synthesis module to produce a variety of images that eventually converge to the domain of the photos. The paper by Dutta et al. [24] introduced "StyleGuide", a unique retrieval method for ZS-SBIR that employs style-guided fake-image generation.

3. Proposed StyleGen for ZS-SBIR

In this paper, we put forward an innovative technique for ZS-SBIR using a combination of StyleGen and a stacked Siamese Neural network (SSiNN). Our method leverages the power of generative adversarial networks (GANs) to synthesize photographic images from sketches and then employs a stacked Siamese network to perform efficient image retrieval. A thorough evaluation of our methodology is conducted using benchmark datasets, wherein it is shown to outperform current techniques. Our findings highlight the potential of our method to revolutionize the field of sketch-based image retrieval and open up new avenues for research in this area. In Figure 1, we present a high-level block diagram that provides a schematic representation of the proposed StyleGen approach, elucidating its core components and their interrelationships.

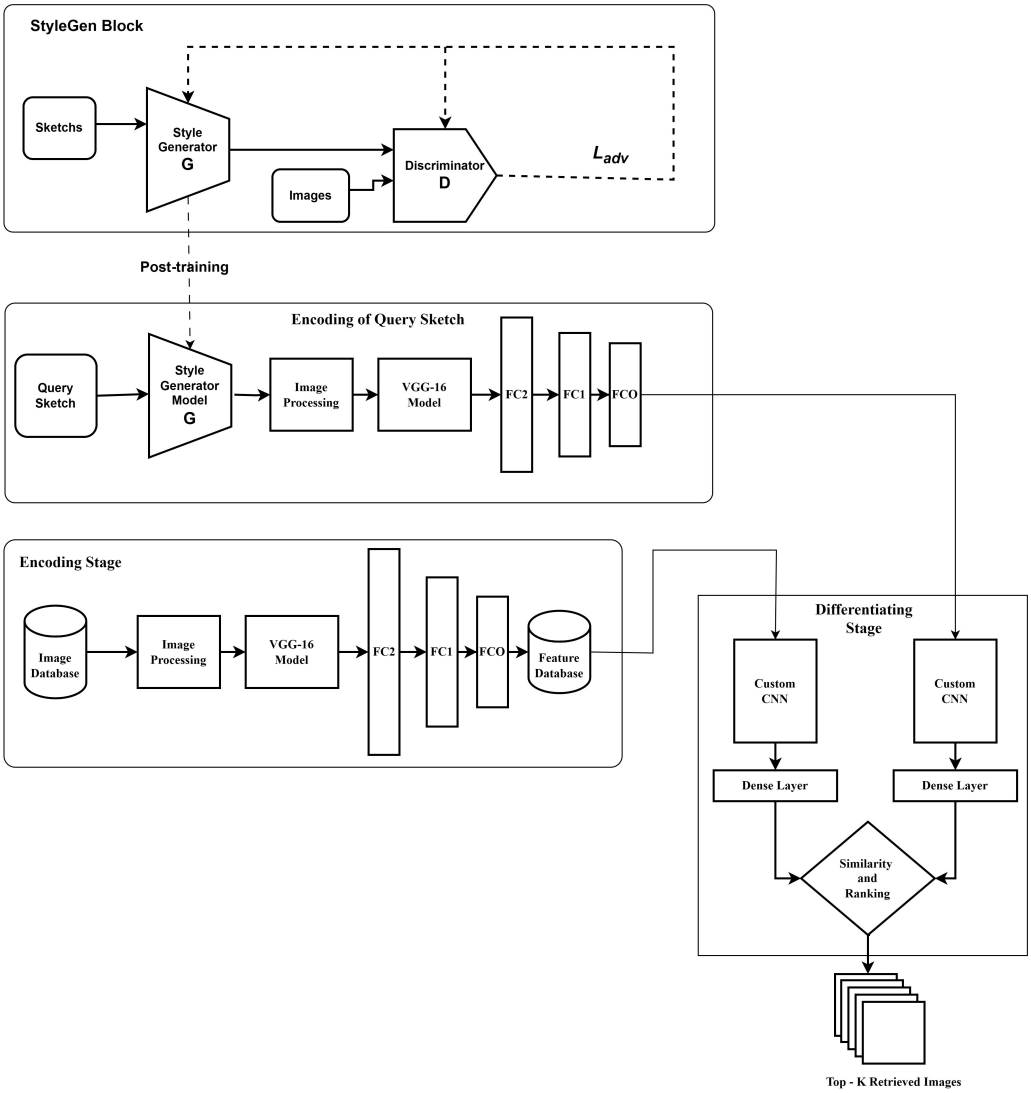

Figure 1. The StyleGen block contains a GAN network, which is trained to generate images equivalent to the input sketches in the domain of the images. The trained generator model is fed with the query sketch to generate its corresponding StyleGen image. This image is encoded using the encoder block of the SSiNN and is differentiated against the encoded versions of the database images. The differentiating stage ranks and furnishes the top K relevant images.

3.1. Problem Formulation

The problem formulation for dividing the dataset into training and test datasets for ZS-SBIR can be expressed as follows: Given a dataset consisting of sketches and corresponding images, the goal is to partition the dataset into two subsets, D_{train} and D_{test}, such that D_{train} contains a set of sketches and corresponding images from a subset of the classes in the dataset. These sketches and images are used to train the ZS-SBIR system. D_{test} contains the remaining set of sketches and corresponding images from the remaining classes in D. These sketches are used as queries to evaluate the performance of the ZS-SBIR methodology. The partitioning of the dataset should be done in such a way that the classes in D_{train} and D_{test} are disjoint, i.e., no class should appear in both sets. Moreover, the partitioning should ensure that there is a sufficient number of examples from each class in both D_{train} and D_{test} to ensure a fair evaluation of the system's performance. The quality of the partitioning can have a significant impact on the accuracy of the system's performance and, therefore, should be carefully considered during system development.

The methodology can be divided into two phases, each addressing a sub-problem of the overall ZS-SBIR. Phase I focuses on domain gap reduction between the sketches and images. The effectiveness of this phase determines the overall efficacy of the system, as it minimizes the discrepancy between the feature space representations of sketches and images. Phase II focuses on the actual feature space representation, which effectively encodes the images and sketches so as to maximize the retrieval performance.

