Solving Mixed-modal Jigsaw Puzzle for Fine-Grained Sketch-Based Image Retrieval

Kaiyue Pang[1,2] Yongxin Yang[1] Timothy M. Hospedales[1,3] Tao Xiang[1] Yi-Zhe Song[1]
[1]SketchX, CVSSP, University of Surrey [2]Queen Mary University of London [3]The University of Edinburgh
kaiyue.pang@qmul.ac.uk, t.hospedales@ed.ac.uk, {yongxin.yang, t.xiang, y.song}@surrey.ac.uk

Abstract

ImageNet pre-training has long been considered crucial by the fine-grained sketch-based image retrieval (FG-SBIR) community due to the lack of large sketch-photo paired datasets for FG-SBIR training. In this paper, we propose a self-supervised alternative for representation pre-training. Specifically, we consider the jigsaw puzzle game of recomposing images from shuffled parts. We identify two key facets of jigsaw task design that are required for effective FG-SBIR pre-training. The first is formulating the puzzle in a mixed-modality fashion. Second we show that framing the optimisation as permutation matrix inference via Sinkhorn iterations is more effective than the common classifier formulation of Jigsaw self-supervision. Experiments show that this self-supervised pre-training strategy significantly outperforms the standard ImageNet-based pipeline across all four product-level FG-SBIR benchmarks. Interestingly it also leads to improved cross-category generalisation across both pre-train/fine-tune and fine-tune/testing stages.

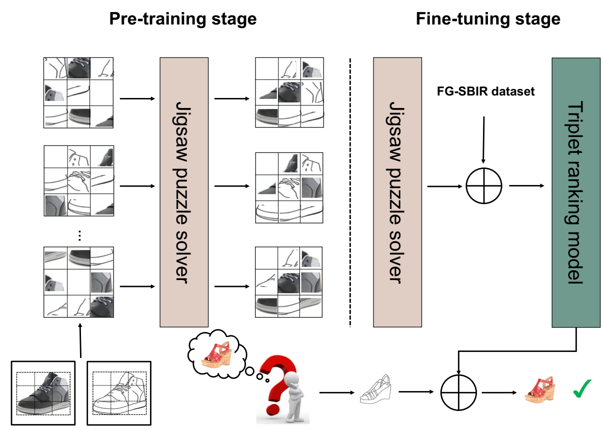


Figure 1: Conventionally, a competitive FG-SBIR system relies on two prerequisites: ImageNet pre-training and triplet fine-tuning. Here we investigate substituting the former with a mixed-domain jigsaw puzzle solver, leading to improved FG-SBIR accuracy and generalisation.

1. Introduction

Fine-grained sketch-based image retrieval (FG-SBIR) methods enable users to express their mental image or visual intention via free-hand sketch, so as to retrieve a photo of a *specific* object instance. Due to its commercial potential, the field has flourished recently with various research angles being considered including CNN architecture [19], attention [28], choice of instance matching loss [23], improving efficiency via hashing [36], and data augmentation via heuristics [34] or abstraction [22].

Despite the great strides made, almost all contemporary competitive FG-SBIR models depend crucially on one necessary condition: the model must be fine-tuned from the pre-trained weights of an ImageNet [6] classifier. The reason behind this is that collecting instance-level sketch-photo pairs for FG-SBIR is very expensive, with the largest current single product-category dataset being only on a scale of thousands. Scaling such data collection to the size required to train a contemporary deep CNN from scratch is infeasible. Thus, ImageNet pre-training is ubiquitously leveraged to provide initialisation for FG-SBIR.

While useful in ameliorating the otherwise fatal lack of data for FG-SBIR, ImageNet pre-training suffers from mismatch to the intended downstream task. Training for object category classification requires detecting high-level primitives that characterise different object categories, while learning to ignore certain fine-grained details critical for the instance-level recognition task in FG-SBIR. Crucially, ImageNet only contains images from the photo modality, while FG-SBIR requires cross-modality matching between photo and sketch. This suggests that ImageNet classification may not be the most effective pre-training strategy for FG-SBIR. Indeed, recently [20] explored the self-supervised task of matching a photo with its edgemap to substitute the sketch-photo pair for model training. This could potentially be used for pre-training as well. However, its effectiveness is limited because the task boils down to edge detection and is not challenging enough for the model to learn fine-grained cross-modal discriminative patterns for matching.

This paper challenges the longstanding practice of resorting to ImageNet pre-training for FG-SBIR, and introduces a simple and effective self-supervised alternative. Specifically, we propose to perform representation pre-training by recovering an image from mixed-modal shuffled patches. That is, patches drawn randomly from photo and edgemap domains. Solving this problem, as illustrated in Figure 1, requires learning to bridge the domain discrepancy, to understand holistic object configuration, and to encode fine-grained detail in order to characterise each patch accurately enough to infer their relative placement.

Note that jigsaw solving has been studied before [15, 5] for single-modal recognition problems. In this work, differently, we deal with a more challenging mixed-modal jigsaw problem. Solving jigsaw puzzle as a task itself is hard; as a result, instead of directly solving it, i.e., recovering the unshuffled original image where all patches are put back to the right places, most prior work [15, 11, 5] poses jigsaw solving as a recognition task. In contrast, we frame the jigsaw solving problem as a permutation inference problem and solve it using Sinkhorn iterations [3, 24]. Our experiments show that this formalisation of a jigsaw solver provides a much stronger model for self-supervised representation pre-training. A surprising outcome is that this approach can completely break the category associations between representation pre-training and FG-SBIR fine-tuning without harming performance, as well as lead to improved generalisation across categories between FG-SBIR fine-tuning and run-time testing stage.

