MosAIc: Finding Artistic Connections across Culture with Conditional Image Retrieval

Mark Hamilton[1,2], **Stephanie Fu**[2], **Mindren Lu**[2], **Johnny Bui**[2], **Darius Bopp**[2], **Zhenbang Chen**[2], **Felix Tran**[2], **Margaret Wang**[2], **Marina Rogers**[2], **Lei Zhang**[1], **Chris Hoder**[1], **William T. Freeman** [2,3]
[1]*Microsoft,* [2]*MIT,* [3]*Google*

Editors: Hugo Jair Escalante and Katja Hofmann

Figure 1: Conditional image retrieval results on artwork from the Metropolitan Museum of Art and Rijksmuseum using media and culture (text above images) as conditioners.

Abstract

We introduce MosAIc, an interactive web app that allows users to find pairs of semantically related artworks that span different cultures, media, and millennia. To create this application, we introduce Conditional Image Retrieval (CIR) which combines visual similarity search with user supplied filters or "conditions". This technique allows one to find pairs of similar images that span distinct subsets of the image corpus. We provide a generic way to adapt existing image retrieval data-structures to this new domain and provide theoretical bounds on our approach's efficiency. To quantify the performance of CIR systems, we introduce new datasets for evaluating CIR methods and show that CIR performs non-parametric style transfer. Finally, we demonstrate that our CIR data-structures can identify "blind spots" in Generative Adversarial Networks (GAN) where they fail to properly model the true data distribution.

Keywords: Image Retrieval, Search, GANs, KNN, Ball Trees, Style Transfer, Art, Reverse Image Search

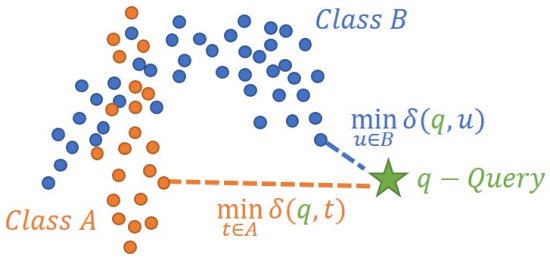

Figure 2: Conditional K-Nearest Neighbors for a query point, q, and distance, δ, on a simple two class dataset.

Component	Space Efficiency	Measured
Data	$\mathcal{O}(n \times d)$	16 GB
Tree	$\mathcal{O}((2n/l) \times d)$	65 MB
Cond. Index	$\mathcal{O}(c \times 2n/l)$	6.4 MB

Table 1: Space efficiency of a binary CKNN Tree with number of points, n, dimensionality, d, leaf size l, and number of classes in the index, c. Measured results are from a tree built on the Conditional Art dataset: $n = 1000000, d = 2048, l = 500, c = 200$.

1. Introduction

In many Image Retrieval (IR) applications, it is natural to limit the scope of the query to a subset of images. For example, returning similar clothes by a certain brand, or similar artwork from a specific artist. Currently, it is a challenge for IR systems to restrict their attention to sub-collections of images on the fly, especially if the subset is very distinct from the query image. This work explores how to create image retrieval systems that work in this setting, which we call "Conditional Image Retrieval" (CIR). We find that CIR can uncover pairs of artworks within the combined open-access collections of the Metropolitan Museum of Art (Met, 2019) and the Rijksmuseum (Rij, 2019) that have striking visual and semantic similarities despite originating from vastly different cultures and millennia and introduce an interactive web app MosAIc (`www.aka.ms/mosaic`) to demonstrate the approach. To understand our methods better, we evaluate CIR on the FEI Face Database (Thomaz and Giraldi, 2010) as well as two new large-scale image datasets that we introduce to help evaluate these systems. These experiments show that CIR can perform a non-parametric variant of "style transfer" where neighbors in different subsets have similar content but are in the "style" of the target subset of images.

We also investigate ways to improve IR system performance in the conditional setting. One challenge current systems face is that a core component of many IR systems, K-Nearest Neighbor (KNN) data-structures, only support queries over the entire corpus. Restricting retrieved images to a particular class or filter requires filtering the "unconditional" query results, switching to brute force adaptively (Matsui et al., 2018), or building a new KNN data-structure for each filter. The first approach is used in several production image search systems (DeGenova, 2017; Bing, 2017; Mellina, 2017), but can be costly if the filter is specific, or the query image is far from valid images. Switching to brute force adaptively can mitigate this problem but is limited by the speed of brute force search, and its performance will degrade if the target subset far from the query point. Finally, maintaining a separate KNN data-structure for each potential subset of the data is costly and can result in 2^n data-structures, where n is the total number of images. In this work, we show that tree-based data-structures provide a natural way to improve the performance of CIR. More specifically, we prove that Random Projection Trees (Dasgupta and Freund, 2008) can flexibly adapt to subsets of data through pruning. We use this insight to design a modification to existing tree-based KNN methods that allows them to quickly prune their structure to adapt to any subset of their original data using an inverted index. These structures outperform the commonly used CIR heuristics mentioned above. Finally, we investigate the structure of conditional KNN trees to show that they can reveal

Figure 3: A pair of cross cultural images found with CIR. Left: *Model Paddling Boat* from 1980 BC Egypt. Right: *Immortal Raft* from 18th Century China.

areas of poor convergence and diversity ("blind spots") in image based GANs. We summarize the contributions of this work as follows:

- We introduce an interactive web application to discover connections across cultures, artists, and media in the visual arts.
- We prove an efficiency lower bound for solving CIR with pruned Random Projection trees.
- We contribute a strategy for extending existing KNN data-structures to allow users to efficiently filter resulting neighbors using arbitrary logical predicates, enabling efficient CIR.
- We show that CIR data-structures can discover "blind spots" where GANs fail to match the true data.

2. Background

IR systems aim to retrieve a list of relevant images that are related to a query image. "Relevance" in IR systems often refers to the "semantics" of the image such as its content, objects, or meaning. Many existing IR systems map images to "feature space" where distance better corresponds to relevance. In feature space, KNN can provide a ranked list of relevant images (Manning et al., 2008). Good features and distance metrics aim to align with our intuitive senses of similarity between data (Yamins et al., 2014) and show invariance to certain forms of noise (Gordo et al., 2016). There is a considerable body of work on learning good "features" for images (Bengio et al., 2013; Zhang et al., 2016; Radford et al., 2015; Koch et al., 2015; Huh et al., 2016). In this work we leverage features from intermediate layers of deep supervised models, which perform well in a variety of contexts and are ubiquitous throughout the literature. Nevertheless, our methods could apply to any features found in the literature including those from collaborative filtering, text, sound, and tabular data.