Let Y denote the set encompassing all labels present within the dataset. Partition Y in to 2 disjoint sets Y_{train} and Y_{test}. Hence

$$Y_{train} \cap Y_{test} = \phi \tag{1}$$

$$Y_{train} \cup Y_{test} = Y \tag{2}$$

Let D_{train} and D_{test} are the Train and Test datasets, respectively. Both training and test sets are a combination of sets of images and sketches.

$$D_{train} = X_{train} \cup S_{train} \tag{3}$$

$$D_{test} = X_{test} \cup S_{test} \tag{4}$$

3.2. No Pair Assumption

In the realm of ZS-SBIR, the "no pair assumption" signifies that during the training phase of the retrieval system, there are not any directly corresponding or "paired" sketches and images available. The no-pair assumption is fundamental in many SBIR methods, particularly in zero-shot SBIR, where the goal is to retrieve images of classes not seen during training. This assumption allows the SBIR system to learn a mapping from the sketch space to the image space that can be generalized to unseen classes. If paired sketches and images were used during training, the SBIR system could simply learn to memorize the paired examples rather than learning a generalizable mapping between the sketch and image spaces. This would result in poor performance in novel classes during testing. An example of paired and unpaired cases is illustrated in Figure 2.

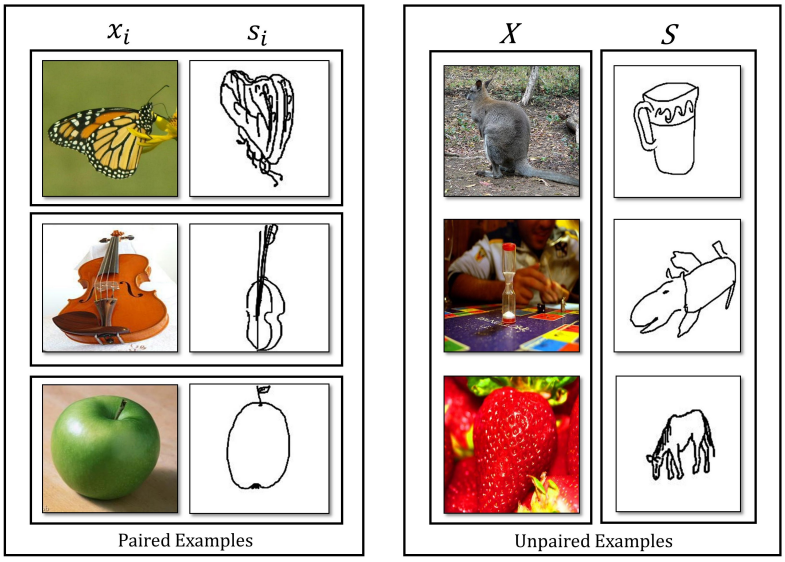

Figure 2. The image is divided into two distinct sections. On the left side, there are "paired examples", which consist of images and sketches that are presented in matching or corresponding pairs, demonstrating a relationship or connection between them. Each pair could be similar in appearance, function, or concept, indicating a clear link or duality. On the right side, the "unpaired examples" are displayed. These consist of images and sketches that stand alone without an apparent match or counterpart. They may be varied in appearance, function, or concept, lacking the obvious pairing found in the examples on the left.

3.3. Neural StyleGen for Domain Gap Reduction

StyleGen is the process of transferring the domain characteristics of images onto the sketches while preserving their intents. We will use generative neural network models to learn the domain style and content representations of the images and then infuse them into the sketch intent to produce a generated image with the desired style and domain. We use CycleGAN architecture [34] for the StyleGen part of the ZS-SBIR system proposed. CycleGAN architecture learns the features of one set of images and applies them to another set of images without the need for paired training examples. CycleGAN uses two sets of GANs, one for each direction of translation, each consisting of a generator network that maps images from one domain to another and a discriminator network that attempts to distinguish between the generated images and the real images in the target domain. Collaboratively, the generator and discriminator networks undergo adversarial training, during which the generator attempts to generate near-real images with the intention of deceiving the discriminator. On the other hand, the discriminator endeavors to accurately differentiate between real and generated images.

3.4. StyleGen Approach

Let X_{train} and S_{train} be the image set and sketch set, respectively, from the training data.

Definition 1. *The StyleGen function can be defined as SG, which transforms the sketch images into image-like StyleGen images.*

$$\{\forall s_i \in S_{train}, \exists\, \mathrm{SG} \textit{ such that } \mathrm{sg_i} = \mathrm{SG}(s_i)\} \tag{5}$$

where $\mathrm{SG} : S_{train} \rightarrow X_{train}$

For simplicity, let us denote X as the image set and S as the sketch set. Let G be the generator function, which transforms sketches into images. The purpose of G is to take the samples from the sketch database S and to generate corresponding StyleGen images.

$$G : S \rightarrow X \tag{6}$$

Let G' be the generator function, which does the transformation in the reverse direction.

$$G' : X \rightarrow S \tag{7}$$

The aim of the learning functions above can be defined as follows:

$$G'(G(s)) \approx s \tag{8}$$

$$G(G'(x)) \approx x \tag{9}$$

Let D be the discriminator function; we define two discriminator functions, D_s and D_x. The purpose of these functions is to distinguish real and generated images in their respective domains. Discriminators in CycleGAN provide feedback to the generators and help them fine-tune themselves, thus enhancing the generated images' quality. The discriminator acts as a sort of "adversary" that challenges the generators to produce better and more realistic images. The discriminator, D_s, differentiates between s and the transformed images, $G'(x)$, whereas D_x differentiates x and $G(s)$.

