A Recipe for Efficient SBIR Models: Combining Relative Triplet Loss with Batch Normalization and Knowledge Distillation

Omar Seddati
ISIA Lab (UMONS)
omar.seddati@umons.ac.be

Nathan Hubens
ISIA Lab (UMONS)
nathan.hubens@umons.ac.be

Stéphane Dupont
MAIA (UMONS)
stéphane.dupont@umons.ac.be

Thierry Dutoit
ISIA Lab (UMONS)
thierry.dutoit@umons.ac.be

Abstract

Sketch-Based Image Retrieval (SBIR) is a crucial task in multimedia retrieval, where the goal is to retrieve a set of images that match a given sketch query. Researchers have already proposed several well-performing solutions for this task, but most focus on enhancing embedding through different approaches such as triplet loss, quadruplet loss, adding data augmentation, and using edge extraction. In this work, we tackle the problem from various angles. We start by examining the training data quality and show some of its limitations. Then, we introduce a Relative Triplet Loss (RTL), an adapted triplet loss to overcome those limitations through loss weighting based on anchors similarity. Through a series of experiments, we demonstrate that replacing a triplet loss with RTL outperforms previous state-of-the-art without the need for any data augmentation. In addition, we demonstrate why batch normalization is more suited for SBIR embeddings than l_2-normalization and show that it improves significantly the performance of our models. We further investigate the capacity of models required for the photo and sketch domains and demonstrate that the photo encoder requires higher capacity than the sketch encoder, which validates the hypothesis formulated in [34]. Then, we propose a straightforward approach to train small models, such as ShuffleNetv2 [22] efficiently with a marginal loss of accuracy through knowledge distillation. The same approach used with larger models enabled us to outperform previous state-of-the-art results and achieve a recall of 62.38% at $k = 1$ on The Sketchy Database [30].

Keywords: Sketch-based image retrieval, Triplet Networks, Knowledge Distillation, ShuffleNet

1. Introduction

Sketch-Based Image Retrieval (SBIR) is a fundamental task in multimedia retrieval, where the goal is to retrieve images that match a given sketch query. During the last few decades, the rapid growth in digital media has spurred great interest in multimedia retrieval solutions like SBIR. With the widespread use of touchscreen devices in our daily lives, SBIR solutions have become well-suited for various applications. For instance, an SBIR solution can be integrated into an e-commerce system, where the user draws a sketch to find a specific product. Sketching offers the user a powerful way to convey details beyond the product's category, including global product design and detailed patterns and their actual spatial configuration, which can be difficult to communicate using a text-based query.

Despite the obvious advantages of SBIR solutions, there are several challenges that the computer vision community is still working to overcome. These challenges are related to the abstract nature of sketches, and the gap between natural images and sketches that requires efficient cross-domain features. In addition, the complexity of sketching and sketching quality assessment make new database creation complex and time-consuming. In recent years, researchers have proposed various solutions to address these challenges through more efficient training pipelines, transfer learning, and data augmentation. In this work, we tackle several aspects of the SBIR problem with the aim of identifying a recipe that would provide practical improvements beneficial to any SBIR application. To achieve our goal, we require a recipe that enables SBIR models to reach high accuracy, and that provides enough flexibility when it comes to choosing the models' size to meet different application requirements (e.g. applications running on devices with limited computing power and storage capacity).

We observe two major limitations of current SBIR so-

lutions, severely hindering their performance. In particular, those observations are (1) Data Unreliability, i.e. there exist multiple instances of the same photo that are so similar that they cannot be differentiated by sketches (which we will refer to as *Ambiguous Samples*); (2) Cross-Domain Misalignment, i.e. there exists a discrepancy in the embedding representation of the photo and sketch models, which should be addressed to ensure a faithful image retrieval.

In this work, we tackle the Data Unreliability problem by proposing the RTL, a modified version of the triplet loss that takes into account the similarity between anchors (the photos) to weigh the calculated loss. The goal of the RTL is to reduce the impact of *Ambiguous Samples*, which is discussed in more detail in section 3.1. The Cross-Source Domain Misalignment is tackled by introducing batch normalization layers on the embeddings of both models involved. Further explanations regarding this choice will be presented in section 3.3.