There are a wide variety of KNN algorithms, each with their own strengths and weaknesses. Typically, these methods are either tree-based, graph-based, or hash-based (Aumüller et al., 2018). Tree-based methods partition target points into hierarchical subsets based on their spatial geometry and include techniques such as the KD Tree (Bentley, 1975), PCA Tree (Bachrach et al., 2014), Ball Tree (Omohundro, 1989), some inverted index approaches (Baranchuk et al., 2018), and tree ensemble approaches (Yan et al., 2019). Some tree-based data-structures allow exact search with formal guarantees on their performance (Dasgupta and Freund, 2008). Graph-based methods rely on

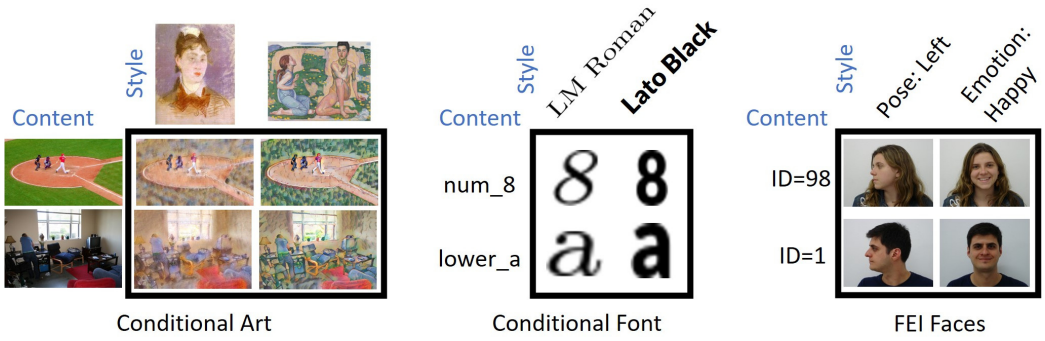

Figure 4: Representative samples from the ConditionalArt dataset (left), ConditionalFont dataset (middle), and FEI Face dataset (right). CIR systems conditioned on style should retrieve images of the same content.

greedily traversing an approximate KNN graph of the data, and have gained popularity due to their superior performance in the approximate NN domain (Aumüller et al., 2018; Johnson et al., 2019). There are many hash-based approaches in the literature and Wang et al. (2014) provides a systematic overview. To our knowledge, neither graph nor hash-based retrieval methods can guarantee finding the nearest neighbor deterministically. However, fast approximate search is often sufficient for many applications. In our work we focus on tree-based methods because it is unclear how to create an analogous method for graph-based data-structures. Nevertheless, tree-based methods are widely used, especially when exact results are needed. Surprisingly, conditional KNN systems have only received attention recently, even though conditional queries appear in shopping, search, and recommendation systems. To our knowledge, (Matsui et al., 2018) is the only effort to improve performance of these systems by adaptively switching from a "query-then-filter" strategy to brute-force at a particular size threshold.

3. Conditional Image Retrieval

To generalize an IR system to handle queries over any image subset we generalize the KNN problem to this setting. More formally, the Conditional K-Nearest Neighbors (CKNNs) of a query point, q, are the k closest points with respect to the distance function, δ, that satisfy a given logical predicate (condition), $\mathcal{S}$. We represent this condition as a subset of the full corpus of points, $\mathcal{X}$:

$$CNN(q, \mathcal{S} \subseteq \mathcal{X}) = \underset{t \in \mathcal{S}}{\operatorname{argmin}}\, \delta(q, t)$$

When the conditioner, $\mathcal{S}$, equals the full space, $\mathcal{X}$, we recover the standard KNN definition. Figure 2 shows a visualization of CKNN for a two-dimensional dataset with two classes. With conditional KNN queries it's possible to combine logical predicates and filters with geometry-based ranking and retrieval.

To create a CIR system, one can map images to a "feature-space", where distance is semantically meaningful, prior to finding CKNNs. One of the most common featurization strategies uses the penultimate activations of a supervised network such as ResNet50 (He et al., 2016) trained on ImageNet (Deng et al., 2009). Alternatively, using "style" based features from methods like AdaIN

		Featurization Method								
Dataset	Metric	RN50	RN101	MN	SN	DN	RNext	dlv3101	MRCNN	Random
CA	@1	.50	.51	.55	.44	**.59**	.46	.37	.45	.0002
	@10	.70	.68	.71	.62	**.76**	.65	.55	.63	.002
CF	@1	.41	.37	.39	**.44**	.43	.38	.33	**.44**	.016
	@10	.77	.76	.76	**.80**	.79	.76	.73	.79	.16
FEI	@1	.80	.84	.85	.79	**.87**	.86	.72	.78	.005
	@10	.94	.93	.94	.89	**.95**	.94	.86	.92	.05

Table 2: Performance of CIR (Accuracy @N) on content recovery across style variations for both the ConditionalFont (CF) and ConditionalArt (CA) datasets using a variety of features from pre-trained networks. Results show CIR retrieves the same content image across different styles. For full details on experimental conditions see Section 11

.

(Huang and Belongie, 2017) enable CIR systems that retrieve images by "style" as opposed to content. Deep features capture many aspects of image semantics such as texture, color, content, and pose (Olah et al., 2017) and KNNs in deep feature space are often both visually and semantically related. We aim to explore whether this observation holds for conditional matches across disparate subsets of images, which requires a more global feature-space consistency.

4. Discovering Shared Structure in Visual Art

We find that CIR on the combined Met and Rijksmusem collections finds striking connections between art from different histories and mediums. These matches show that even across large gaps in culture and time CIR systems can find relevant visual and semantic relations between images. For example, Figure 3 demonstrates a pair of images that, despite being separated by 3 millennia and 7,000 Kilometers, have an uncanny visual similarity and cultural meaning. More specifically, both works play a role in celebrating and safeguarding passage into the afterlife (Werner, 1922; Oppenheim et al., 2015; Hayes, 1990). Matches between cultures also highlight cultural exchange and shared inspiration. For example, the similar ornamentation of the Dutch Double Face Banyan (left) and the Chinese ceramic figurine (top row second from left) of Figure 1 can be traced to the flow of porcelain and iconography from Chinese to Dutch markets during the 16^{th}-20^{th} centuries (Le Corbeiller, 1974; Volker, 1954). CIR also provides a means for diversifying the results of visual search engines through highlighting conditional matches for cultures, media, or artists that are less frequently explored. We hope CIR can help the art-historical community and the public explore new artistic traditions. This is especially important during the COVID-19 pandemic as many cultural institutions cannot accept visitors. To this end, we introduce an interactive art CIR application, `aka.ms/mosaic`, and provide more details in Section 5. In Section B of the Appendix we also provide additional examples and representative samples.