3.5. Adversarial Loss

The adversarial loss is the objective function used to train the GAN. It is based on the idea that the generator network should be able to produce synthetic data that is indistinguishable from real data. The adversarial loss is evaluated as the cross-entropy of the discriminator's output relative to the true label, where the true label is 1 for real data and 0 for synthetic data. In other words, the generator network attempts to minimize the

adversarial loss by generating synthetic data that maximally fool the discriminator into believing that it is real. The discriminator network, on the other hand, attempts to maximize the adversarial loss by correctly classifying real and synthetic data. Overall, the adversarial loss plays a crucial role in the GAN architecture, as it drives the competition between the generator and discriminator networks and encourages the generator to produce high-quality synthetic data. Let the data distribution of sketch data be $q(s)$, and that of image data be $p(x)$.

$$s \sim q(s) \tag{10}$$

$$x \sim p(x) \tag{11}$$

$$L_{\text{adv}}(G, D_X) = \mathbb{E}_p[\ln D_X(x)] + \mathbb{E}_q[\ln(1 - D_X(G(s)))] \tag{12}$$

$$L_{\text{adv}}(G', D_S) = \mathbb{E}_q[\ln D_S(s)] + \mathbb{E}_p[\ln(1 - D_S(G'(x)))] \tag{13}$$

The overall adversarial loss function is as follows:

$$L_{\text{adv}} = L_{\text{adv}}(G, D_X) + L_{\text{adv}}(G', D_S) \tag{14}$$

3.6. Cycle Consistency Loss

The second component is the cycle consistency loss, which ensures that the generator produces images that can be transformed back to the original domain without losing information. It is defined as follows:

$$L_{\text{cyc}}(G, G') = \mathbb{E}_q\left[\left|\left|G'(G(s)) - s\right|\right|\right] + \mathbb{E}_p\left[\left\|G(G'(x)) - x\right\|\right] \tag{15}$$

3.7. Identity Loss

The identity loss ensures that the generator network is able to produce outputs that are not only realistic but also retain some of the original characteristics of the input data. By incorporating identity loss into the GAN training process, the generator is encouraged to produce outputs that are both realistic and retain the original properties of the input data. This can improve the quality and fidelity of the generated data and can also help mitigate the mode collapse problem, where the generator produces only a limited set of outputs.

$$L_{\text{identity}} = \mathbb{E}_x[|G(x) - x|_1] + \mathbb{E}_s\left[\left|G'(s) - s\right|_1\right] \tag{16}$$

where:

- $G(.)$ denotes the sketch-to-image generator function.
- x indicates an image sample from the image dataset.
- $\mathbb{E}_x$ denotes the expected value, evaluated over image data distribution.
- $G'(.)$ denotes the image-to-sketch generator function.
- s indicates a sketch sample from the sketch dataset.
- $\mathbb{E}_s$ denotes the expected value, evaluated over sketch data distribution.

3.8. Overall Objective Function

The overall combined objective function is as follows:

$$L_{\text{objective}} = L_{\text{adv}} + \alpha L_{\text{cyc}}(G, G') + \beta L_{\text{identity}} \tag{17}$$

Here, α and β are hyperparameters, where α denotes cycle-consistency loss weight and β denotes identity loss weight.

For our experiments: $\alpha = 10$ and $\beta = 0.5$.

3.9. Network Architecture of Generator

The generator is implemented using a deep convolution network for transforming input images into corresponding target representations. Our network initiation incorpo-

rates a 2D convolution layer featuring a 7×7 kernel, Instance normalization, and ReLU activation follow. As we delve deeper into the network, layers are structured to systematically augment the channel depth and concurrently down-sample the spatial dimensions. This is achieved through 3×3 convolutional filters, together with a stride of 2. These downsampling blocks are complemented with instance normalization and ReLU activation functions. We implement two such downsampling blocks in the generator design. Central to our design is the incorporation of eight residual blocks. Every block is structured with a pair of convolutional layers, both furnished with 3×3 kernels and further enhanced by instance normalization and ReLU activations. A defining characteristic of these blocks is the summation of the incoming input with the processed output, enabling the network to discern and adapt to residual functions. In the advanced segments of our generator, we employed transposed convolutions to methodically revert the downsampling operations. This ensures a progressive recovery of spatial resolutions while concurrently tapering the channel depth. Culminating our design, the final output generation is entrusted to a 7×7 convolutional layer, which is succeeded by a *tanh* activation, thus ensuring that all output values adhere to the range $[-1, 1]$. The network architecture and the parameters are presented in Table 1.

Table 1. Network topology of the generator network.

Layer	Block	Configuration	Channels	Output Shape	Parameters
Convolution2D	Input Block	Filters: 64 Kernel Size: 7×7 Padding: (3,3), Reflect Stride: 1	In: 3 Out: 8	$8 \times 224 \times 224$	1184
InstanceNorm ReLU Activation	Input Block	none	In: 8 Out: 8	$8 \times 224 \times 224$	0
Convolution2D	Down Sampling Block 1	Filters: 128 Padding: (1,1) Kernel Size: 3×3 Stride: 2	In: 8 Out: 16	$16 \times 112 \times 112$	1168
InstanceNorm ReLU	Down Sampling Block 1	none	In: 16 Out: 16	$16 \times 112 \times 112$	0
Convolution2D	Down Sampling Block 2	Filters: 256 Kernel Size: 3×3 Padding: (1,1) Stride: 2	In: 16 Out: 32	$32 \times 56 \times 56$	4640
InstanceNorm ReLU	Down Sampling Block 2	none	In: 32 Out: 32	$32 \times 56 \times 56$	0
Convolution2D	Residual Block 1	Filters: 256 Kernel Size: 3×3 Padding: (1,1) Stride: 1	In: 32 Out: 32	$32 \times 56 \times 56$	9248
InstanceNorm ReLU	Residual Block 1	none	In: 32 Out: 32	$32 \times 56 \times 56$	0
Convolution2D	Residual Block 1	Filters: 256 Kernel Size: 3×3 Padding: (1,1), Reflect Stride: 1	In: 32 Out: 32	$32 \times 56 \times 56$	9248
InstanceNorm	Residual Block 1	none	In: 32 Out: 32	$32 \times 56 \times 56$	0

Table 1. *Cont.*

Layer	Block	Configuration	Channels	Output Shape	Parameters
⋮	Residual Blocks 2 - 8	⋮	⋮	⋮	⋮
ConvTranspose 2D	Upsampling Block 1	Filters: 128 Kernel Size: 3 × 3 Padding: (1,1) Stride: 2	In: 32 Out: 16	16 × 112 × 112	4624
ConvTranspose 2D	Upsampling Block 2	Filters: 64 Kernel Size: 3 × 3 Padding: (1,1) Stride: 2	In: 16 Out: 8	8 × 224 × 224	1160
InstanceNorm ReLU	Upsampling Block 2	none	In: 8 Out: 8	8 × 224 × 224	0
Convolution 2D	Output Block	Filters: 3 Kernel Size: 7 × 7 Padding: (3,3), Reflect Stride: 1	In: 8 Out: 3	3 × 224 × 224	1179
Tanh	Output Block	none	In: 3 Out: 3	3 × 224 × 224	0

3.10. Network Architecture of Discriminator Neural Network

The discriminator network adapted from [35] comprises a series of convolutional layers, progressively increasing in channels from 3 to 512. A combination of LeakyReLU activation and instance normalization was adopted across layers for stability and performance. The topology of the network is presented in Table 2.