In [34], the authors draw attention to the fact that a typical SBIR application has primarily two distinct steps: (1) an *offline* step, where representations for all photos are first extracted and then stored in a database; (2) an *online* step, where a user draws a sketch used as a query to find the corresponding photo. In the first step, features are extracted for all the photos in a collection (such as product photos on an e-commerce site), and occasional updates are made when new photos are added to the database. This offline part concerns only photos and offers more flexibility in terms of resources (time and computing resources) that can be allocated to this task. The extracted features are then stored in a database intended to be used with a kNN-like search approach. During the second step, resource allocation becomes significantly more critical. On the one hand, we have the user who draws a sketched query and waits for a response. On the other hand, the less resource-intensive the model is, the easier it will be to extract the features of the query sketch locally (on a mobile phone, for example). In [34], the authors have shown that, despite the fact that in all previous works, researchers systematically use the same architecture (e.g. ResNet50) for both modalities (sketch/photo), this was not a mandatory condition. They trained a ResNet18 for sketches and a ResNet34 for photos using the triplet loss and achieved state-of-the-art results. Additionally, they hypothesized that efficiently encoding sketches could be achieved with a smaller model compared to encoding photos. In this study, we adopt the principle of hybrid architectures to conduct a series of experiments. The results of these experiments show that the latter hypothesis is valid. Additionally, by using a new pipeline that combines the benefits of RTL, batch normalization, and knowledge distillation, we can achieve state-of-the-art results. In our experiments, we replace a ResNet34 with a model as small as a ShuffleNetV2 for sketch encoding without any

significant decrease in performance.

To summarize, this paper proposes several contributions to improve SBIR, including:

- Examining the training data quality and identifying the issue related to *Ambiguous Samples*;

- Proposing a Relative Triplet Loss (RTL) to overcome the limitations of traditional triplet loss through loss weighting based on anchor similarity;

- Showing that batch normalization is more suited for SBIR embeddings and that it significantly improves the performance of the models;

- Investigating and validating the hypothesis made in [34] about the SBIR encoders requirements. We show that indeed photo encoders require higher capacity than sketch encoders;

- Proposing a straightforward approach to efficiently train small models, such as ShuffleNetV2, with a marginal loss of accuracy through following a straightforward recipe;

- Outperforming the state-of-the-art on the most used large-scale benchmark for SBIR, The Sketchy database, to reach a recall of 62.38% at $k = 1$.

Overall, the proposed contributions aim to address the challenges of SBIR and provide a practical recipe for improving its accuracy and adapting to different resource requirements.

2. Related Works

In the field of computer vision, supervised learning using Convolutional Neural Networks (CNNs) has been delivering state-of-the-art results for a few years now [9, 39, 11, 5, 3, 20, 24]. The impressive ability of CNNs to extract relevant features directly from pixels without the need for classic feature extraction methods has made them a popular and powerful tool for multiple computer vision tasks. Furthermore, when compared to hand-crafted features (i.e. shallow features), CNN features have been shown to achieve higher performance [10, 40, 36, 12, 13, 23] in generating representations for tasks such as content-based image retrieval (CBIR). This same ability of CNN has also significantly improved sketch recognition and SBIR, outperforming previous solutions based on hand-crafted features [14, 25, 6, 15, 44].

Several studies have utilized CNNs to develop solutions for sketch-edge map matching. For instance, in [38, 26, 31, 27], researchers employed a CNN, initially trained for sketch recognition, to extract features and build an SBIR solution. Their underlying assumption was that

photos' edge maps are visually closer to sketches. In a similar vein, Qi et al. [26] used a Siamese CNN architecture for category-level Sketch-Based Image Retrieval (the pose of the object is ignored, only the category matters). The Siamese architecture involves two branches- one for the sketches and the other for the edge maps. During training, the model was fed with sketch-edge map pairs, and a binary label determined if both the sketch and the edge map belonged to the same category. The loss function was then computed, and the model parameters are updated to extract improved representations. Moreover, pair losses were used in a more generalized approach to project inputs into a feature space that minimizes the distance between positive pairs (similar inputs) by a margin of m_p while ensuring the distance for negative pairs is larger than a second margin, m_n. However, using a fixed margin for all pairs is a significant drawback as it fails to account for the variance of (dis)similarity between different pairs. To overcome this limitation, researchers have turned to Triplet Loss, which presents inputs as triplets consisting of a reference sample, a positive sample (similar to the reference), and a negative sample (dissimilar to the reference). During training, the model learns to project inputs into a space where a positive example is closer to the reference than a negative one, based on a relative distance measure. This approach allows the model to manage arbitrary feature space distortions and is more suitable for CBIR/SBIR applications, which has garnered significant attention in recent years [37, 8, 7]. Bui et al. [4] investigated in-depth weight-sharing strategies and generalization capabilities of triplet CNNs for SBIR. In [34], the authors conducted a comprehensive study of classic triplet CNN training pipelines in the SBIR context and proposed several avenues for improvement. They highlight the importance of several choices made when building SBIR solutions such as embedding normalization, model sharing, margin selection, batch size, and hard mining selection. To overcome the lack of annotated sketches, Bhunia et al. [2] proposed a photo-to-sketch generator using a GAN architecture to synthesize sketches for unlabeled photos. The synthetic sketch-photo pairs were then used to train a triplet CNN. In [35], the authors proposed a modified sampling pipeline used during training that makes it harder through mini-batches partially filled with samples flipped and a higher number of samples belonging to the same category. In [46], Zhang et al. incorporated a deformable CNN layer to handle sketch variability. In addition to the triplet loss, Lin et al. [17] experimented with a combination of three loss functions (SoftMax loss, Spherical loss[19], and Center loss [42]). Attention modules were also added by some researchers to improve the capturing of fine information [37, 8, 7, 33]. Alternatively, researchers introduced quadruplet networks in [32] to encode semantic information similarly to triplets for local information. In