5. The MosAIc Web Application

As an application of CIR for the public, we introduce MosAIc (`aka.ms/mosaic`), a website that allows users to explore art matches conditioned on culture and medium. Our website aims to

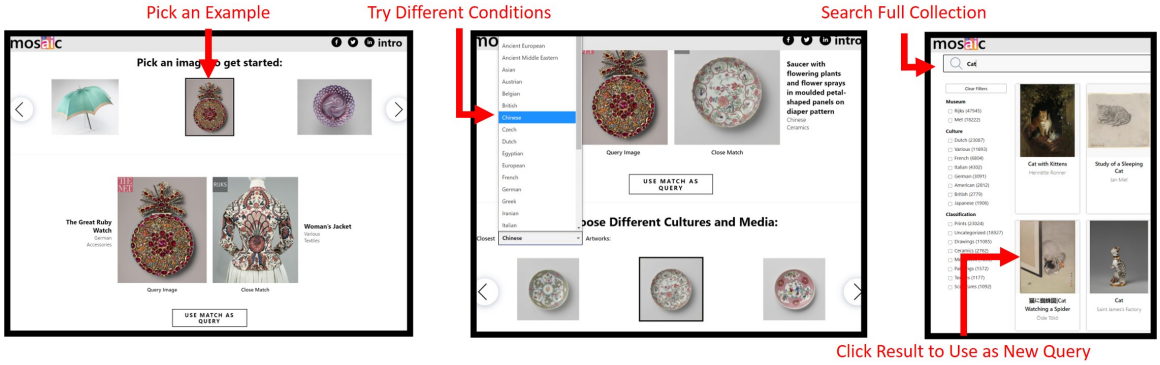

Figure 5: Using the MosAIc web application (aka.ms/mosaic). After watching a short video explaining the app, users can select a work of art to find conditional matches with (left). Users can find conditional matches for a variety of different cultures and media (middle). To further explore the collection, users can search for new query objects using a conventional search index (right). Users can also construct chains of conditional matches using the "Use match as query" button below the main matches.

show how conditional image retrieval can find surprising and uncanny pairs of artworks that span millennia. We also aim to make it easy for interested users to find new artworks in cultures they might not think to explore during a physical museum visit. Using the MosAIc application, users can choose from a wide array of example objects to use as conditional search queries as shown in the left panel of Figure 5. Users can select from an array of different cultures and media to condition their searches as in Figure 5 middle. Selecting a specific medium or culture, allows the user to browse the top conditional matches in that category and use these matches as new query images. This enables traversing the collection using conditional searches to find relevant content in different areas of the collection. Additionally, for users who want to use a specific work of art as a starting point we have added a conventional text based search engine to quickly find specific works relating to a keyword as in Figure 5 right.

The mosaic application combines a React (Fedosejev, 2015) front-end with a back-end built from Azure Kubernetes Service, Azure Search, and Azure App Services. Our front-end features responsive design principles to support for mobile, tablet, desktop, and ultra-wide displays. We also aim to use high-contrast design to make the application more accessible to the low-vision community. To create the conditional search index, we featurize the combines Metropolitan Museum of Art and Rijksmuseum open access collections using ResNet50 from torchvision Marcel and Rodriguez (2010). We then add these features to a Conditional Ball tree for real-time conditional retrieval and deploy this method as a RESTful service on Azure Kubernetes Service. Additionally, we store image metadata, automatically generated image captions, and detected objects in an Azure Search index which allows querying for additional information, and supports text search. To add captions and detected objects to over 500k images we use the Cognitive Services for Big Data Hamilton et al. (2020).

6. Evaluating CIR Quality

Though finding connections between art is of great importance to the curatorial and historical communities, it is difficult to measure a system's success on this dataset as there are no ground truth on which images *should* match. To understand the behavior of CIR systems quantitatively we investigate datasets with known content images aligned across several different "styles" or subsets to retrieve across. More specifically, if the conditioning information represents the image "style" and the features represent the "content", CIR should find an image with the same content, but constrained to the style of the conditioner, such as "Ceramic" or "Egyptian" in Figure 1. Through this lens, CIR systems can act as "non-parametric" style transfer systems. This approach differs from existing style transfer and visual analogy methods in the literature (Huang and Belongie, 2017; Gatys et al., 2016) as it does not generate new images, but rather it finds analogous images within an existing corpora.

To this end, we apply CIR to the FEI face database of 2800 high resolution faces across 200 participants and 14 poses, emotions, and lighting conditions. We also introduce two new datasets with known style and content annotations: the ConditionalFont and ConditionalArt datasets. The ConditionalFont dataset contains 15687 32×32 grayscale images of 63 ASCII characters (content) across 249 fonts (style). The ConditionalArt dataset contains 1,000,000 color images of varying resolution formed by stylizing 5000 content images from the MS COCO (Lin et al., 2014) dataset with 200 style images from the WikiArt dataset (Nichol, 2016) using an Adaptive Instance Normalization (Huang and Belongie, 2017). Although this dataset is "synthetic", (Jing et al., 2019) show that neural style transfer methods align with human intuition. We show representative samples from each dataset in Figure 4.

With these datasets it's possible to measure how CIR features, metrics, and query strategies affect CIR's ability to match content across styles. To measure retrieval accuracy, we sampled 10000 random query images. For each random query image, we use CIR to retrieve the query image's KNNs conditioned on a random style. We then check whether any retrieved images have the same content as the original query image. In Table 2, we explore how the choice of featurization algorithm affects CIR systems. All methods outperform the random baseline of Table 2, indicating that they are implicitly performing non-parametric content-style transfer. DenseNet (DN) (Iandola et al., 2014) and Squeezenet (SN) (Iandola et al., 2016) tend to perform well across all datasets. CIR performs well across all three tasks *without* fine tuning to the structure of the datasets, indicating that this approach can apply to other zero-shot image-to-image matching problems.

7. Fast CKNN with Adaptive Tree Pruning

In Section 6 we have shown that CIR is semantically meaningful in several different contexts, but the question remains as to whether this approach affords an efficient implementation that can scale to large datasets with low latency. Conventional IR systems scale to this setting using dedicated data-structures such as trees, spatial hashes, or graphs. There are a wide variety of strategies with provable guarantees in the unconditional setting, but it is not known if existing data-structures can apply naturally to the conditional setting. In this work we focus on extending tree-based methods to the conditional setting. Tree-based methods are some of the only methods that guarantee *exact* KNN retrieval, and there are already several theoretical results on the performance of these methods (Dasgupta and Freund, 2008; Dhesi and Kar, 2010). In particular, (Dasgupta and Freund, 2008) show that RandomProjection-Max (RP) trees can adapt to the intrinsic dimensionality of the data

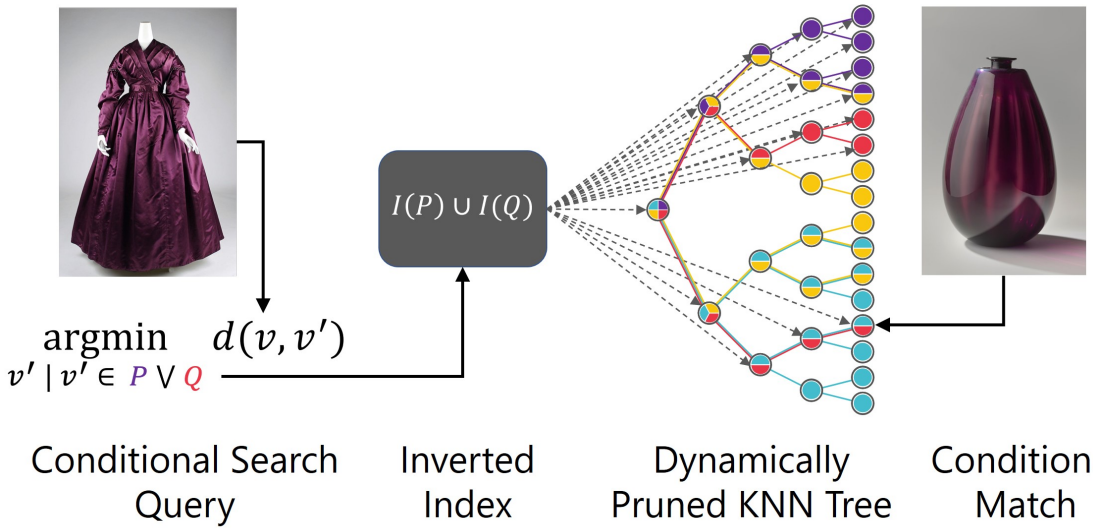

Figure 6: Dynamic tree pruning based CIR architecture. The user specified condition, $P \vee Q$, is translated to an inverted index query and the result is used to prune the unconditional KNN tree where nodes are colored based on which conditions they contain. This pruned tree accelerates conditional search for any subset by reducing the number of nodes considered in tree traversal.