Table 2. Network topology of the discriminator network.

Layer	Kernel Size	Stride	Channels	Output Shape	Activation	Parameters
InstanceNorm ReLU	Upsampling Block 1	none	In: 16 Out: 16	16 × 112 × 112	0	
Conv 2D	4 × 4	2	In: 3 & Out: 64	112 × 112 × 64	LeakyReLU	392
Conv 2D	4 × 4	2	In: 64 & Out: 128	56 × 56 × 128	LeakyReLU	2064
Conv 2D	4 × 4	2	In: 128 & Out: 256	28 × 28 × 256	LeakyReLU	8224
Conv 2D	4 × 4	1	In: 256 & Out: 512	28 × 28 × 512	LeakyReLU	32,832
Conv 2D	4 × 4	1	In: 512 & Out: 1	28 × 28 × 1	LeakyReLU	1025

3.11. Training the Networks of StyleGen Phase

The StyleGen framework employs the generator and discriminator networks described in previous sections. For the image retrieval application, we are exclusively concerned with the forward process of transforming sketches into images. The inverse process—from images back to sketches—is not required because the major goal of this process is to combat the domain gap problem, thereby restyling sketches into the image domain, which can then be used for image retrieval tasks. The dataset is partitioned using the approach specified in the current section's problem formulation sub-section. The partition specifics for the

datasets utilized will be expanded on in the experimental results section. Once the training is complete, the generator model can be used for transforming the sketches to the image domain. The generator model, which we call the StyleGen model, is used to generate images equivalent to the query sketches. By doing this, the SBIR task would now be boiled down to a CBIR task where the input query is an image.

3.12. Stacked Siamese Neural Network for Image Retrieval

We use a CBIR technique from [26] for the retrieval task. SSiNN is a two-stage CBIR system that uses a pre-trained model customized to the dataset of study for encoding the input images and uses a Siamese neural network on the encoded images to differentiate and rank the database images for the effective retrieval of the intended images. To train the SSiNN, we use the image sub-set from the training partition of the partitioned dataset. To train the first stage of the SSiNN, we employ VGG-16 architecture and use the image sub-set of the training data. To be precise, the overall dataset is partitioned into training and test datasets, with disjoint classes as described in the partition strategy. From the training dataset, we only use the image sub-set and exclude the sketch sub-set for the training process. To train the second stage, the model of the first stage is used to encode the training dataset, and the encoded vectors are used to train the second stage.

1. Input the StyleGen model with the query sketch to obtain the image domain equivalent representation.
2. To extract the latent space representations of the database Images, run them through the first stage of the SSiNN.
3. Run the StyleGen output from step 1 through the first stage of the SSiNN to acquire the sketch's latent space representation.
4. Now, pass the sketch representation through one input of the Siamese neural network and the latent space representations from the database through the other input of the Siamese neural network.
5. Rank the outputs and provide the top-K images corresponding to the representations as the SBIR system's output.

Simplified Decision-Making Process

In this subsection, we present a simplified overview of the decision-making process. The intricacies of network training and the detailed architecture have been extensively covered in previous sections with a higher degree of technical specificity. Additionally, to maintain a streamlined representation, this subsection does not delve into the optimizations employed, such as the storage of latent space representations in the database.

- The SSiNN model, employed for the retrieval operation, is a two-input model. Henceforth, this model shall be designated as the retrieval model.
- The first input channel of the retrieval model is ingested with images from the database, from which relevant images are to be extracted.
- The second input channel of the retrieval model is allocated for processing the image representation derived from the sketch-based query.
- The transformation of the sketch into its image representation is facilitated by the StyleGen model.
- Subsequently, the retrieval model computes similarity metrics across the dataset images, ranking them based on these scores. Images attaining the highest similarity metrics are identified as the most relevant matches to the input sketch query.

4. Experimental Results

This section provides an exposition of the experimental outcomes obtained from the employed methodology. We evaluate the effectiveness of our approach using two benchmark datasets: "Sketchy Extended" and "TU-Berlin Extended". We present the retrieval performance and compare it to existing approaches.

4.1. Datasets

The "Sketchy Extended" dataset comprises a total of 75,481 sketches and 73,002 photos, including 60,502 images from the ImageNet dataset and 12,500 sourced from [14]. With 125 categories, the dataset encompasses a diverse array of everyday objects, animals, vehicles, and more. To conduct experiments on ZS-SBIR, the dataset is partitioned [22], such that the data sourced from ImageNet, distributed over 21 classes, are allocated for testing, while the remaining data from other classes are utilized for training purposes.

The "TU-Berlin Extended" dataset, as described in [36], consists of 20,000 sketches across 250 object classes. It also includes 204,070 photo images furnished by Liu et al. [3]. The dataset is partitioned into training and test sets under the zero-shot setting by utilizing the partitioning protocol illustrated in [37]. For testing, 30 classes are randomly selected, ensuring each class has a minimum of 400 photo images, and the rest of the classes are used for training purposes.

The choice to utilize the TU-Berlin and Sketchy Extended datasets was made after thoughtful consideration, keeping in mind their rich diversity, the wide array of categories, and the varied drawing styles they encompass. These characteristics are pivotal for assessing the generalization ability of ZS-SBIR systems in a robust manner. Moreover, the dataset, with its comprehensive set of photo-sketch pairs, presents an unparalleled opportunity to delve into cross-modal retrieval challenges within a controlled yet demanding environment, which is crucial for pushing the boundaries of zero-shot learning research. The decision was driven by the objective of thoroughly evaluating our model's adaptability and effectiveness across diverse and rigorous conditions, thereby ensuring its reliability and practical utility in scenarios reflective of the real world. Leveraging these datasets allows for meaningful benchmarking against the current state-of-the-art, contributing significantly to the advancement of knowledge in the field of ZS-SBIR.

4.2. Evaluation Metric

To assess the performance of ZS-SBIR systems in this research, we utilize the mean average precision (mAP) as our evaluation metric. The mAP offers a single scalar value representing the overall average precision across different queries. It is calculated by first determining the average precision (AP) for each sketch query and then computing the mean of these AP values. The mAP effectively encapsulates the system's overall retrieval performance, making it a reliable and widely accepted metric for such evaluations.