a more sophisticated approach, Wang et al. [41] proposed a three-stage solution for SBIR, where textual descriptions were used as additional input to the pipeline to reduce the gap between sketches and images. In [29], Sain et al. proposed a cross-modal variational autoencoder to disentangle the semantic content and sketcher style in sketches to build a style-agnostic model. In [35, 28], the authors used a transformer architecture to achieve state-of-the-art results. In this work, we conduct a comprehensive analysis of various studies to identify the best recipe for creating efficient SBIR solutions. We define an efficient SBIR solution as one that achieves high performance while taking into account the peculiarities of the problem under study, as well as the practical peculiarities that can be leveraged for even greater effectiveness.

3. Methodology

In this section, we present the methodology that we follow to build our recipe and the intuitions behind the different choices we make. We start by introducing the Regional Maximum Activation of Convolutions (RMAC) approach, which we use to measure the similarity between the training photos. Next, we describe our proposed RTL, which overcomes the limitations of the standard triplet loss by incorporating a similarity-based loss weighting mechanism. We then discuss the importance of batch normalization for embeddings, which improves the cross-domain misalignment between the photo and sketch embeddings. Finally, we describe our approach to training efficiently a small model for sketch encoding.

3.1. Identifying ambiguous samples using RMAC

In [40], the authors demonstrated that a CNN approach can compete with traditional methods on challenging image retrieval benchmarks. To extract features, they discarded the fully connected layers of a pre-trained VGG16 and used the resulting fully convolutional network for feature extraction. For each image input, the output feature maps form a 3D tensor of shape $C \times W \times H$, where C is the number of channels, and (W, H) are the width and height of the feature maps. By representing this tensor as a set of 2D feature maps $\mathcal{X} = \mathcal{X}c$, $c = 1...C$, the Maximum Activations of Convolutions (MAC) can be computed using $\max x \in \mathcal{X}_c x$ for each c. To compute the RMAC descriptor, Tolias et al. [40] proposed a method to sample a set of square regions $R = R_i$ within $\mathcal{X}$ using a sliding window approach with a square kernel of width $k_w = 2 \times \min(W, H)/(l + 1)$ and stride $60\% \times k_w$ at $L = 3$ different scales. Then, for each region, the descriptor f_{R_i} is computed using $\sum_{x \in \mathcal{R}_{i,c}} x^{\alpha}$ with $\alpha = 10$ and normalized using l_2 normalization, PCA-whitening, followed by an additional normalization. Finally, all the re-

sulting vectors are combined and normalized to obtain the final RMAC descriptor. Several variants have been proposed [10, 36, 16, 12] to build a stronger RMAC descriptor through modifications such as using multi-resolution inputs, features extracted from different layers, normalization, and aggregation. In this work, we adopt some of these modifications: replacing approximate pooling with max pooling, removing PCA-whitening, and using a multi-resolution RMAC descriptor (we use three resolutions for the photos: $S = 384, 512, 768$). We compute the RMAC descriptor for all the photos on the training set and use the euclidean distance to compare them. Then, we visually checked the top 100 similar pairs of photos to check if the number of *Ambiguous Samples* is significant. As we can see in Figure 1, in several cases the images to be discriminated against are too similar or even identical, making it impossible to discriminate them with a simple sketch.

3.2. Relative Triplet Loss

In this work, we propose RTL, a modified version of triplet loss that aims to incorporate relativity in the loss computation to address the problem described above. In an efficient SBIR solution, the photo encoding must be discriminating enough to meet the margin constraint imposed by the triplet loss. During our experiment, we assume our photo encoder to perform the encoding sufficiently well in order to find a correspondence between a sketch and a photo. Furthermore, as a side-effect of the SBIR training, we assume that our photo encoder is improving at its task during the learning phase. Under these assumptions, the embeddings extracted for photos to compute the triplet loss can be used to measure the similarity, computed using the euclidean distance, between the current mini-batch photos. Let us assume that we have a mini-batch with bs samples (photos). We first compute the similarity matrix $M_{bs \times bs}$ between all the photos in the mini-batch. We then normalize $M_{bs \times bs}$ by dividing it by its maximum value $max(M_{bs \times bs})$ to obtain our weighting matrix $W_{bs \times bs}$. To switch from a classic triplet loss to RTL, once we have the triplet loss matrix, we multiply it element-wise by the matrix $W_{bs \times bs}$ before aggregation. The complete pipeline of our approach is detailed in Algorithm 1.