and prove bounds that demonstrate the effectiveness of the data-structure. (Dhesi and Kar, 2010) continue this line of reasoning and prove a packing lemma using a bound on the aspect ratio of RP tree cells. These works show that RP trees are effective at capturing the geometry of the training data. Our aim is to show that they also capture the geometry of *subsets* of the training data through their sub-trees. More specifically, we show that for any subset of the training data, one can derive probabilistic bound the number of nodes in the tree that contain this subset. More formally:

Theorem 1 *Suppose an* RPTREE-MAX, $\mathcal{T}$, *is built using a dataset* $\mathcal{X} \subset \mathbb{R}^D$, *of diameter* W, *with doubling dimension* $\leq d$. *Further suppose* $\mathcal{T}$ *is balanced with a cell-size reduction rate bounded above by* γ. *Let* $\mathcal{S} \subseteq \mathcal{X}$ *be a subset of the dataset used to build the tree and* $\mathcal{B}$ *a finite set of radius* $R > 0$ *balls that cover* $\mathcal{S}$. *For every* $0 < \epsilon < 1$ *there exists a constant,* $c > 0$, *such that with probability* $> 1 - \epsilon$ *the fraction of cells that contain points within* $\mathcal{S}$ *is bounded above by* $|\mathcal{B}|2^{-log_\gamma(W/R')}$ *where* $R' = cRd\sqrt{d}\log(d)$

We point readers to (Dhesi and Kar, 2010), for the precise definition of an RPTree, cell-size, and the doubling dimension. To sketch the proof, we first generalize an aspect bound from (Dhesi and Kar, 2010) to show that, with high probability, small radius balls can be completely inscribed within small radius RP tree cells. Because it takes several levels before the tree's cells shrink to this size, we can bound this cell's depth and thus the size of its sub-tree relative to the full tree. By considering a collection of balls that cover our target subset, we arrive at the final bound. See section C of the Appendix for a full proof.

This theorem not only shows that sub-trees of an RP tree capture the geometry of training dataset subsets, but also points to a method to improve the speed of CKNN. Namely, we can prune tree nodes that do not hold points within our target subset prior searching for conditional neighbors.

Algorithm 1: Querying a CKNN Tree

input : A point, q, a condition, $\mathcal{S} \subseteq \mathcal{X}$, a tree, $root$, and an inverted index, I
output: Closest point, $p^* \in \mathcal{S}$, to q
$validNodes \leftarrow \bigcup_{s \in \mathcal{S}} I(s)$; $p^* \leftarrow$ null
def `SearchNode(`n`)`**:**
  if $n \in validNodes$ **then**
    if n *is a leaf node* **then**
      $p \leftarrow$ closest point in $\mathcal{S}$
      if $d(p, q) < d(p^*, q)$ **then**
        $p^* \leftarrow p$
      end
    else
      $potentials \leftarrow$ children of n which could hold a closer point
      for *child in* $potentials$ **do**
        `SearchNode(`*child*`)`
      end
    end
  end
`SearchNode(`*root*`)`; **return** $p*$

We diagram this procedure in Figure 6, and provide pseudo-code in Algorithm 1. We now turn our attention to quickly computing the proper sub-trees for each subset of the data. To this end, one can use an inverted index (Knuth, 1997), I, that maps points, $x \in \mathcal{X}$ to the collection of their dominating nodes, $I(x) = \{n : x \text{ below node } n\}$. One can compute the subset of nodes that remain after pruning by taking the union of dominating nodes as shown in the first line of Algorithm 1 and in the illustration of the full search architecture in Figure 6. Evaluating the predicate on points within leaf nodes can also reduce computation.

Additionally, if the predicates of interest have additional structure, such as representing class labels, one can define a smaller class-based inverted index, $I_{class}(c)$ which maps a class label, c, to the set of dominating nodes. For these predicates, union and intersection operators commute through the class-based inverted index:

$$\begin{aligned} I(\mathcal{S}_a \cap \mathcal{S}_b) &= I_{class}(a) \cap I_{class}(b) \\ I(\mathcal{S}_a \cup \mathcal{S}_b) &= I_{class}(a) \cup I_{class}(b) \end{aligned} \tag{1}$$

where $\mathcal{S}_a$ is the subset of points with label a. This principle speeds a broad class of queries and accelerates document retrieval frameworks like ElasticSearch (Gormley and Tong, 2015) and its backbone, Lucene (McCandless et al., 2010). We stress that this approach does not use Lucene to filter images or documents directly, but rather to filter nodes of a KNN retrieval data-structure at query time. This enables a rich "predicate push-down" (Hellerstein and Stonebraker, 1993; Levy et al., 1994) logic for KNN methods independent of how the tree splits points (Ball, Hyperplane, Cluster), the branching factor, and the topology of the tree. It also applies to ensembles of trees and to multi-probe LSH methods by pruning hash buckets. We note that our proposed indexing structure is small compared to the size of the underlying dataset, and unconditional KNN tree, and provide an analysis of memory footprints in Table 1.