Our contributions are as follows: (1) We provide the first study of pre-training approaches for FG-SBIR. (2) We propose a novel mixed-modality jigsaw puzzle solver as an effective pre-training strategy. (3) Extensive experiments on all four publicly available product-level FG-SBIR datasets show for the first time that ImageNet classification is unnecessary as a pre-training strategy for FG-SBIR, and confirm the superiority of our jigsaw approach. The results also show that this leads to improved generalisation across object categories.

2. Related Work

Fine-grained SBIR The problem of fine-grained SBIR was first proposed in [12], which employed a deformable part-based model (DPM) representation and graph matching. More recently, deep learning approaches are heavily favoured, which usually rely on the two stage paradigm of "ImageNet pre-training + FG-SBIR fine-tuning" [23, 34, 28, 18, 36]. This work focuses on replacing the first stage ImageNet pre-training with a more challenging yet more relevant mix-modal jigsaw puzzle solving task. Note that although FG-SBIR is the only problem studied in this work, the proposed method can potentially be applied to any cross-modal matching tasks.

Pre-training + Fine-tuning Many deep CNN based computer vision models assume that a rich universal representation has been captured in ImageNet pre-trained CNN [31, 7, 35, 26], which can then be fined-tuned with task-specific data using various strategies [13, 30, 25, 21, 8]. Especially for tasks with limited training data, fine-tuning an ImageNet pre-trained model is a near-ubiquitous step, to an extent that its efficacy is rarely questioned. Very recently, [10] challenged the conventional wisdom of ImageNet pre-training for downstream tasks like object detection, and demonstrate how similar results can be obtained by training from scratch. However, even in this study, the scale of data required for effective generalisation is beyond that of typical FG-SBIR datasets, thus pre-training is a must. We show that an appropriately designed self-supervised task (mixed-modal jigsaw solving) and model (permutation inference) leads to a strong initial representation for FG-SBIR that outperforms the classic ImageNet pre-training.

Solving Jigsaw Puzzles The first jigsaw puzzle created is believed to have served an educational purpose for royal children to learn geography because of the visuo-spatial processing it involves [1]. Jigsaws have since been a popular form of entertainment for children and adults. Recently, its potential to act as a self-supervisory signal for representation learning has been realised by the computer vision community [15, 24, 11, 14, 5]. Existing jigsaw solvers have posed uni-modal jigsaw tasks, while we show that mixed-modal jigsaws are beneficial for multi-modal representation learning. A more significant factor that differentiates existing approaches is that whether they simplify the learning problem by framing it as a classification task over a predefined set of permutations or directly tackle the permutation problem itself. The latter is more technically demanding as a sparse binary assignment matrix has to be formed with the constraint that each row and column has exactly one "1". It has been shown for certain target tasks, e.g, classification/detection [24], the difference between the two approaches is minor. However, one key finding of this paper is to show that the Sinkhorn-permutation solution to Jigsaw pre-training is crucial to obtaining dramatic improvement in the downstream FG-SBIR (see Sec. 6.1).

3. Jigsaw Pre-training for FG-SBIR

Overview This work aims to introduce a self-supervised pre-training strategy in the form of solving mixed-modal jigsaw puzzles. The whole FG-SBIR training pipeline thus consists of two stages: self-supervised jigsaw pre-training and supervised FG-SBIR triplet fine-tuning. The first self-stage will use photos p and corresponding programmati-

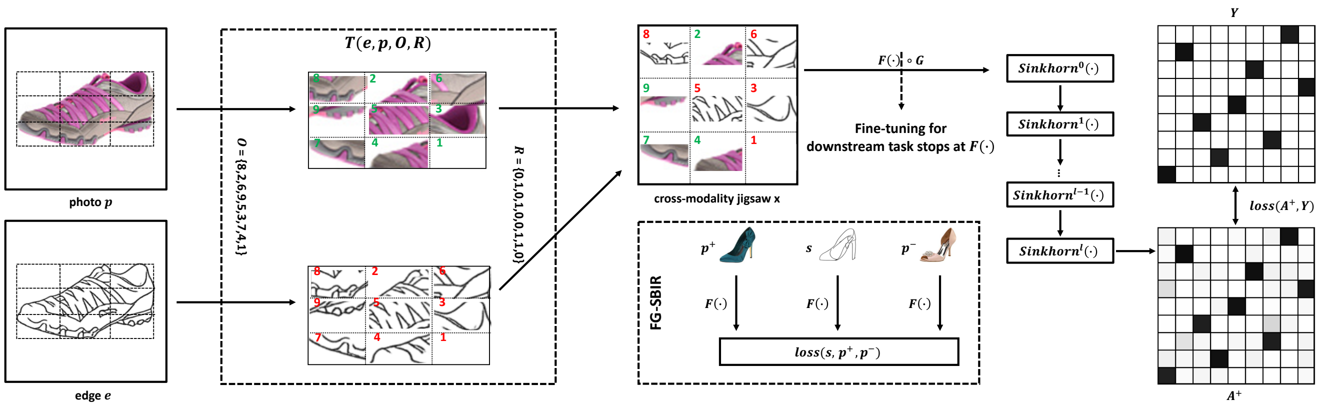


Figure 2: **Schematic of our proposed Jigsaw pre-training for FG-SBIR**. We take a jigsaw puzzle of 9 tiles as an example. Both photo p and its edgemap counterpart e are first divided into 3×3 grid and reshuffled based on a permutation order O. Using a random binary vector R, these are then stitched into the final mixed-modality jigsaw x. x is fed to our jigsaw solver $J(x) = G(F(x))$ including a ConvNet feature extractor $F(\cdot)$ and Sinkhorn-based permutation solver $G(\cdot)$ to obtain the permutation matrix A^+ that solves the jigsaw. After pre-training, we take the CNN module $F(\cdot)$ and use it as a feature extractor for FG-SBIR fine-tuning.

cally produced edgemaps e to produce mixed modal jigsaw images x. Our jigsaw solver $J(x)$ trains a representation by learning to solve these jigsaws. In the second stage, we use the learned representation as an initial condition, and fine-tune a FG-SBIR model by supervised triplet ranking on annotated pairs of free-hand sketches and photos.