Average Precision (AP): The average precision for a query is the mean of the precision scores obtained for each relevant item in the retrieved list, indicating the precision of the system at the rank of that item.

$$\text{AP} = \frac{\sum_{k=1}^{N} P(k) \times \text{r}(k)}{\text{Number of Relevant Items}} \tag{18}$$

where

- $P(k)$ signifies precision at rank k.
- N is the cardinality of the retrieved list.
- $\text{r}(k) = \begin{cases} 1 & \text{if the item is relevant,} \\ 0 & \text{otherwise.} \end{cases}$

mAP@all (Mean average precision at all ranks): This metric computes the mean of the average precision (AP) scores across all queries, with each AP score calculated using the entire ranked list of retrieved items. Its formula is given by the following:

$$\text{mAP@all} = \frac{\sum_{q=1}^{|Q|} \text{AP}_q}{|Q|} \tag{19}$$

where:

- $|Q|$ is the cardinality of the query list.
- AP_q is the average precision for the qth query calculated over the entire retrieved list.

Average Precision at K (AP@K): This is the precision calculated at the K^{th} rank in the retrieved list, specifically considering only the top-K items. It is a useful measure in scenarios where the focus is on the relevance of the top part of the ranked list. The formula is as follows:

$$\mathrm{AP@K} = \frac{\sum_{k=1}^{K} P(k) \times \mathrm{r}(k)}{\min(K, \text{Number of relevant items})} \tag{20}$$

- K is the predefined number of top items to consider in the list.
- $P(k)$ represents the precision at rank k.
- $\mathrm{r}(k) = \begin{cases} 1 & \text{if the item is relevant,} \\ 0 & \text{otherwise.} \end{cases}$

mAP@K (Mean average precision at top-K ranks): This metric calculates the mean of the AP@K scores across all queries, providing a single measure that summarizes the effectiveness of a retrieval system at ranking relevant items within the top-K positions of the ranked list. It is defined as follows:

$$\mathrm{mAP@K} = \frac{\sum_{q=1}^{|Q|} \mathrm{AP@K}_q}{|Q|} \tag{21}$$

where:

- $|Q|$ is the cardinality of the query list.
- $\mathrm{AP@K}_q$ denotes the average precision at K for the q^{th} query.

4.3. Performance Comparison

The proposed methodology of ZS-SBIR, which employs the combination of StyleGen and SSiNN, is benchmarked against the current state-of-the-art approaches employed in the domain through extensive experimental evaluations. The experimental results, in comparison with existing approaches, for the TU-Berlin dataset, are presented in Table 3, while those for the Sketchy Extended dataset are detailed in Table 4. Our results clearly indicate that the ZS-SBIR method leveraging StyleGen and SSiNN exhibits superior performance metrics when compared to its contemporaries. The elevated effectiveness of our approach is evident, establishing a new performance benchmark in ZS-SBIR for the aforementioned datasets. Some of the retrieval results are presented in Figures 3 and 4 for Sketchy Extended and TU-Berlin extended datasets, respectively.

Robustness of the approach: The datasets employed contain images and sketches that are notably diverse within each category. This diversity encompasses a wide range of drawing styles, levels of detail, and artistic interpretations, providing a robust foundation for evaluating the effectiveness of our method across varied real-world scenarios.

Handling of zero-shot learning: Based on the experimental findings, it is evident that our method outperforms current techniques in terms of performance. This enhanced performance can be attributed to two primary factors. Firstly, the implementation of the no-pair assumption within the StyleGen component significantly contributes to the model's ability to generalize effectively, enabling the accurate generation of StyleGen images from previously unseen sketches. Secondly, the application of the stacked Siamese neural network (SSiNN) has been finely tuned to excel with zero-shot samples, further bolstering our method's efficacy.

Table 3. Performance comparison of StyleGen plus the SSiNN methodology compared to existing approaches for the TU-Berlin dataset.

Approach	mAP@all	AP@100
Zero-shot sketch image hashing [37]	22.0	29.1
Content style decomposition [38]	25.4	35.5
Semantically tied paired cycle consistency [39]	29.3	39.2
OCEAN [40]	33.3	46.7
Domain smoothing network [41]	48.1	58.6
Progressive domain-independent feature decomposition network [42]	48.3	60.0
Norm-guided adaptive visual embedding [43]	49.3	60.7
Relationship-preserving knowledge distillation [44]	48.6	61.2
ACNet [23]	57.5	65.8
Proposed approach	59.4	66.3

Table 4. Performance comparison between the StyleGen plus SSiNN methodology and the existing approaches for the Sketchy Extended dataset.

Approach	mAP@200
Conditional variational autoencoder [22]	22.5
Content style decomposition [38]	35.8
Doodle [45]	47.0
Semantic-aware knowledge preservation [46]	49.7
Relationship-preserving knowledge distillation [44]	50.2
ACNet [23]	51.7
Proposed approach	52.8

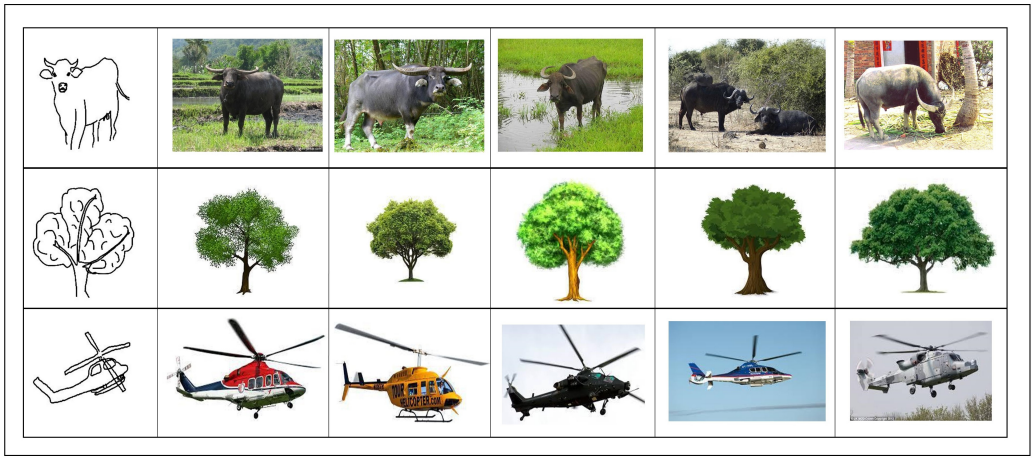

Figure 3. Retrieval results for the Sketchy Extended dataset. The image to the left is the input query sketch and the rest of the images are the retrieved images.