Figure 1. The top 25 similar pairs of photos retrieved from the training set of the Sketchy benchmark using the RMAC descriptor. In several cases, the images are too similar or even identical to be distinguished using a simple sketch.

Algorithm 1 Relative Triplet Loss

Require:
1: Batch size: bs
2: Batch of photos: P, P_i where $i = 0...bs$
3: Batch of sketches: S, S_i where $i = 0...bs$ and S_i is a sketch matching the photo P_i
4: Margin: m
5: Photos embedding function: $f_p(\cdot)$
6: Sketches embedding function: $f_s(\cdot)$
7: Distance function: $D(\cdot, \cdot)$
8: Rectified linear unit: $ReLU(\cdot)$
9: Identity matrix: I_{bs}

Ensure: RTL loss: L_{RTL}
10: Mini-batch photos embeddings: $P\_embs = f_p(P)$
11: Mini-batch sketches embeddings: $S\_embs = f_s(S)$
12: Distance between the anchors and positive samples: $d_{a,p} = D(P\_embs_i, S\_embs_j)$ with $i = j$
13: Distance between the anchors and negative samples: $d_{a,n} = D(P\_embs_i, S\_embs_j)$ with $i \neq j$
14: We expand $d_{a,p}$ and compute the triplet loss matrix: $TL_{matrix} = ReLU(d_{a,p} - d_{a,n} + m)$
15: Then, we compute the weighting matrix: $W_{bs \times bs} =$

3.3. Batch normalization for embeddings

The internal feature distributions of neural networks are highly dependent on the domain that they are operating on, which makes it difficult to directly compare distributions in a cross-source setting. To alleviate such a distribution shift and encourage a better distribution alignment between our two models, we propose to draw inspiration from the batch normalization technique and to normalize the output activations of each domain model via domain-specific normalization statistics. Because it is less sensitive to outliers, batch normalization better preserves the representation range of the embeddings than other commonly used normalization schemes such as l_2-normalization. As a result, we find that models using batch normalization on their embeddings have well-behaved training dynamics and reach better performances. We hypothesize that thanks to its learnable parameters, batch normalization allows embeddings originating from the sketch model and those from the photo model to be represented in comparable distributions.

3.4. Training a small model for sketches encoding

As explained in the introduction, in order to reduce the resources needed for the online part of an SBIR solution, we can use smaller models to encode sketches. In our case, we have opted for ShuffleNetV2 (we use the pre-trained shufflenet_v2_x1_0 from torchvision), a state-of-the-art model that achieves high accuracy with low computation costs. Its tradeoff between accuracy and low computation costs makes it an adequate candidate for SBIR applications. ShuffleNetV2 was designed to meet the needs of mobile devices (limited computing power and storage capacity) and real-time applications (fast inference speed).

Early experiments conducted on ShuffleNetV2 as a sketch encoder have revealed struggles in convergence, leading to a significant decrease in performance when compared to larger models such as ResNet34. This phenomenon can be explained by the drastically low number of parameters (ShuffleNetV2 has almost 20 times fewer parameters than ResNet34), making it difficult for such a small model to capture the non-negligible complexity of the cross-domain inputs. In order to overcome this last hurdle, we came up with the idea of using knowledge distillation to transfer knowledge from a large model pre-trained to encode sketches and that has proven its effectiveness, to a smaller model (ShuffleNetV2 in our case). In this manner, we can circumvent the complexity related to the cross-modality nature of the training and focus more on the validity of the initial hypothesis (small models are enough for sketch encoding). Knowledge Distillation techniques work by transferring the knowledge of a large and powerful model, the teacher, to a smaller and simpler one, the student, by having the student model regress the output of the teacher. Such a method usually leads to students having better generalization capabilities since the teacher's output implicitly encodes more information about the similarity between training samples and their distribution than hard labels.

In this work, we propose to apply such a training strategy to our models. In particular, we use a response-based knowledge distillation technique, where the student learns to mimic the output embeddings of a teacher. In that regard, several learning objectives have been addressed, providing different convergence abilities to the student. The respective output embeddings of the teacher and the student have been compared according to (1) Mean-Squared Error; (2) Huber Loss; (3) A combination of Mean-Squared Error and Mean-Absolute Error.

We also explore variants of the traditional knowledge distillation techniques, by using students of comparable or even larger capacities than the teacher. Such an alteration has been shown to lead to student models learning a model ensemble jointly with regular knowledge distillation and to lead to a better-performing student [1].