Jigsaw Puzzle Generation We first define a cross-modality shuffling operator $x = T(e, p, O, R)$, that transforms a photo p and its edgemap counterpart e to form a mixed-modal jigsaw image x. Assume the jigsaw image is to contain N patches in a $\sqrt{N} \times \sqrt{N}$ array. O is then a random permutation of an array $[1 \dots N]$ that describes the mapping of input image patches to the jigsaw patches in x, and R is a N-dimensional vector of Bernoulli samples that will determine whether input patches are drawn from photo p or edgemap e. Thus, as shown in Figure 2, x is generated by drawing the ith patch from location O_i of the inputs, specifically from sketch if $R_i = 1$ and photo if $R_i = 0$.

Jigsaw Puzzle Solver Our jigsaw solver $J(x)$ processes the mixed-modal jigsaw image x and returns A^+, a $N \times N$ assignment matrix that maps each jigsaw patch to the target patch of an un-shuffled image (Figure 2).

The jigsaw solver $J(x) = G(F(x))$ is implemented via a CNN feature extractor $F(\cdot)$, followed by a permutation solver $G(\cdot)$. The solver applies a fully connected layer W on the CNN's output to produce an affinity matrix $A \in \mathbb{R}^{N \times N}$, where A_{ij} describes the CNNs preference strength for assigning the i^{th} input puzzle location to the j^{th} target location. It then infers the most likely global assignment of jigsaw patches to output patches by applying the Sinkhorn operator to the affinity matrix $A^+ = \text{Sinkhorn}(A)$. This will complete the un-shuffle the input patches and solve the jigsaw by producing an assignment matrix with constraints: (i) all elements are either 0 or 1; (ii) each row and column has exactly one assignment. For instance, $A^+_{ij} = 1$ means assigning i^{th} input patch to the j^{th} target patch, and the mapping between input and output patches is 1-to-1.

Sinkhorn Operator $\text{Sinkhorn}(\cdot)$ To implement the Sinkhorn operator, we follow [3] and iteratively normalise its rows of the input in order to approximate the doubly stochastic matrix A^+:

$$\begin{aligned} \text{Sinkhorn}^0(A) &= \exp(A) \\ \text{Sinkhorn}^l(A) &= T_c(T_r(\text{Sinkhorn}^{l-1}(A))) \\ \text{Sinkhorn}(A) &= \lim_{l \to \infty} \text{Sinkhorn}^l(A) \end{aligned} \tag{1}$$

where $T_r(X) = X \oslash (X \mathbf{1}_N \mathbf{1}_N^T)$, $T_c(X) = X \oslash (\mathbf{1}_N \mathbf{1}_N^T X)$ as the row and column-wise normalisation operations of a matrix, with $\oslash$ denoting the element-wise division and $\mathbf{1}_N$ a column vector of ones. l is a hyper-parameter to control the number of Sinkhorn iterations used to estimate the assignment.

Loss Functions For jigsaw pre-training, our loss function aims to close the distribution gap between A^+ and the true assignment matrix Y (generated from from O), defined as:

$$\text{loss}(A^+, Y) = - \sum_{i=1}^{N} \sum_{j=1}^{N} [\log(A^+_{ij}) \times Y_{ij} + \log(1 - A^+_{ij}) \times (1 - Y_{ij})] \tag{2}$$

Summary At each iteration, training images are edge extracted, and randomly shuffled and modality mixed. Training the jigsaw solver J to un-shuffle the images requires the CNN to learn a feature extractor which is both modality invariant, and encodes enough fine-grained detail to enable the permutation solver to successfully un-shuffle.

4. FG-SBIR Fine-Tuning

In the fine-tuning stage we perform supervised learning of free-hand sketch to photo retrieval. Specifically, we strip off the permutation solver module G and use the feature extractor $F(\cdot)$ in the standard triplet ranking loss:

$$\begin{aligned}&\mathrm{loss}(s, p^+, p^-) = \\ &\max(0, \Delta + d(F(s), F(p^+)) - d(F(s), F(p^-)))\end{aligned} \tag{3}$$

where s is a query sketch, p^+ and p^- are positive and negative photo examples, $d(s, p) = ||F(s) - F(p)||_2^2$, and Δ is a hyper-parameter as the margin between the positive and negative example distance. For evaluation we retrieve the photo p with minimum distance to a query sketch s according to $d(s, p)$.

5. Experimental Settings

To pinpoint the advantages of jigsaw pre-training, we control all baselines and ablated variants to use the same CNN architecture and optimisation strategy. Learning rates and hyper-parameters are not grid-searched for optimal performance. Only training iteratiosn may vary across datasets.