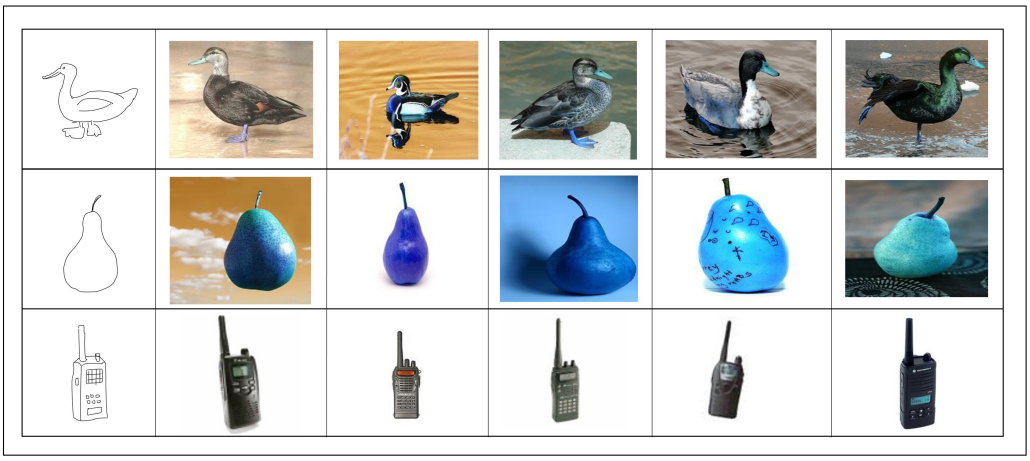

Figure 4. Retrieval results for the TU-Berlin extended dataset. The image to the left is the input query sketch and the rest of the images are the retrieved images.

Estimation of computation time: The proposed methodology is implemented using the PyTorch framework [47] on a Windows PC with Intel Core I7 and Nvidia Geforce RTX. The models are built and optimized for 10 epochs, with Adam as the optimizer and a learning rate of 1×10^{-3} for the generator and 1×10^{-4} for the discriminator. The number of parameters in the generator network is 32,451, and that of the discriminator is 44,537. The time taken for running one epoch is an average of 8 h. So, the time taken for the one overall training cycle with 10 epochs is 80 h. In model training, it is often necessary to run through all training epochs multiple times to achieve optimal performance and robustness. This repetitive process allows the model to continually refine its parameters, learn from the dataset's variability, and explore different solutions, improving generalization and stability. Considering this, the time taken for the overall model building will be in multiples of 80 h. Our approach to addressing the zero-shot sketch-based image retrieval (ZS-SBIR) problem by dividing it into two distinct stages—each focusing on a specific aspect of the challenge—is a strategic method that contributes to its superior performance. The following is a deeper analysis of why this method outperforms others, along with its limitations and potential areas for improvement:

Strengths and reasons for superior performance: Targeted problem-solving approach—in this approach, the ZS-SBIR problem is divided into two stages, with each focusing on a specific challenge—domain gap and knowledge gap. This allows for specialized strategies tailored to each aspect, potentially leading to more effective solutions.

- Stage 1—Domain gap solution with StyleGen framework: The first stage employs the StyleGen framework to specifically address the domain gap problem. By transforming sketches into a style more akin to the target images, we enhance feature compatibility, improving retrieval accuracy.
- Stage 2—Knowledge-gap solution with SSiNN: In the second stage, we utilize the stacked Siamese neural network (SSiNN) to tackle the knowledge gap problem.
- Separate optimization: By separately optimizing each stage, our approach achieves a higher degree of fine-tuning for each specific challenge, contributing to overall superior performance.

The overall performance of the proposed approach depends on the performance of the individual stages. Let P_{overall} be the overall system precision, P_{stylegen} be the precision of the StyleGen stage, and P_{ssinn} be the precision of the SSiNN stage. The performance of StyleGen in CBIR tasks [26] is as high as 99% for the CIFAR-10 dataset; however, for the datasets used in this work, it is around 94%, particularly attributed to the diverse nature

of the classes in these datasets. A simplified estimate of the overall system efficacy can be represented as the product of the individual stages' efficacies.

$$P_{\text{overall}} = P_{\text{stylegen}} \times P_{\text{ssinn}} \tag{22}$$

Despite the high performance of the retrieval block, the overall precision of the framework, as measured by mAP@all for the TU-Berlin extended dataset, is 59.4%, and for the Sketchy Extended dataset, it is 52.8%, as measured by mAP@200. The reason for the drop in the overall precision is attributed to the image equivalent representation conducted by the StyleGen block. The reason for this is the ambiguous nature of the sketches as discussed in the introduction section.

Limitations and Areas for Improvement:

- Complexity and resource intensity: The proposed two-stage process, while effective, is complex and resource-intensive compared to single-stage methods, which could be a limitation in terms of computational efficiency and practicality.
- Integration and cohesion between stages: Ensuring seamless integration and effective cohesion between the two stages is crucial. Any misalignment could potentially reduce the overall effectiveness.
- Cross-dataset generalization: Testing and refining our method on a broader range of datasets is a key focus, aiming to improve its generalizability and applicability to different real-world scenarios.
- Interpretability: While the approach demonstrates impressive performance in matching sketches to images, understanding the decision-making process of these models remains a challenge. This lack of transparency can be problematic, especially in applications where understanding the reasoning behind each match is crucial.

4.4. Ablation Studies

4.4.1. Hyperparameter Selection

This subsection delves into the selection of the hyperparameters α and β from Equation (17), crucial for optimizing our model's performance. It outlines the rationale and experimental process behind choosing specific values for these parameters. The choice of $\alpha = 10$ is based on the CycleGAN paper [34]. The parameter β is chosen based on experimental choice; experiments were conducted with three choices of $\beta = \{0.5, 5, 10\}$. Based on the experimental results in Table 5, 0.5 was the better choice.

Table 5. Experimental results for the selection of hyperparameters for the datasets.