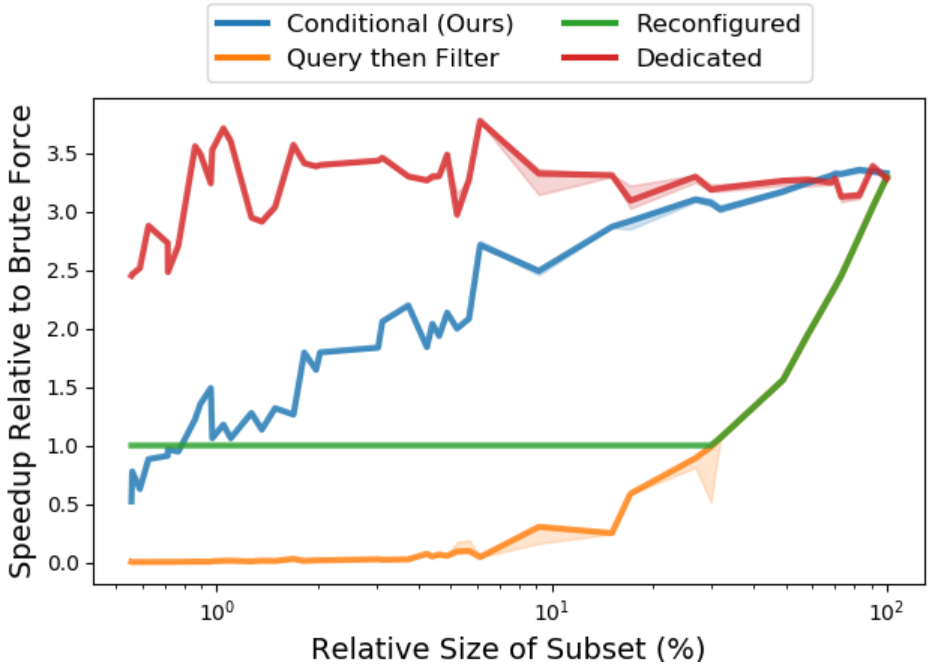

Figure 7: Query time of Conditional KNN approaches. Our approach (Conditional) achieves query performance approaching that of an tree recreated specifically for each query (Dedicated) *without the expensive re-creation cost*, and does not perform poorly with small conditions like "Query then Filter" strategies. Furthermore, our method accelerates queries across much smaller subsets than the reconfiguration strategy of (Matsui et al., 2018). Please see Section 11 and 7 for method details.

8. Performance

In Figure 7, we show the relative performance of several strategies for CIR on 488k Resnet50-featurized images ($dim = 2048$) from the combined MET and Rijksmusem open-access collections with a randomly chosen test set ($n = 1000$). We condition on artwork media, culture, and several combinations of these to create a variety of condition sizes. We measure the speedup compared to a vectorized Brute-Force search using NumPy arrays (Walt et al., 2011). We implement CKNN methods with respect to one of the most used implementations of KNN, Sci-kit Learn's Ball Tree algorithm (Pedregosa et al., 2011). We compare our approach (Conditional) to, the standard "query-then-filter" approach, and adaptive switching to brute force search (Reconfigured) (Matsui et al., 2018). Finally, we compare to a "best-case" scenario of a KNN data-structure pre-computed for every subset (Dedicated). Though in practice it is often impossible to make an index for each subset, this setting provides an upper bound on the performance of any approach. Our analysis shows that adaptive pruning (Conditional) outperforms other approaches and is close to optimal for large subsets of the dataset. Additionally, the performance of the "Query-then-filter" strategy quickly degrades for small subsets of the dataset as expected. Our approach is also compatible with prior work on adaptively switching to brute force and allows one to set the "switch-point" over 10x lower. We also note that these results hold with randomized conditions, and across other similar datasets.

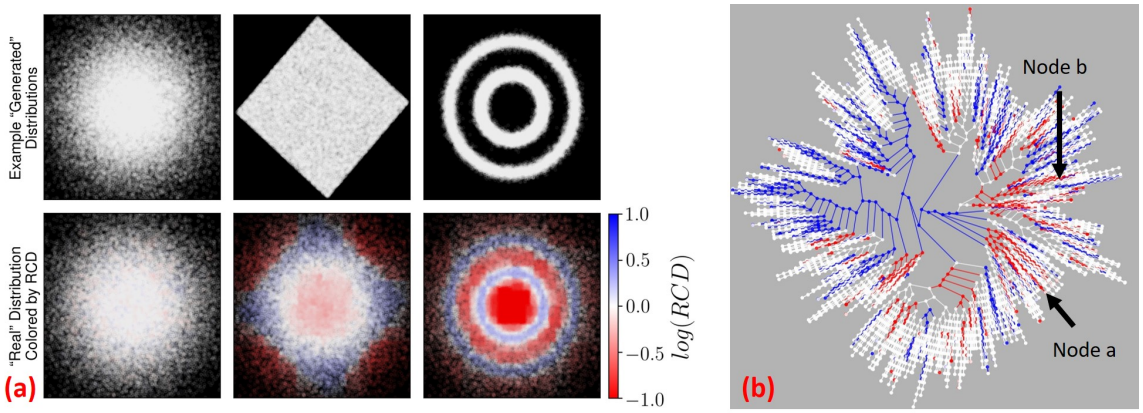

Figure 8: (a): Visualization of the RCD between several example distributions and a standard normal of "real" data ($n = 50k$). Upper plots show generated distributions, and bottom plots show the "real" distribution colored by the RCD induced by a CKNN Tree. Even though these datasets are identical under the popular Frechét inception distance (FID), the RCD detects areas where generated data over (blue) and under (red) samples the real data. (b): Nodes of a CKNN tree (Center node is the root) colored by statistically significant deviations of RCD from 1 ($p < 0.01$). This shows widespread differences between GAN outputs and true data. Red nodes represent areas where the GAN under samples the empirical distribution, and blue nodes over-sample. High discrepancy nodes a and b from Figure 9 are annotated.

Finally, we stress that the goal of this work is **not** to make the fastest unsupervised KNN method, but rather to evaluate generic strategies to transform these approaches to the conditional setting. There is a considerable body of work on fast, *approximate, unconditional* KNN methods which often outperform Scikit Learn's exact retrieval algorithms. We point readers to (Aumüller et al., 2018) for more details. We stress that exhaustive benchmarking of unconditional KNN indices and approaches is outside the scope of this work. For implementation, experimentation, environment, and computing details please see Section 11.

8.1. Implementation

We implement adaptive tree pruning for the existing Ball Tree and KD tree implementations in the popular SciKit-learn framework. Our implementation supports exact retrieval with several metrics, OpenMP parallelization (Dagum and Menon, 1998), and Cython acceleration(Behnel et al., 2011). We also provide accelerations such as dense bit-array set operations, and caching node subsets on repeated conditioner queries. For larger scale datasets, we contribute a Spark based implementation of a Conditional Ball Tree to Microsoft ML for Apache Spark (Hamilton et al., 2018a,b).

To enable integration with differentiable architectures common in the community, we provide a high-throughput, PyTorch module (Paszke et al., 2017) for CIR. This implementation is fundamentally brute force but uses Einstein-summation to retrieve conditional neighbors for multiple queries and multiple conditions simultaneously, and can increase throughput by over 100x compared to naive PyTorch implementations.

9. Limitations

This work does not aim to create the fastest KNN algorithm, but rather presents a formally motivated technique to speed up existing tree-based KNN methods in the conditional setting. KNN retrieval chooses some items significantly more than others, due to effects such as the "hubness problem" and we direct readers to (Dinu et al., 2014) for possible solutions. We present additional diversity reducing geometries in Section A of the Appendix. Our approach does not modify the KNN construction, simply prunes it afterwards. This may not be the most efficient solution when conditioner sizes are small, but it is orders of magnitude faster than recreating the tree. We also note that the performance of our conditional KNN methods are dependent on the underlying unconditional KNN tree, which often performs better on datasets with smaller intrinsic dimension.