Dataset and Pre-processing **For Jigsaw pre-training**: The FG-SBIR benchmarks used are the Shoe, Chair and Handbag product search datasets from [2]. For pre-training, additional photo images of the same category are collected. (1) Shoes – we take all 50,025 product images from [32]. (2) Handbags – we randomly select 50k photos from Handbag-137k [37] which is crudely crawled from Amazon without manual refinement. We filter out the ones with noisy background or irrelevant visuals, e.g., a handbag with a human model, which leaves a final size of 42,734. (3) Chair – we collect chair images from various sources to assure their diversity, including MADE, IKEA and ARGOS, and contribute 7,813 chair photos overall. We take 90% of these photos for self-supervised training, and use the rest as validation for model selection. We extract edgemaps from photos using [38]. **For Triplet fine-tuning**: We use all four publicly available product FG-SBIR datasets [2] to evaluate our methods, namely QMUL_Shoe_V1, QMUL_Shoe_V2, QMUL_Chair and QMUL_Handbag, with 419, 6,648, 297, 568 sketch-photo pairs respectively. Of these, we use 304, 5,982, 200, 400 pairs for training and the rest for testing following the same splits as in [2]. Since noticeable data bias exists between edgemaps for pre-training and sketches in fine-tuning, e.g., stroke width, blurriness, we process both sketches and edgemaps via a cleanup and simplification model [27]. We scale and centre all input images at both stages on a 256x256 blank canvas before feeding into a model. The data and code will be released soon.

Implementation Details All experiments are carried out with a base architecture $F(\cdot)$ of GoogleNet [29] running on Tensorflow with a single NVIDIA 1080Ti GPU. **For Jigsaw pre-training**: the initial learning rate is set to 1e-3 for 50k iterations and decreased to 1e-4 for another 10k with a batch size of 128. Since product images have white background, it's likely when dividing it into a $N \times N$ grid that some corner patches will be completely empty. Thus in practice, we first draw bounding boxes around the object (by simple pixel-value thresholding) in both photo and edgemap images and perform patch shuffling within them. The number of iterations l for the Sinkhorn operator is set to $5, 10, 15, 20$ for the patch number $N = 4, 9, 16, 25$ respectively. Intuitively, denser jigsaws pose more complicated un-shuffling problems and thus require more Sinkhorn iterations. To discourage overfitting to patch-edge statistics [15], we leave a random gap between the patches. For **triplet fine-tuning**: We train triplet ranking with a batch size of 16. We train 50k iterations for QMUL_Shoe_V2 and 20k iterations for the rest. The learning rate is set 1e-3 with a fixed margin value $\Delta = 0.1$. As a run-time augmentation, we also adopt the multi-cropping strategy as in [34]. In both stages, common training augmentation approaches including horizontal flipping and random cropping, as well as colour jittering are applied. `MomentumOptimizer` is used with momentum value 0.9 throughout.

Evaluation Metrics Following community's convection, FG-SBIR performance is quantified by acc@K, the percentage of sketches whose true-match photos are ranked in top K. We focus on the most challenging scenario of K=1 through our experiments. Each experiment is run five times. The mean and standard deviation of the results obtained over the five trials are then reported.

Baselines As our focus is on pre-training, our baselines consist of alternative pre-training approaches, while the final triplet fine-tuning is kept the same throughout. **Counting** [16] and **Rotation** [9]: These are two popular self-supervised alternatives to Jigsaws. The former asks for the total number of visual primitives in each split tile to equate that in the whole image. The latter requires the model to recognise the 2d rotation applied to an image. We found the common 2x2 split for learning to count may seemingly suffice for categorisation purpose, but empirically too coarse for fine-grained matching. Therefore in our implementation, we enhance it to count within 3x3 split, which is equivalent to training a 11-way Siamese network (9 tiles + 1 original image + 1 contrastive negative image[1] to circumvent trivial learning). We follow the same definition of geometric

[1]A potential shortcut is that it can easily satisfy the constraint by learning to count as few visual primitives as possible, so many entries of the feature embedding may collapse to zero without a contrastive signal.

Method	Pre-training Self-supervised?	FG-SBIR Dataset QMUL_Shoe_V1$^{4\times4}$	QMUL_Shoe_V2$^{3\times3}$	QMUL_Chair$^{3\times3}$	QMUL_Handbag$^{4\times4}$
Counting [16]	✓	41.74%± 2.30	30.42%± 0.54	72.78%± 4.35	54.05%± 2.77
Rotation [9]	✓	32.17%± 2.68	28.83%± 0.40	70.31%± 3.45	38.33%± 1.86
CPC [17]	✓	21.91%± 1.69	8.65%± 0.34	35.24%± 0.42	15.36%± 0.69
Matching [20]	✓	39.13%± 0.87	31.05%± 0.84	75.69%± 1.53	50.36%± 0.68
ImageNet [29]	✗	43.48%± 1.74	33.99%± 1.09	85.16%± 1.56	52.62%± 2.04
Ours/1000-way	✓	42.78%± 3.75	30.24%± 1.74	79.59%± 1.53	49.40%± 3.97
Ours/ImageNet	✗✓	48.00%± 2.91	31.26%± 0.65	79.59%± 1.34	61.07%± 1.50
Ours	✓	**56.52%± 2.75**	**36.52%± 0.84**	**85.98%± 2.01**	**62.97%± 2.04**

Table 1: Comparisons with different baselines as pre-training approaches for FG-SBIR task. The top-right superscript on each dataset name indicates the granularity of the jigsaw game solved that brings the best FG-SBIR performance respectively.

rotation set [9] by multiples of 90 degrees, i.e., 0, 90, 180, and 270 degrees, which makes a 4-way classification objective. **CPC** [17]: A state of the art self-supervised method that learns representation by predicting the future in latent space by using powerful autoregressive models. We follow the authors' implementations by predicting up to five rows from the 7x7 grid. **Matching**: This trains a triplet ranking model between an edgemap query and the positive and negative photo counterparts [20]. **ImageNet [29]**: this corresponds to the standard pre-trained 1K classification model on ImageNet, GoogleNet in our case. **Our/1000-way**: we adapt our mixed-modality jiasaw solving based model, but instead of solving it, we follow [15, 11] to solve a substitute problem of 1000-way jigsaw pattern classification. Lastly, **Ours** and **Ours/ImageNet**, two means of training our proposed method either from scratch or building upon the initialised weights of ImageNet.