α	β	Sketchy Extended mAP@200	TU-Berlin Extended mAP@all
10.0	10.0	47.7	55.6
10.0	5.0	49.5	59.1
10.0	0.5	52.8	59.4

4.4.2. Effectiveness of Identity Loss Function

The purpose of identity loss is to ensure that when an input from the target domain is provided to a generator, the output is identical or very close to the input, thereby preserving the original identity of the input in the absence of a domain shift. In this experiment, the effectiveness of the identity loss function in the overall performance is measured. The results demonstrate that the presence of identity loss marginally enhances the overall performance. For the TU-Berlin dataset, mAP@all increases from 58.8% to 59.4%, and for the Sketchy Extended dataset, mAP@200 increases from 52.3% to 52.8% when the identity loss function is considered in the overall loss function. The results are presented in Table 6.

Table 6. Experimental results for the assessment of the effectiveness of identity loss function.

Dataset	Metric	Without $L_{identity}$	With $L_{identity}$
TU-Berlin Extended	mAP@all	58.8	59.4
Sketchy Extended	mAP@200	52.3	52.8

4.4.3. Retrieval Block Selection

To optimize this second stage, we conducted experiments with two distinct approaches for the retrieval block: one leveraging an autoencoder [48] and the other utilizing SSiNN [26]. The comparative analysis of these approaches, as detailed in our results in Table 7, clearly demonstrates that the SSiNN-based retrieval method significantly outperforms the autoencoder-based method.

Table 7. Experimental results for the selection of the retrieval block.

Dataset	Metric	Autoencoder	SSiNN
TU-Berlin Extended	mAP@all	46.1	59.4
Sketchy Extended	mAP@200	39.9	52.8

5. Conclusions

In conclusion, our research successfully introduces a novel technique for ZS-SBIR that harnesses the potential of StyleGen and SSiNN (stacked Siamese neural networks). This approach has been empirically validated to outperform existing methods, marking a significant advancement in the field of content-based image retrieval. By ingeniously integrating the generative capabilities of StyleGen with the discriminative prowess of SSiNN, our method not only enhances the accuracy of zero-shot retrieval but also enriches the interpretability of the results. Our method effectively bridges the gap between sketches and photos, even in the absence of paired instances.

Author Contributions: Conceptualization, V.R.M.K.G. and M.D.; methodology, V.R.M.K.G. and M.D.; software, V.R.M.K.G.; validation, V.R.M.K.G. and M.D.; formal analysis, V.R.M.K.G. and M.D.; investigation, V.R.M.K.G. and M.D.; resources, V.R.M.K.G. and M.D.; writing—original draft preparation, V.R.M.K.G.; writing—review and editing, V.R.M.K.G.; visualization, V.R.M.K.G. and M.D.; supervision, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The original data presented in the study are openly available at https://dl.acm.org/doi/10.1145/2897824.2925954 (accessed on 1 October 2023), https://cybertron.cg.TU-Berlin.de/eitz/projects/classifysketch/ (accessed on 1 October 2023) and https://www.image-net.org/ (accessed on 1 October 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Eitz, M.; Hildebrand, K.; Boubekeur, T.; Alexa, M. Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors. *IEEE Trans. Vis. Comput. Graph.* **2010**, *17*, 1624–1636. [CrossRef] [PubMed]
2. Hu, R.; Collomosse, J. A Performance Evaluation of Gradient Field HOG Descriptor for Sketch Based Image Retrieval. *Comput. Vis. Image Underst.* **2013**, *117*, 790–806. [CrossRef]
3. Liu, L.; Shen, F.; Shen, Y.; Liu, X.; Shao, L. Deep Sketch Hashing: Fast Free-Hand Sketch-Based Image Retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2862–2871.
4. Song, J.; Yu, Q.; Song, Y.-Z.; Xiang, T.; Hospedales, T.M. Deep Spatial-Semantic Attention for Fine-Grained Sketch-Based Image Retrieval. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5551–5560.
5. Nyemeesha, V.; Ismail, B.M. Implementation of Noise and Hair Removals from Dermoscopy Images Using Hybrid Gaussian Filter. Network Model. *Anal. Health Inform. Bioinform.* **2021**, *10*, 1–10.