4. Experiments

In this section, we detail the different experiments conducted for this study. For the whole study, we utilize The Sketchy benchmark [30], a large-scale comprehensive collection designed specifically for SBIR. This benchmark comprises 75,471 sketches for 12,500 unique objects across 125 categories (the benchmark contains 100 photos per category). To create this dataset, crowd workers were instructed to sketch various photographic objects, resulting in a diverse range of sketch styles and interpretations. For each photo, there are at least five sketches from different workers to ensure a robust set of fine-grained associations between sketches and photos. To ensure consistency, the authors provide a series of guidelines to follow, including a test set list to split data into a training and test set. Specifically, 90% of the data are used for training, and the remaining 10% are used at test time. We follow these guidelines to ensure a fair and reliable evaluation of our models' performance.

4.1. RTL and batch Normalization for better embeddings

At the beginning of our experiments, we follow the pipeline proposed in [35] with some minor modifications. In particular, we do not use a ResNet50 or a Transformer model but use a ResNet18 and a ResNet34 instead. As in [35] we use pre-trained versions of these models (trained on ImageNet [9]) provided by the torchvision library. We also use the output of the last pooling layer (adaptive average pooling) to extract the embeddings (without applying l_2-normalization). We use two distinct encoders for sketches and photos. We train our models for 200 epochs. We set the learning rate to $lr = 10^{-4}$ for the first 100 epochs and

Model	$Recall@1\%$
$R18$ [35]	52.98
$R18_{RTL}$	55.27
$R18_{RTL+BN}$	57.20
$R34$ [35]	56.10
$R34_{RTL}$	58.50
$R34_{RTL+BN}$	59.99

Table 1. Our results achieved on The Sketchy Database with RTL and batch normalization (BN) compared to [35].

Model	Epochs	$Recall@1\%$
$ShuffleNetV2_{sketches}$	200	51.96
$ShuffleNetV2_{sketches}$	300	52.81
$ShuffleNetV2_{sketches}$	500	54.19
$ShuffleNetV2_{sketches}$	600	54.86
$ShuffleNetV2_{sketches}$	700	55.66
$ShuffleNetV2_{sketches}$	800	55.97
$ShuffleNetV2_{sketches}$	900	56.01

Table 2. Our results achieved on The Sketchy Database with $ShuffleNetV2_{sketches}$ and $R34_{photos}$.

Model	$Recall@1\%$
$ShuffleNetV2_{KL}$	53.3
$ShuffleNetV2_{KL+SM}$	56.6
$ShuffleNetV2_{MSE}$	57.71
$ShuffleNetV2_{MSE+MAE}$	58.31
$ShuffleNetV2_{Huber}$	58.5

Table 3. Our results achieved on The Sketchy Database after knowledge distillation from $R34_{sketches}$ to $ShuffleNetV2_{sketches}$. The $R34_{photos}$ is used as the photo encoder.

we change it to $lr = 10^{-6}$ for the second 100 epochs. The batch-size bs and the margin m are kept constant for this study, we use $bs = 256$ (instead of $bs = 128$ in [35]) and $m = 3$.

In Table 1, we compare our results with equivalent models from [35] that we consider as baselines. We can see that replacing triplet loss with RTL and adding a batch normalization layer, both bring significant improvements.

4.2. Training efficiently a small encoder for sketches

4.2.1 Training with RTL and batch normalization

We used our best photo encoder ($R34_{RTL+BN}$) from previous experiments (we freeze all the layers of the photo encoder, including batch normalization parameters) and a $ShuffleNetV2_{sketches}$ for sketch encoding (we replaced the classification layer with a fully connected layer with 512 outputs to reduce the number of channels, followed by a batch-normalization layer). We followed the same training pipeline as before with RTL and batch normalization. We trained the model for 200 epochs and noticed the performance decreased by more than 8%. To verify if such a decrease is an indicator of a model limitation or that the model is struggling to converge, we train for longer. As shown in Table 2, we can see that after relatively long training, the model ends up reaching results closer to those achieved with $R18_{RTL+BN}$ (Table 1).

However, despite these satisfactory results, we still have nearly 4% decrease in accuracy compared to an $R34_{sketches}$ and a training strategy that starts to show some weaknesses that should not be ignored in our quest for a straightforward recipe for efficient SBIR solutions.

4.2.2 Knowledge distillation

As mentioned in section 3.4, knowledge distillation offers an attractive solution to avoid dealing directly with the complicated nature of cross-modality training. In addition, it provides a good solution to test the initial hypothesis about the sketch encoder size. During the training, we noticed that with this approach the training became fast (it needs less than 200 epochs to converge) and smooth (the evo-

lution is stable). In Table 3 (we removed the $_{sketches}$ subscript for better readability), we report our results for the experiments with $R34_{sketches}$ knowledge transfer to $ShuffleNetV2_{sketches}$ (pre-trained on ImageNet). We can see that this approach enabled us to almost reach our initial goal, at this point we are only 1.5% far away.