6. Results and Analysis

6.1. Comparison with Baselines

Our first discovery is that self-supervised jigsaw pretraining from scratch on *target* category photos (i.e., For FG-SBIR on shoe products, collect un-annotated shoe photos for pre-training) followed by standard FG-SBIR finetuning is highly effective. Belows is more detailed analysis of the results with reference to Table 1.

Is solving a cross-modality jigsaw task a better strategy than ImageNet pre-training? Yes. It is evident that the proposed method (Ours) outperforms all the other baselines including the conventional ImageNet pre-training based one (ImageNet) on all four datasets, sometimes with significant margins. Furthermore ImageNet pre-training does not provide any benefits, but harmful when combined with our jigsaw solver (Ours/ImageNet). These results show that training for single-modality object classification is of limited relevance compared to our mixed-modal pre-training strategy.

Does the way the jigsaw puzzle is solved matter? Yes. The significant gap between Ours and Ours/1000-way confirms the significance of our technical choice: Formalising jigsaw solving as permutation estimation via Sinkhorn operator to actually solve it. This difference in efficacy is due to two reasons: (i) How to choose the pre-defined permutation set for classification determines the ambiguity of the task. Despite efforts to maximise task efficacy via evolution of classification sets [15], classifying among a fixed set of permutations is worse than our assignment matrix estimation which must select among *all possible* permutations. (ii) The Sinkhorn operator provides a direct representation and estimation of permutation, so that latent features are properly learned to support this purpose, rather than a coarse correlate to permutation.

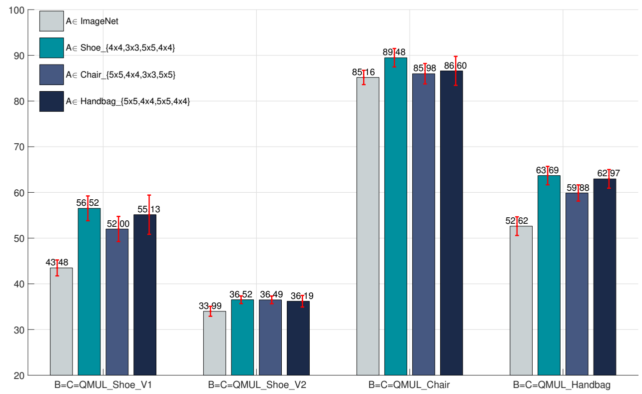
Why is Edge-photo-matching ineffective? At the first glance, training an edge-photo matching model [20] seems a natural task choice for pre-training FG-SBIR, given the similarity between edges and human sketches[2]. However, the very poor performance of the baseline (Matching) suggests that even though the edgemap is useful substitute to sketch (as demonstrated by our method), how to design the cross-modal task matters. The Edge-photo-matching task only requires whole image level photo to edgemap matching, which can be effectively solved by learning an edge detector. In contrast, our mixed-modal jigsaw puzzle problem is much harder – solving it requires the model to understand the two modalities both at the image level and the local patch level.

Why do the improvements vary across datasets? It is noted that our method exhibits a bigger margin on the shoes and handbags compared to chairs. We believe this is because overall solving jigsaw puzzles on shoes and handbags are harder than chairs due to the more complicated and diverse design styles they present, and thus better model capabilities are required and gained through the jigsaw solving pretraining stage.

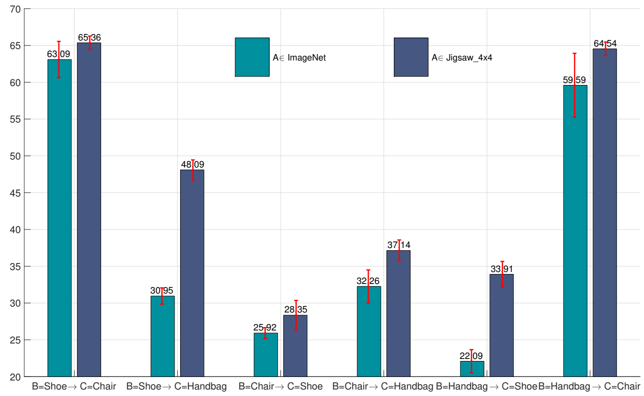
6.2. Cross-Category Generalisation of Jigsaws

Our second discovery is that models pre-trained to solve jigsaw puzzles are surprisingly generalisable. Pre-training

[2]Indeed, especially in the field of image-to-image translation, people tend to treat the terms sketch and edgemap interchangeably.