10. Discovering "Blind Spots" in GANs

Efficient high-dimensional KNN search data-structures adapt to the geometry and intrinsic dimensionality of the dataset (Dasgupta and Freund, 2008; Dhesi and Kar, 2010). Moreover, some recent KNN methods use approaches from unsupervised learning like hierarchical clustering (Wang, 2011) and slicing along PCA directions (Bachrach et al., 2014). In this light, CKNN trees allow us to measure and visualize the "heterogeneity" of conditioning information within a larger dataset. More specifically, by analyzing the relative frequency of labels within the nodes of a CKNN tree, one can find areas with abnormally high and low label density. More formally, we introduce the Relative Conditioner Density (RCD) to measure the degree of over or under representation of a class c with corresponding subset $\mathcal{S}_c \subseteq \mathcal{X}$, at node n in the KNN tree:

$$RCD(n, c) = \frac{|n \cap \mathcal{S}_c|}{|n|} \frac{|\mathcal{X}|}{|\mathcal{S}_c|} \tag{2}$$

Here, $|n|$ is the number of points below node n in the tree. The RCD measures how much a node's empirical distribution of labels differs from that of the full dataset. $RCD > 1$ occurs when the node over-represents class c, and $RCD < 1$ occurs when the node under-represents a class, c. We apply this statistic to understand how samples from generative models, such as image-based GANs, differ from true data. In particular, one can form a conditional tree containing true data and generated samples, each with their own classes, c_t and c_g respectively. In this context, nodes with $RCD(\cdot, c_g) \ll 1$ are regions of space where the network under-represents the real dataset. To illustrate this effect, Figure 8a shows several simple 2d examples. Even though these datasets are identical with respect to the Fréchet Distance (Heusel et al., 2017), coloring points based on their parent node RCD's can highlight areas of over and under sampling of the true distribution by each "generated" distribution. In Figure 8b, we form a CKNN tree on samples from a trained Progressive GAN (Karras et al., 2017) and it's training dataset, CelebA HQ (Liu et al., 2015). Coloring the nodes by RCD reveals a considerable amount of statistically significant structural differences between the two distributions. By simply thresholding the RCD (< 0.6), we find types of images that GANs struggle to reproduce. We show samples from two low-RCD nodes in Figure 9 and also note their location in Figure 8b. Within these nodes, Progressive GAN struggles to generate realistic images of brimmed hats and microphones. Though we do not focus this work on thoroughly investigating issues of diversity in GANs, this suggests GANs have difficulty representing data that is not in the majority. This aligns with the findings of (Bau et al., 2019), without requiring GAN inversion, additional object detection labels, or a semantic segmentation ontology. Furthermore, we note that

Figure 9: Samples from two statistically significant nodes from Figure 8. Images are randomly chosen and representative of those found at the node. Almost every real image in Node a contains microphones whereas no GAN generated outputs could create a microphone. Node b shows a clear bias towards brimmed hats, and the GAN samples have significant visual artifacts.

the FID cannot capture the full richness of why two distributions differ, as this metric just measures differences between high dimensional means and co-variances. Using CKNN trees can offer more flexible and interpretable ways to understand the differences between two high dimensional distributions.

11. Experimental Details

All experiments use an Ubuntu 16.04 Azure NV24 Virtual Machine with Python 3.7 and scikit-learn v0.22.2 (Pedregosa et al., 2011). We use scikit-learn's Ball Tree and KD Tree and use numpy v1.18.1 (Walt et al., 2011) for brute force retrieval. For query-then-filter strategies we first retrieve 50 points, then increase geometrically (x5) if the query yeilds no valid matches. To form image features for Table 2, we use trained networks from torchvision v0.6 (Marcel and Rodriguez, 2010). In particular, we use ResNet50 (RN50) (He et al., 2016), ResNet101 (RN101), MobileNetV2 (MN) (Sandler et al., 2018), SqueezeNet (SN) (Iandola et al., 2016), DenseNet (DN) (Iandola et al., 2014), ResNeXt (RNext) (Xie et al., 2016), DeepLabV3 ResNet101 (dlv3101) (Chen et al., 2017), and Mask R-CNN (MRCNN) (He et al., 2017). Features are taken from the penultimate layer of the backbone, and the matches of Table 2 are computed with respect to cosine distance. We use trained a Progressive GAN from the open-source Tensorflow implementation accompanying (Karras et al., 2017).

12. Related Work

Image retrieval and nearest neighbor methods have been thoroughly studied in the literature, but we note that the conditional setting has only received attention recently. There are several survey works on KNN retrieval, but they only mention unconditional varieties (Bhatia et al., 2010; Wang et al., 2014). (Marchiori, 2009) has studied the mathematical properties of conditional nearest neighbor classifiers but works primarily with graph based methods as opposed to trees. They do not apply this to modern deep features and do not aim to improve query speed. There are a wide variety featurization strategies for IR systems. Gordo et. al (Gordo et al., 2016) learn features optimized for IR. Siamese networks such as FaceNet embed data using tuples of two data and a

similarity score and preserving this similarity in the embedding (Koch et al., 2015; Schroff et al., 2015). Features from these methods could improve CIR systems. Conditional Similarity Networks augment tuple embedding approaches with the ability to handle different notions of similarity with different embedding dimensions (Veit et al., 2016). This models conditions as similarities but does not generically restrict the search space of retrieved images to match a user's query. These features have potential to yield neighbor trees that, when pruned, have a similar structure and performance to dedicated trees. Sketch-based IR uses line-drawings as query-images but does not aim to restrict the set of candidate images generically (Lu et al., 2018). Style transfer (Jing et al., 2017) and visual analogies (Liao et al., 2017) yield results like our art exploration tool but generate the analogous images rather than retrieve them from an existing corpus. (Traina et al., 2004) split IR systems into conditional subsystems, but do not tackle generic conditioners or provide experimental evaluation. (Plummer et al., 2018) create an IR system conditioned on text input, but do not address the problem of generically filtering results. (Gao et al., 2020) and (Liao et al., 2018) respectively learn and use a hierarchy of concepts concurrently with IR features, which could be a compelling way to *learn* useful conditions for a Conditional IR system.

13. Conclusion

We have shown that Conditional Image Retrieval yields new ways to find visually and semantically similar images across corpora. We presented a novel approach for discovering hidden connections in large corpora of art and have creates an interactive web application, MosAIc to allow the public to explore the technique. We have shown that CIR performs non-parametric style transfer on the FEI faces and two newly introduced datasets. We proved a bound on the number of nodes that can be pruned from RandomProjection trees when focusing on subsets of the training data and used this insight to develop a general strategy for generalizing tree-based KNN methods to the conditional setting. We demonstrated that this approach speeds conditional queries and outperforms baselines. Lastly, we showed that CKNN data-structures can find and quantify subtle discrepancies between high dimensional distributions and used this approach to identify several "blind spots" in the ProGAN network trained on CelebA HQ.

Acknowledgments

We would like to thank the Microsoft Garage program for supporting the development of the MosAIc application especially Chris Templeman, Linda Thackery, and Jean-Yves Ntamwemezi. Additionally we would like to thank Anand Raman, Markus Weimar, and Sudarshan Raghunathan for their feedback on the work and for their support of the work.

References

The Metropolitan Museum of Art Open Access CSV, 2019. URL https://github.com/metmuseum/openaccess.

The Rijksmuseum Open Access API, 2019. URL https://data.rijksmuseum.nl/.

Martin Aumüller, Erik Bernhardsson, and Alexander John Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *CoRR*, abs/1807.05614, 2018. URL http://arxiv.org/abs/1807.05614.

Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 257–264, 2014.

Dmitry Baranchuk, Artem Babenko, and Yury Malkov. Revisiting the inverted indices for billion-scale approximate nearest neighbors. *CoRR*, abs/1802.02422, 2018. URL http://arxiv.org/abs/1802.02422.

David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate, 2019.

Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

Bing. Beyond text queries: Searching with bing visual search, Jun 2017. URL https://blogs.bing.com/search-quality-insights/2017-06/beyond-text-queries-searching-with-bing-visual-search.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL http://arxiv.org/abs/1706.05587.

Leonardo Dagum and Ramesh Menon. Openmp: an industry standard api for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55, 1998.

Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimensional manifolds. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 537–546, 2008.

Vinny DeGenova. Recommending visually similar products using content based features, Dec 2017. URL https://tech.wayfair.com/data-science/2017/12/recommending-visually-similar-products-using-content-based-features/.

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.

Aman Dhesi and Purushottam Kar. Random projection trees revisited. In *Advances in Neural Information Processing Systems*, pages 496–504, 2010.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

Artemij Fedosejev. *React. js essentials*. Packt Publishing Ltd, 2015.

Xinjian Gao, Tingting Mu, John Yannis Goulermas, Jeyarajan Thiyagalingam, and Meng Wang. An interpretable deep architecture for similarity learning built upon hierarchical concepts. *IEEE Transactions on Image Processing*, 29:3911–3926, 2020.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016.

Clinton Gormley and Zachary Tong. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

Edouard Grave, Armand Joulin, and Quentin Berthet. Unsupervised alignment of embeddings with wasserstein procrustes, 2018.

Mark Hamilton, Sudarshan Raghunathan, Akshaya Annavajhala, Danil Kirsanov, Eduardo Leon, Eli Barzilay, Ilya Matiach, Joe Davison, Maureen Busch, Miruna Oprescu, et al. Flexible and scalable deep learning with mmlspark. In *International Conference on Predictive Applications and APIs*, pages 11–22, 2018a.

Mark Hamilton, Sudarshan Raghunathan, Ilya Matiach, Andrew Schonhoffer, Anand Raman, Eli Barzilay, Karthik Rajendran, Dalitso Banda, Casey Jisoo Hong, Manon Knoertzer, et al. Mmlspark: Unifying machine learning ecosystems at massive scales. *arXiv preprint arXiv:1810.08744*, 2018b.

Mark Hamilton, Nick Gonsalves, Christina Lee, Anand Raman, Brendan Walsh, Siddhartha Prasad, Dalitso Banda, Lucy Zhang, Lei Zhang, and William T Freeman. Large-scale intelligent microservices. *arXiv preprint arXiv:2009.08044*, 2020.

William Christopher Hayes. *The scepter of Egypt: a background for the study of the Egyptian antiquities in the Metropolitan Museum of Art*, volume 1. Metropolitan Museum of Art, 1990.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

Joseph M Hellerstein and Michael Stonebraker. Predicate migration: Optimizing queries with expensive predicates. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 267–276, 1993.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song. Neural style transfer: A review. *CoRR*, abs/1705.04058, 2017. URL http://arxiv.org/abs/1705.04058.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, page 1–1, 2019. ISSN 2160-9306. doi: 10.1109/tvcg.2019.2921336. URL http://dx.doi.org/10.1109/tvcg.2019.2921336.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

Clare Le Corbeiller. *China Trade Porcelain: Patterns of Exchange: Additions to the Helena Woolworth McCann Collection in the Metropolitan Museum of Art*. Metropolitan Museum of Art, 1974.

Alon Y Levy, Inderpal Singh Mumick, and Yehoshua Sagiv. Query optimization by predicate movearound. In *VLDB*, pages 96–107, 1994.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

Lizi Liao, Xiangnan He, Bo Zhao, Chong-Wah Ngo, and Tat-Seng Chua. Interpretable multimodal retrieval for fashion products. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1571–1579, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Peng Lu, Gao Huang, Yanwei Fu, Guodong Guo, and Hangyu Lin. Learning large euclidean margin for sketch-based image retrieval. *CoRR*, abs/1812.04275, 2018. URL http://arxiv.org/abs/1812.04275.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

Elena Marchiori. Class conditional nearest neighbor for large margin instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):364–370, 2009.

Yusuke Matsui, Ryota Hinami, and Shin'ichi Satoh. Reconfigurable inverted index. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1715–1723, 2018.

Michael McCandless, Erik Hatcher, Otis Gospodnetić, and O Gospodnetić. *Lucene in action*, volume 2. Manning Greenwich, 2010.

Clayton Mellina. Introducing similarity search at flickr, Mar 2017. URL https://code.flickr.net/2017/03/07/introducing-similarity-search-at-flickr/.

K Nichol. Painter by numbers, wikiart, 2016.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Stephen M Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

Adela Oppenheim, Dorothea Arnold, Dieter Arnold, and Kei Yamamoto. *Ancient Egypt Transformed: The Middle Kingdom*. Metropolitan Museum of Art, 2015.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. URL http://arxiv.org/abs/1801.04381.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and vision computing*, 28(6): 902–913, 2010.

C. Traina, A. J. M. Trains, and J. M. de Figuciredo. Including conditional operators in content-based image retrieval in large sets of medical exams. In *Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems*, pages 85–90, 2004.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks, 2016.

Tijs Volker. *Porcelain and the Dutch East India Company: as recorded in the Dagh-Registers of Batavia Castle, those of Hirado and Deshima and other contemporary papers; 1602-1682*, volume 11. Brill Archive, 1954.

Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *CoRR*, abs/1408.2927, 2014. URL http://arxiv.org/abs/1408.2927.

Xueyi Wang. A fast exact k-nearest neighbors algorithm for high dimensional search using k-means clustering and triangle inequality. In *The 2011 International Joint Conference on Neural Networks*, pages 1293–1299. IEEE, 2011.

Edward TC Werner. Myths and legends of china. london: George g. *Harrap. Disertasi*, 1922.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. URL http://arxiv.org/abs/1611.05431.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

Donghui Yan, Yingjie Wang, Jin Wang, Honggang Wang, and Zhenpeng Li. K-nearest neighbors search by random projection forests. *IEEE Transactions on Big Data*, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

Appendix A. Visualizing Failure Cases

Figure 10 (a) shows how conditioners that do not share a common support can yield low diversity conditional neighbors. Though sharing a common support is certainly helpful, it is not mandatory as shown by Figure 10 (b). Some potential mitigations for these effects could be to fine tune learned embeddings to promote diverse queries, or to re-weight query outputs based on diversity. Additionally, an initial alignment with an optimal transport method could mitigate these effects (Grave et al., 2018).