4.2.3 Double guidance for finetuning after knowledge distillation

An intuitive and obvious next step after the success achieved with knowledge distillation was to finetune the new efficient $ShuffleNetV2_{sketches}$ with RTL and $R34_{photos}$. To do so, we started following the previously used pipeline. But unfortunately, even after extensive hyperparameter tuning, the accuracy continued to drop with training. We started to believe that a partial ability acquired during the knowledge distillation phase, is not a requirement for the triplet loss constraint. If this assumption is valid, then it is possible that the model loses it during the finetuning.

In order to alleviate that deficiency, we propose a double guidance pipeline for finetuning after knowledge distillation. In this new pipeline, we use both, the $R34_{photos}$ and the $R34_{sketches}$ at the same time to train our $ShuffleNetV2_{sketches}$. The parameters of $R34_{photos}$ and $R34_{sketches}$ are not updated, the models are only used to guide the $ShuffleNetV2_{sketches}$. While the latter learns to extract embeddings that respect the RTL constraint with those of $R34_{photos}$, at the same time, it also learns to mimic

Figure 2. The proposed double guidance pipeline for an efficient finetuning after knowledge distillation. In this novel pipeline, an additional branch with a sketch encoder teacher is used to guide the student model with Huber Loss.

the embeddings generated with $R34_{sketches}$ for sketches thanks to the additional Huber loss as shown in Figure 2. This new pipeline enabled us to gain an additional 0.6% to reach a $recall@1 = 59.1$. In Table 4 (we removed the $_{sketches}$ subscript for better readability), we report our result for double guidance finetuning, in addition to results of knowledge distillation experiments with models larger than $ShuffleNetV2$ for sketch encoding, and even larger than the teacher model $R34_{sketches}$. As we can see, comparable results were achieved with models of significantly different sizes. We assume that these results are sufficient proof of the validity of the initial hypothesis for sketch encoding. Following our pipeline, a small sketch encoder can be used for SBIR applications with a marginal loss of accuracy.

4.3. Training Large Encoders for Photos

The second part of the initial assumption was about the necessity of relatively big models to encode photos efficiently. We proceed in a similar manner as before to check the validity of this hypothesis. This time, we use the $R34_{photos}$ for knowledge distillation, and we analyze the performance evolution. In Table 5 (we removed the $_{photos}$ subscript for better readability), we summarize the results of our experiments. We can conclude that the size of the photo encoder matters. We can also notice that a large student is even able to surpass the teacher with little effort. Unlike training with triplet loss, knowledge distillation shows more

Model	$Recall@1\%$
$ShuffleNetV2_{Huber}$	58.5
$ShuffleNetV2_{Huber+DG}$	59.1
$R18_{Huber}$	59.8
$R50_{Huber}$	59.89
$R101_{Huber}$	60.24
$R152_{Huber}$	59.7

Table 4. Our results achieved on The Sketchy Database after knowledge distillation from $R34_{sketches}$ to multiple models. For this experiment, the $R34_{photos}$ is used as the photo encoder (DG is used to indicate that a double guidance finetuning pipeline was used after knowledge distillation).

Model	$Recall@1\%$
$ShuffleNetV2_{Huber}$	54.31
$R18_{Huber}$	56.62
$R50_{Huber}$	60.7
$R101_{Huber}$	61.98
$R152_{Huber}$	62.38

Table 5. Our results on The Sketchy Database after knowledge distillation from $R34_{photos}$ to multiple models. For this experiment, the $R34_{sketches}$ is used as the sketch encoder.

Photo encoder	$Recall@1\%$
$R34$	59.1
$R50_{Huber}$	59.18
$R101_{Huber}$	60.88
$R152_{Huber}$	61.45

Table 6. Our results on The Sketchy Database after knowledge distillation and double guidance of the sketch encoder ($ShuffleNetV2_{Huber+DG}$) tested with multiple photo encoders.

stability during training, in addition, hyperparameter tuning is easier and less time-consuming.

4.4. Combining a small sketch encoder with large photo encoders

In Table 6 (we removed the $_{photos}$ subscript for better readability), we report the results obtained when combining our $ShuffleNetV2_{Huber+DG}$ and different models larger than the teacher model ($R34$). We notice that we have been able to surpass even the initial performance achieved with a ResNet34 used for both encoders. In addition, the proposed recipe offers attractive flexibility that enables the development of SBIR solutions with multiple backbones meeting different requirements.

4.5. Comparison with state-of-the-art methods

In this section, we compare some of our study results with those of previous research on The Sketchy benchmark.