(a)



(b)

Figure 3: Cross-Category generalisation in pre-training and FG-SBIR. Symbols A, B, C refer to FG-SBIR model learning pattern A+B→C, where A represents our jigsaw training data, further fine-tuned by a triplet ranking model on category B, and finally testing on category C. We slightly abuse the notation here, as sometimes A can also be ImageNet. We use the notation = to denote using the same category for two of these stages. (a) Cross-category generalisation between jigsaw pre-training and fine-tuning/testing. Fine-tuning/testing is kept the same throughout (B=C). (b) Cross-category generalisation between pre-training/fine-tuning and testing. Pre-training/fine-tuning are kept the same throughout (A=B). Best viewed in zoom.

on one category followed by triplet fine-tuning and testing on another category is similar or sometimes even better compared with two stages within the same category.

Analysis of Jigsaw-informed Pre-training Model We first investigate the importance of having the same object category during jigsaw pre-training and triplet fine-tuning stages. From the results in Figure 3(a), we make the following observations: (i) Matching pre-training and fine-tuning category is not crucial. Indeed using the Shoe dataset for pre-training tends to provide the best performance across all four fine-tuning/testing categories. (ii) This suggests what really matters is not whether the pre-train/fine-tune categories are aligned, but the richness of each individual pre-training dataset itself. In this regard we observe Shoe>Handbag>Chair in terms of which dataset provides the most effective pre-training across a variety of target datasets. This result also coincides with our intuition that a good pre-training model should be category-agnostic. (iii) Overall, as long as pre-training uses our proposed jigsaw strategy, and is provided with a moderate sized set of product photos from any fashion category, the standard ImageNet pre-training strategy can be beaten. A key implication of these results are to provide a new route to scaling FG-SBIR systems in practice. While collecting large annotated free-hand sketch-photo pair datasets for each object category is prohibitively expensive, collecting product photos in any fashion category at large scale is quite feasible and can be used to boost FG-SBIR performance.

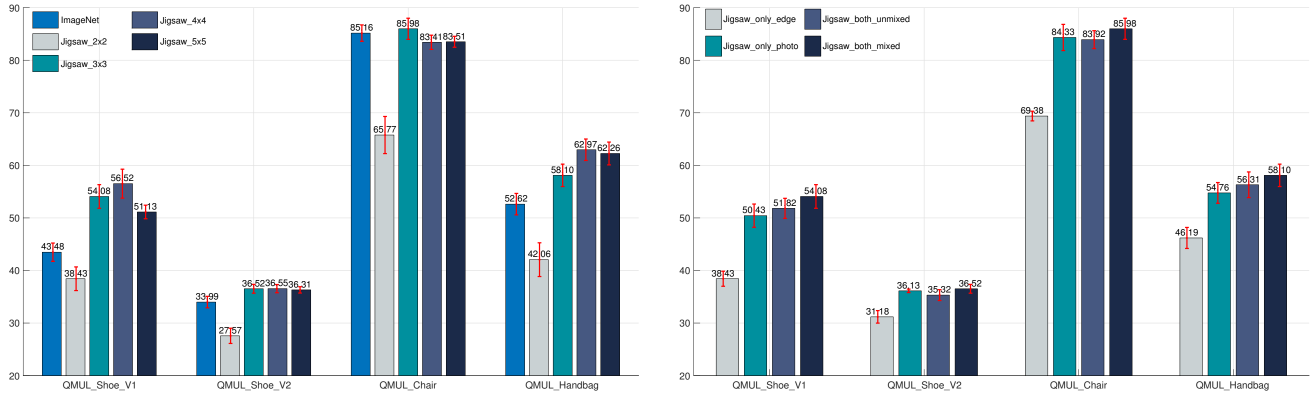
Analysis of Jigsaw-enabled FG-SBIR Model A second type of generalisability we explore is the impact of the chosen pre-training approach on the ability of the resulting FG-SBIR model to transfer across categories between training and testing. From the results in Figure 3(b), we can see that as expected, the performance drops in this cross-category testing setting compared to Figure 3(a). However, in every case Jigsaw pre-training leads to better cross-category generalisation than standard ImageNet pre-training.

6.3. Ablation Study

In this section, we compare our proposed method with a few variants to validate some key design choices in our jigsaw puzzles pre-training paradigm.

Granularity of Puzzle The difficulty of the jigsaw game depends on the granularity of the pieces shuffled for recomposition. If the granularity is very coarse, e.g., 2×2, the task is relatively simple and may not pose sufficient challenge for effective feature learning. If the granularity is very fine, e.g., 10×10, it may be too hard for even humans to solve and lead to models overfitting on noise. We explore this effect by enumerating jigsaw sizes from 2×2 to 5×5 and show the results in Figure 4(a). We make the following observations: (i) Except for 2×2, the difference in FG-SBIR results across different granularities is small and all larger jigsaws usually outperform the ImageNet baseline. (ii) The optimal granularity of jigsaw pre-training for each dataset slightly differs, but generally a puzzle of 3×3 or 4×4 provides a good choice.

Construction of Puzzle Modality Given the collected photos and extracted edgemaps of one category, there are four ways to construct the modality of the pre-training puzzles, namely: photo domain only, edgemap domain only, photo and edgemap mixed at image-level (both modalities



(a) (b)

Figure 4: Comparisons between different ablated variants of the proposed jigsaw pre-training on the performance of FG-SBIR task – (a) Granularity of the jigsaw. (b) Data modality of the image. The red error bar represents the standard deviation among the five repeated trials. More details in text. Best viewed in zoom.

of images are used, but each puzzle only contains a single randomly chosen modality), photo and edgemap mixed at patch-level (ours). We summarise the results of these variants in Figure 4(b) and draw some conclusions: (i) Although our downstream task is cross-domain, pre-training on photo domain only seemingly sufficient for quite good performance across datasets. This is in contrast to using edgemaps alone where performance plummets. (ii) Mixing photo and edgemap images into a single dataset of both modalities provides limited benefit over photo only (Jigsaw_both_unmixed). (iii) Our patch-wise mixed-modal input strategy (Jigsaw_both_mixed) leads to the best performance on all four datasets.