6. Ismail, B.M.; Reddy, T.B.; Reddy, B.E. Spiral Architecture Based Hybrid Fractal Image Compression. In Proceedings of the 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, India, 9–10 December 2016; pp. 21–26.
7. Belongie, S.; Malik, J.; Puzicha, J. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [CrossRef]
8. Zhu, W.; Peng, B. Manifold-Based Aggregation Clustering for Unsupervised Vehicle Re-identification. *Knowl.-Based Syst.* **2022**, *235*, 107624. [CrossRef]
9. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C. Deep Learning for Person Re-identification: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 2872–2893. [CrossRef] [PubMed]
10. Sain, A.; Bhunia, A.K.; Chowdhury, P.N.; Koley, S.; Xiang, T.; Song, Y.-Z. CLIP for All Things Zero-Shot Sketch-Based Image Retrieval, Fine-Grained or Not. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2765–2775.
11. Kobayashi, K.; Gu, L.; Hataya, R.; Mizuno, T.; Miyake, M.; Watanabe, H.; Takahashi, M.; Takamizawa, Y.; Yoshida, Y.; Nakamura, S.; et al. Sketch-Based Semantic Retrieval of Medical Images. *Med. Image Anal.* **2024**, *92*, 103060. [CrossRef] [PubMed]
12. Jain, A.K.; Klare, B.; Park, U. Face Matching and Retrieval in Forensics Applications. *IEEE Multimed.* **2012**, *19*, 20. [CrossRef]
13. Cao, Y.; Wang, C.; Zhang, L.; Zhang, L. Edgel Index for Large-Scale Sketch-Based Image Search. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 761–768. [CrossRef]
14. Sangkloy, P.; Burnell, N.; Ham, C.; Hays, J. The Sketchy Database: Learning to Retrieve Badly Drawn Bunnies. *ACM Trans. Graph.* **2016**, *35*, 1–12. [CrossRef]
15. Hu, R.; Barnard, M.; Collomosse, J. Gradient Field Descriptor for Sketch Based Retrieval and Localization. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 1025–1028.
16. Collomosse, J.; Bui, T.; Wilber, M.J.; Fang, C.; Jin, H. Sketching with Style: Visual Search with Sketches and Aesthetic Context. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2660–2668.
17. Li, B.; Yuan, J.; Ye, Y.; Lu, Y.; Zhang, C.; Tian, Q. 3D Sketching for 3D Object Retrieval. *Multimed. Tools Appl.* **2021**, *80*, 9569–9595. [CrossRef]
18. Madhavi, D.; Mohammed, K.M.C.; Jyothi, N.; Patnaik, M.R. A Hybrid Content Based Image Retrieval System Using Log-Gabor Filter Banks. *Int. J. Electr. Comput. Eng. (IJECE)* **2019**, *9*, 237–244. [CrossRef]
19. Madhavi, D.; Patnaik, M.R. Genetic Algorithm-Based Optimized Gabor Filters for Content-Based Image Retrieval. In *Intelligent Communication, Control and Devices: Proceedings of ICICCD 2017*; Springer: Singapore, 2018; pp 157–164.
20. Bhunia, A.K.; Yang, Y.; Hospedales, T.M.; Xiang, T.; Song, Y.-Z. Sketch Less for More: On-the-Fly Fine-Grained Sketch-Based Image Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9779–9788.
21. Bhunia, A.K.; Koley, S.; Khilji, A.F.U.R.; Sain, A.; Chowdhury, P.N.; Xiang, T.; Song, Y.-Z. Sketching without Worrying: Noise-Tolerant Sketch-Based Image Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 999–1008.
22. Yelamarthi, S.K.; Reddy, S.K.; Mishra, A.; Mittal, A. A Zero-Shot Framework for Sketch Based Image Retrieval. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 300–317.
23. Ren, H.; Zheng, Z.; Wu, Y.; Lu, H.; Yang, Y.; Shan, Y.; Yeung, S.-K. ACNet: Approaching-and-Centralizing Network for Zero-Shot Sketch-Based Image Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 5022–5035. [CrossRef]
24. Dutta, T.; Singh, A.; Biswas, S. StyleGuide: Zero-Shot Sketch-Based Image Retrieval Using Style-Guided Image Generation. *IEEE Trans. Multimed.* **2020**, *23*, 2833–2842. [CrossRef]
25. Zhang, L.; Xiang, T.; Gong, S. Learning a Deep Embedding Model for Zero-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2021–2030
26. Kumar, G.V.R.M.; Madhavi, D. Stacked Siamese Neural Network (SSiNN) on Neural Codes for Content-based Image Retrieval. *IEEE Access* **2023**, *11*, 77452–77463. [CrossRef]
27. Zhu, M.; Chen, C.; Wang, N.; Tang, J.; Bao, W. Gradually Focused Fine-Grained Sketch-Based Image Retrieval. *PLoS ONE* **2019**, *14*, e0217168. [CrossRef] [PubMed]
28. Bui, T.; Ribeiro, L.; Ponti, M.; Collomosse, J. Sketching Out the Details: Sketch-Based Image Retrieval Using Convolutional Neural Networks with Multi-Stage Regression. *Comput. Graph.* **2018**, *71*, 77–87. [CrossRef]
29. Zhou, W.; Jia, J.; Jiang, W.; Huang, C. Sketch Augmentation-Driven Shape Retrieval Learning Framework Based on Convolutional Neural Networks. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 3558–3570. [CrossRef] [PubMed]
30. Kumar, N.; Ahmed, R.; Honnakasturi, V.B.; Kamath, S.S.; Mayya, V. Sketch-Based Image Retrieval Using Convolutional Neural Networks Based on Feature Adaptation and Relevance Feedback. In Proceedings of the International Conference on Emerging Applications of Information Technology, Online, 13–14 Novemver 2021; pp. 103–113.
31. Xian, Y.; Schiele, B.; Akata, Z. Zero-Shot Learning-The Good, the Bad and the Ugly. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4582–4591.
32. Li, X.; Fang, M.; Li, H.; Wu, J. Zero Shot Learning Based on Class Visual Prototypes and Semantic Consistency. *Pattern Recognit. Lett.* **2020**, *135*, 368–374. [CrossRef]

33. Gupta, S.; Chaudhuri, U.; Banerjee, B.; Kumar, S. Zero-Shot Sketch Based Image Retrieval Using Graph Transformer. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 1685–1691.
34. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2223–2232.
35. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
36. Eitz, M.; Hays, J.; Alexa, M. How Do Humans Sketch Objects? *ACM Trans. Graph.* **2012**, *31*, 1–10. [CrossRef]
37. Shen, Y.; Liu, L.; Shen, F.; Shao, L. Zero-Shot Sketch-Image Hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3598–3607.
38. Dutta, T.; Biswas, S. Style-Guided Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the 30th British Machine Vision Conference (BMVC), Cardiff, UK, 9–12 September 2019; Volume 2.
39. Dutta, A.; Akata, Z. Semantically Tied Paired Cycle Consistency for Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5089–5098.
40. Zhu, J.; Xu, X.; Shen, F.; Lee, R.K.W.; Wang, Z.; Shen, H.T. Ocean: A Dual Learning Approach for Generalized Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
41. Wang, Z.; Wang, H.; Yan, J.; Wu, A.; Deng, C. Domain-Smoothing Network for Zero-Shot Sketch-Based Image Retrieval. *arXiv* **2021**, arXiv:2106.11841.
42. Xu, X.; Yang, M.; Yang, Y.; Wang, H. Progressive Domain-Independent Feature Decomposition Network for Zero-Shot Sketch-Based Image Retrieval. *arXiv* **2020**, arXiv:2003.09869.
43. Wang, W.; Shi, Y.; Chen, S.; Peng, Q.; Zheng, F.; You, X. Norm-Guided Adaptive Visual Embedding for Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 19–27 August 2021; pp. 1106–1112.
44. Tian, J.; Xu, X.; Wang, Z.; Shen, F.; Liu, X. Relationship-Preserving Knowledge Distillation for Zero-Shot Sketch Based Image Retrieval. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 5473–5481.
45. Dey, S.; Riba, P.; Dutta, A.; Llados, J.; Song, Y.-Z. Doodle to Search: Practical Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–19 June 2019; pp. 2179–2188.
46. Liu, Q.; Xie, L.; Wang, H.; Yuille, A.L. Semantic-Aware Knowledge Preservation for Zero-Shot Sketch-Based Image Retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3662–3671.
47. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver BC Canada, 8–14 December 2019; pp. 8026–8037.
48. Öztürk, Ş. Stacked Auto-Encoder Based Tagging with Deep Features for Content-Based Medical Image Retrieval. *Expert Syst. Appl.* **2020**, *161*, 113693. [CrossRef]