These results are reported in Table 7. As can be seen in this table, if we compare our results with others with the same architecture (e.g. ResNet18, ResNet34, ResNet50), we notice that using RTL and batch normalization alone bring a significant improvement. And that they surpass even $ResNet18_{2\times2}$ and $ResNet34_{2\times2}$ proposed in [35], where the last average pooling was modified to reduce the spatial resolution to 2×2, which increases four times the embedding size. In addition, our largest distilled photo encoder $R152_{Huber}$, when used with the sketch encoder $R34_{RTL+BN}$, they achieve comparable results to those of the double vision transformer solution proposed in [35]. And the latter is the only solution that we found in SBIR literature to surpass the results achieved by our hybrid solutions (different architectures for the sketch encoder and photo encoder), even when a $ShuffleNetV2$ is used as sketch encoder.

5. Conclusion

In this paper, we have presented a comprehensive study on improving SBIR solutions by tackling some of its major limitations. Starting with pointing out and demonstrating the existence of an issue with data reliability that has been largely ignored. To address this problem, we have proposed a Relative Triplet Loss (RTL), a modified version of the triplet loss, that takes into account the similarity between anchors to relatively adapt the computed loss. We have also shown that batch normalization is more suitable for SBIR embeddings compared to adding an l_2-normalization layer, and it significantly improves the performance of our models. Furthermore, we have investigated the capacity of models required for the photo and sketch domains and demonstrated that the photo encoder requires a higher capacity than the sketch encoder. Additionally, we have proposed a straightforward recipe based on knowledge distillation to efficiently train small models and even reach higher accuracy with larger ones. Our experimental results demonstrate that our proposed method outperforms previous state-of-the-art results and provides a strong pipeline for building more efficient SBIR solutions. Overall, our work provides a practical recipe for improving both the performance and the efficiency of SBIR systems, which can benefit a wide range of applications, including e-commerce systems.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. More photos are all you need: Semi-supervised learning for fine-grained sketch based image retrieval. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 4247–4256, 2021.

Model	$Recall@1(\%)$
Chance [30]	0.01
Sketch me that shoe [45]	25.87
Siamese Network [30]	27.36
Triplet Network [30]	37.10
Quadruplet_MT [32]	38.21
DCCRM(S+I) [41]	40.16
DeepTCNet [21]	40.81
Triplet attention [33]	41.66
Quadruplet_MT_v2 [32]	42.16
LA [43]	43.1
DCCRM(S+I+D) [41]	46.20
Human [30]	54.27
ResNet18 [8]	45.95
DCCRM [41]	46.20
ResNet50 [8]	52.19
ResNet18 [34]	52.75
ResNet18 [35]	53.61
ResNet101 [8]	54.59
DLA [43]	54.9
$ResNet18_{2\times2}$ [35]	55.10
Our $R18_{RTL}$	**55.27**
ResNet50 [35]	56.29
Our $R18_{RTL+BN}$	**57.20**
MLRM [18]	57.20
ResNet34 [35]	57.43
$ResNet34_{2\times2}$ [35]	58.23
$ResNet50_{2\times2}$ [35]	58.37
Our $R34_{RTL}$	**58.50**
Our $ShuffleNetV2_{Huber+DG}$	**59.1**
Our $R34_{RTL+BN}$	**59.99**
VT [35]	62.25
Our $R152_{Huber}$	**62.38**

Table 7. Our main results compared to state-of-the-art solutions on The Sketchy Database. We can observe that our training recipe brings significant improvements for models with the same architecture.

[3] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021.

[4] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John Collomosse. Generalisation and sharing in triplet convnets for sketch based visual search. *arXiv preprint arXiv:1611.05301*, 2016.

[5] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks. *arXiv preprint arXiv:2212.11696*, 2022.

[6] Xiaochun Cao, Hua Zhang, Si Liu, Xiaojie Guo, and Liang Lin. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 313–320,

2013.

[7] Ushasi Chaudhuri, Biplab Banerjee, Avik Bhattacharya, and Mihai Datcu. Crossatnet-a novel cross-attention based framework for sketch-based image retrieval. *Image and Vision Computing*, 104:104003, 2020.

[8] Yangdong Chen, Zhaolong Zhang, Yanfei Wang, Yuejie Zhang, Rui Feng, Tao Zhang, and Weiguo Fan. Ae-net: Fine-grained sketch-based image retrieval via attention-enhanced network. *Pattern Recognition*, 122:108291, 2022.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, pages 1–18, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Syed Sameed Husain and Miroslaw Bober. Remap: Multilayer entropy-guided pooling of dense cnn features for image retrieval. *IEEE Transactions on Image Processing*, 28(10):5201–5213, 2019.

[13] Jaeyoon Kim and Sung-Eui Yoon. Regional attention based deep feature for image retrieval. In *BMVC*, page 209, 2018.