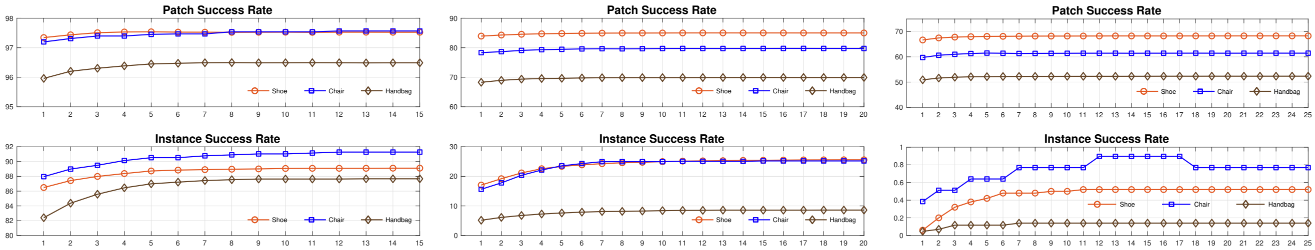
Datasets	Variants	Methods	Acc@1
QMUL_Shoe_V1	**C2FF**	ImageNet	44.57%± 1.58
		Ours$^{shoe_4\times4}$	**55.30**%±**2.27**
	HOLEF	ImageNet	44.18%± 2.25
		Ours$^{shoe_4\times4}$	**54.61**%± **1.13**
	UFG-SBIR	ImageNet	26.96%± 1.74
		Ours$^{shoe_4\times4}$	**35.30**%± **2.92**
QMUL_Chair	**C2FF**	ImageNet	83.30%± 1.85
		Ours$^{shoe_4\times4}$	**91.54**%± **1.98**
	HOLEF	ImageNet	85.77%± 2.24
		Ours$^{shoe_4\times4}$	**89.90**%± **1.34**
	UFG-SBIR	ImageNet	**72.37**%± **2.35**
		Ours$^{shoe_4\times4}$	**72.16**%± **2.53**
QMUL_Handbag	**C2FF**	ImageNet	**57.14**%± **2.59**
		Ours$^{shoe_4\times4}$	**57.38**%± **2.21**
	HOLEF	ImageNet	54.29%± 1.70
		Ours$^{shoe_4\times4}$	**63.33**%± **2.68**
	UFG-SBIR	ImageNet	32.86%± 2.03
		Ours$^{shoe_4\times4}$	**56.43** %± **0.98**

Table 2: Comparisons between our jigsaw approach and ImageNet pre-training when using different FG-SBIR variants.

7. Further Analysis

Sensitivity to Existing FG-SBIR Frameworks Thus far we have focused entirely on different pre-training approaches and datasets, while keeping a standard CNN and FG-SBIR matching architecture to facilitate direct comparison. We next examine to what extent our pre-training methods complement recent improvements in FG-SBIR method design. We consider three FG-SBIR variants, including: (i) Architecture enhancements: Coarse to fine fusion [28, 33], which we denote `C2FF`; (ii) Training objective: [28]: Triplet ranking loss with a higher order learnable energy function - `HOLEF`; (iii) Problem formulation: Unsupervised FG-SBIR - `UFG-SBIR`, where edgemap is treated as a human sketch for SBIR training [20]. From the results in Table 2, we can see that our self-supervised mixed-modal jigsaw pre-training matches or improves on ImageNet performance for each of the FG-SBIR variants tested.

The effect of Sinkhorn Iterations l In practice, there is a trade-off on selecting the value of l: if it is too small, then the resultant assignment matrix will be far from a true permutation one, while when it's unhelpfully big, the optimisation becomes harder as the gradients vanished accordingly. In Figure 5, we show how jigsaw solver reacts to the linear slicing of different values ranging from 1 to l. The following observations can be made: (i) Generally, the jigsaw model saturates when the number is approaching l, with few exceptions that best performance is gained halfway (Figure 5(c)). (ii) For many settings after one round of Sinkhorn normalisation, the jigsaw performance already reaches to a reasonable level. This implies that even if we apply l times of Sinkhorn iteration during training, the model only improve the solving success marginally, but continue to pretrain a better model. (iii) Despite failing to get instances



(a) 3x3 jigsaw puzzle (b) 4x4 jigsaw puzzle (c) 5x5 jigsaw puzzle

Figure 5: Jigsaw solver success rate vs. Sinkhorn iterations once trained under l. Patch success rate and Instance success rate refer to the percentage of the shuffled patches that are correctly ordered and the percentage of the instances where all patches within are perfectly recovered respectively. Note that since it's practically infeasible to test all possible permutations of one sample, for each subfigure, we generate one mix-modality shuffling strategy for each input and apply it to all x-axis values.

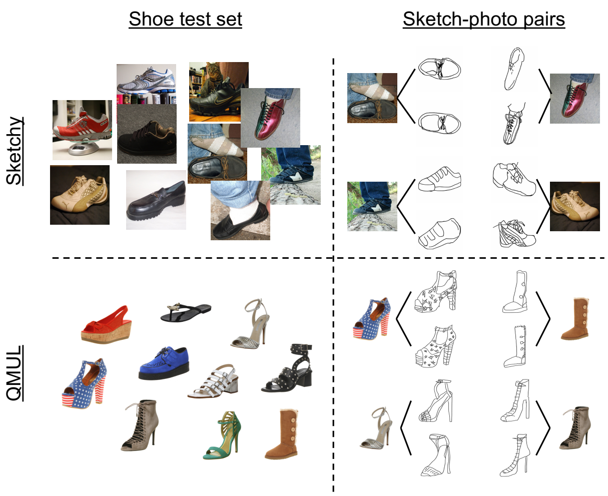


Figure 6: Illustrations of our product-level FG-SBIR dataset and the existing general-purpose counterpart, Sketchy.

perfectly un-shuffled, e.g., less than 1% on 5×5 puzzle, the solver can consistently get a large number of patches right. (iv) Different jigsaw granularities corresponds to very different scales of jigsaw success rates, in a stark contrast with that on FG-SBIR (Figure 4 (a)), where little difference is witnessed as long as the granularity exceeds 2x2.

Caveat: SBIR Dataset Flavours We note that thus far the superiority of our jigsaw pre-training is validated when applied to *product*-level FG-SBIR benchmarks because this is where FG-SBIR is most likely to be applied. Here we consider two other type of datasets: The Flickr15k [4] benchmark for *category-level* SBIR (i.e., the goal is to retrieve any instance of a particular category rather than one specific instance), and Sketchy [23], with sketch-photo paired data covering 125 real-world object categories. We follow the standard splits for these benchmarks, and evaluate our Jigsaw pre-training approach vs. the standard ImageNet pre-training in Table 3. We can see that our Jigsaw strategy is not effective for these benchmarks, and direct ImageNet pre-training clearly leads to the best results. To understand why, we show in Figure 6 the test set photos of the shoe category in Sketchy and a random 10 shoe photos in QMUL_Shoe_V2. It can been seen : (i) *Pose* and *shape* play pivotal roles in matching for sketchy, rather than fine-grained details in product-level FG-SBIR. This lesser pose variability in QMUL_Shoe_V2 contributes to the poor transferability to Sketchy. (ii) Sketchy and Flickr15k images have complicated backgrounds, unlike the white-background product images. Pre-training on product photos thus is unsurprisingly ineffective in teaching a model to deal with complex backgrounds required for Sketchy and Flickr15k. In these cases ImageNet pre-training is understandably more appropriate.

Dataset	Methods		
	Ours$^{shoe_4\times4}$	Ours$^{shoe_4\times4}$/ImageNet	ImageNet
Sketchy	53.45%± 0.28	51.86%± 0.17	**60.26%± 0.16**
Flickr15k	27.23%± 0.81	24.03%± 0.84	**44.15%± 0.30**

Table 3: Performance comparison on coarser-grained SBIR datasets. Values reported on Sketchy and Flickr15k are measured with acc@1 and mAP respectively.

8. Conclusion

We have introduced a new mixed-modal jigsaw self-supervised pre-training strategy for FG-SBIR with a novel solver. We showed that the proposed method outperforms the conventional ImageNet pre-training stage. This strategy generalises well across categories, and furthermore leads to FG-SBIR models with better cross-category generalisation properties. We hope this pre-training strategy can become the norm for future FG-SBIR work, as well as be adopted by other cross-modal retrieval/recognition tasks.

Acknowledgements: Yongxin Yang thanks his daughter – Nina Lee Yang – who inspired the idea of this work.

References

[1] Jigsaw puzzle — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Jigsaw_puzzle. 2

[2] SketchX!-Shoe/Chair Fine-grained-SBIR dataset. http://sketchx.eecs.qmul.ac.uk, 2017. 4

[3] Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011. 2, 3

[4] Tu Bui, L Ribeiro, Moacir Ponti, and John Collomosse. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *CVIU*, 2017. 8

[5] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019. 2

[6] Jia Deng, Wei Dong, R Socher, and Li Jia Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 2

[8] Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016. 2

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 4, 5

[10] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019. 2

[11] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *WACV*, 2018. 2, 5

[12] Yi Li, Timothy M Hospedales, Yi-Zhe Song, and Shaogang Gong. Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*, 2014. 2

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[14] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR*, 2018. 2

[15] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2, 4, 5

[16] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 4, 5

[17] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[18] Kaiyue Pang, Da Li, Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep factorised inverse-sketching. In *ECCV*, 2018. 2

[19] Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. In *BMVC*, 2017. 1

[20] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *ECCV*, 2018. 1, 5, 7

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[22] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018. 1

[23] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *SIGGRAPH*, 2016. 1, 2, 8

[24] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. In *CVPR*, 2017. 2

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[26] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR*, 2014. 2

[27] Edgar Simo-Serra, Satoshi Iizuka, Kazuma Sasaki, and Hiroshi Ishikawa. Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup. *SIGGRAPH*, 2016. 4

[28] 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 *ICCV*, 2017. 1, 2, 7

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4, 5

[30] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[31] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 2

[32] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 4

[33] Qian Yu, Xiaobin Chang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *arXiv preprint arXiv:1711.08106*, 2017. 7

[34] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen Change Loy. Sketch me that shoe. In *CVPR*, 2016. 1, 2, 4

[35] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[36] Jingyi Zhang, Fumin Shen, Li Liu, Fan Zhu, Mengyang Yu, Ling Shao, Heng Tao Shen, and Luc Van Gool. Generative domain-migration hashing for sketch-to-image retrieval. In *ECCV*, 2018. 1, 2

[37] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 4

[38] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 4