[14] Joseph J LaViola Jr and Robert C Zeleznik. Mathpad2: a system for the creation and exploration of mathematical sketches. In *ACM SIGGRAPH 2006 Courses*, pages 33–es. 2006.

[15] Yi Li, Yi-Zhe Song, Shaogang Gong, et al. Sketch recognition by ensemble matching of structured features. In *BMVC*, volume 1, page 2, 2013.

[16] Yang Li, Yulong Xu, Jiabao Wang, Zhuang Miao, and Yafei Zhang. Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. *IEEE Signal Processing Letters*, 24(5):609–613, 2017.

[17] Hangyu Lin, Yanwei Fu, Peng Lu, Shaogang Gong, Xiangyang Xue, and Yu-Gang Jiang. Tc-net for isbir: Triplet classification network for instance-level sketch based image retrieval. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1676–1684, 2019.

[18] Zhixin Ling, Zhen Xing, Jiangtong Li, and Li Niu. Multilevel region matching for fine-grained sketch-based image retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 462–470, 2022.

[19] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[20] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[21] Peng Lu, Hangyu Lin, Yanwei Fu, Shaogang Gong, Yu-Gang Jiang, and Xiangyang Xue. Instance-level sketch-based retrieval by deep triplet classification siamese network. *arXiv preprint arXiv:1811.11375*, 2018.

[22] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[23] Federico Magliani and Andrea Prati. An accurate retrieval through r-mac+ descriptors for landmark recognition. In *Proceedings of the 12th International Conference on Distributed Smart Cameras*, pages 1–6, 2018.

[24] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[25] Tom Y Ouyang and Randall Davis. Chemink: a natural real-time recognition system for chemical drawings. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 267–276, 2011.

[26] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. Sketch-based image retrieval via siamese convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)*, pages 2460–2464. IEEE, 2016.

[27] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *Proceedings of the european conference on computer vision (eccv)*, pages 751–767, 2018.

[28] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14153–14162, 2020.

[29] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8504–8513, 2021.

[30] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016.

[31] Omar Seddati, Stéphane Dupont, and Saïd Mahmoudi. Deepsketch2image: deep convolutional neural networks for partial sketch recognition and image retrieval. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 739–741, 2016.

[32] Omar Seddati, Stéphane Dupont, and Saïd Mahmoudi. Quadruplet networks for sketch-based image retrieval. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 184–191, 2017.

[33] Omar Seddati, Stéphane Dupont, and Saïd Mahmoudi. Triplet networks feature masking for sketch-based image retrieval. In *Image Analysis and Recognition: 14th International Conference, ICIAR 2017, Montreal, QC, Canada, July 5–7, 2017, Proceedings*, pages 296–303. Springer, 2017.

[34] Omar Seddati, Stéphane Dupont, Saïd Mahmoudi, and Thierry Dutoit. Towards human performance on sketch-based image retrieval. In *Proceedings of the 19th International Conference on Content-based Multimedia Indexing*, pages 77–83, 2022.

[35] Omar Seddati, Stéphane Dupont, Saïd Mahmoudi, and Thierry Dutoit. Transformers and cnns both beat humans on sbir. *arXiv preprint arXiv:2209.06629*, 2022.

[36] Omar Seddati, Stéphane Dupont, Saïd Mahmoudi, and Mahnaz Parian. Towards good practices for image retrieval based on cnn features. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1246–1255, 2017.

[37] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 5551–5560, 2017.

[38] Yuxin Song, Jianjun Lei, Bo Peng, Kaifu Zheng, Bolan Yang, and Yalong Jia. Edge-guided cross-domain learning with shape regression for sketch-based image retrieval. *IEEE Access*, 7:32393–32399, 2019.

[39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[40] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[41] Yanfei Wang, Fei Huang, Yuejie Zhang, Rui Feng, Tao Zhang, and Weiguo Fan. Deep cascaded cross-modal correlation learning for fine-grained sketch-based image retrieval. *Pattern Recognition*, 100:107148, 2020.

[42] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 499–515. Springer, 2016.

[43] Jiaqing Xu, Haifeng Sun, Qi Qi, Jingyu Wang, Ce Ge, Lejian Zhang, and Jianxin Liao. Dla-net for fg-sbir: Dynamic local aligned network for fine-grained sketch-based image retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5609–5618, 2021.

[44] KT Yesilbek11, C Sen11, S Cakmak11, and TM Sezgin. Svm-based sketch recognition: which hyperparameter interval to try? 2015.

[45] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.

[46] Xianlin Zhang, Mengling Shen, Xueming Li, and Fangxiang Feng. A deformable cnn-based triplet model for fine-grained sketch-based image retrieval. *Pattern Recognition*, 125:108508, 2022.