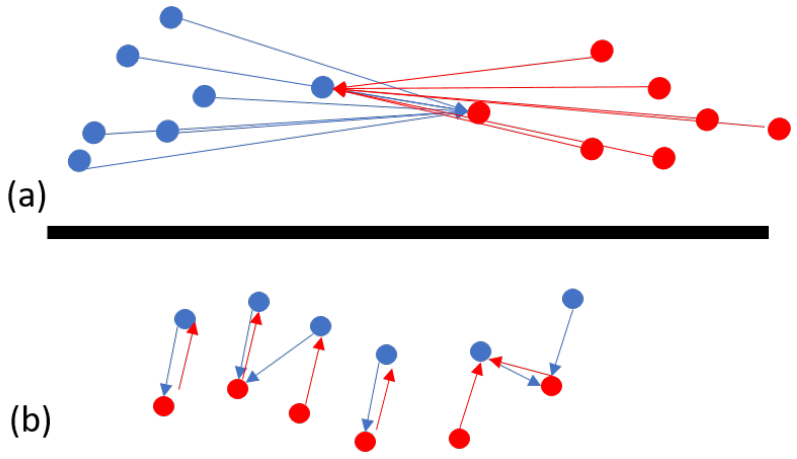

Figure 10: A schematic illustration of how conditional KNN can yield to a lack of diversity in particular geometries. (a) shows how low diversity can occur when there is no overlap of supports. Figure (b) shows how support intersection is not necessary for quality alignment

Appendix B. Additional Matches

In addition to the matches displayed in Figure 1 we provide several additional results. Figure 11 shows additional matches for a single query, and Figure 12 shows matches across several different queries. Figure 13 shows random matches to give a sense of the method's average-case results.

Figure 11: Additional conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using media (top row text) and culture (bottom row text) as conditioners.

Figure 12: Additional conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using top row text as conditioners.

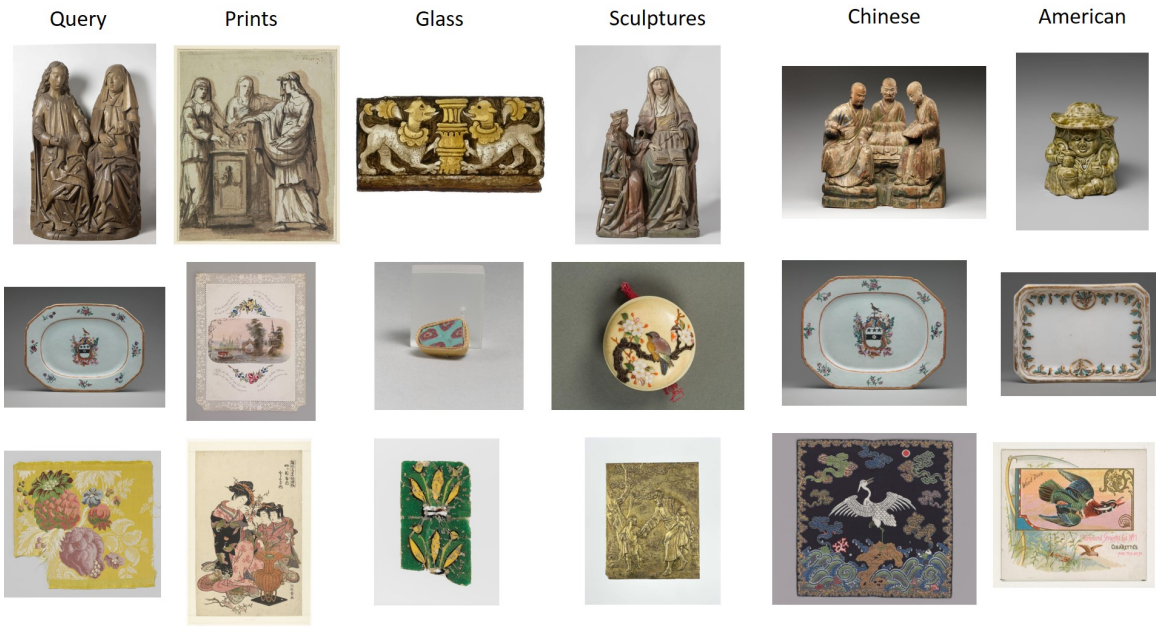

Figure 13: Randomly selected conditional image retrieval results on artworks from the Metropolitan Museum of Art and Rijksmuseum using top row text as conditioners.

Appendix C. Proof of Theorem 1

In the following analysis, suppose an RPTREE-MAX is built using a dataset $\mathcal{X} \subset \mathbb{R}^D$, of diameter W, with doubling dimension $\leq d$. Furthermore, assume that the size reduction rate at any given level of the tree is bounded above by γ.

Lemma 2 *For any ball, B of radius $R > 0$ and any $0 < \epsilon < 1$, there exists a constant $c_1 > 0$ such that with probability $> 1 - \epsilon$, B will be completely inscribed inside of an* RPTREE-MAX *cell of radius no more than $c_1 R d\sqrt{d}\log(d)$*

Proof We modify the proof of Theorem 12 from (Dhesi and Kar, 2010). In particular we let $\Delta^* = \frac{1}{\epsilon} c_5 R d\sqrt{d}\log(d)$, where c_5 refers to the constant of Lemma 11 of (Dhesi and Kar, 2010) The rest of the proof proceeds without modification. ■

Lemma 3 *For any finite set of balls, $\{B_i\}$, with constant radii $R > 0$, and any $0 < \epsilon < 1$, there exists a constant $c_2 > 0$ such that with probability $> 1 - \epsilon$, every B_i will be completely inscribed inside of an* RPTREE-MAX *cell of radius no more than $c_2 R d\sqrt{d}\log(d)$*

Proof We proceed by induction on the number of balls. Lemma 2 provides the base case of $|\{B_i\}| = 1$. For the inductive case we assume the lemma holds for a set $\{B_i\}$ of size n, with $\epsilon' = \frac{\epsilon}{8}$ and constant c_2'. Given an additional B_{n+1}, we can leverage our base case to select an $\epsilon'' = \frac{\epsilon}{8}$ and constant c_2''. We can see that the probability that both events occur simultaneously is bounded above by:

$$(1-\epsilon')(1-\epsilon'') = (1-\frac{\epsilon}{8})(1-\frac{\epsilon}{8}) = 1 - \frac{\epsilon}{4} - \frac{\epsilon^2}{64} < 1 - \epsilon$$

Finally, using the new constant, $c_2 = \max(c_2', c_2'')$, the radii criterion holds for all balls. ■

Theorem 4 *(Restatement of Theorem 1) Suppose an* RPTREE-MAX*, $\mathcal{T}$, is built using a dataset $\mathcal{X} \subset \mathbb{R}^D$, of diameter W, with doubling dimension $\leq d$. Further suppose $\mathcal{T}$ is balanced with a cell-size reduction rate bounded above by γ. Let $\mathcal{S} \subseteq \mathcal{X}$ be a subset of the dataset used to build the tree and $\mathcal{B}$ a finite set of radius $R > 0$ balls that cover $\mathcal{S}$. For every $0 < \epsilon < 1$ there exists a constant, $c > 0$, such that with probability $> 1 - \epsilon$ the fraction of cells that contain points within $\mathcal{S}$ is bounded above by $|\mathcal{B}|2^{-log_\gamma(W/R')}$ where $R' = cRd\sqrt{d}\log(d)$*

Proof We begin by invoking Lemma 3, which shows that each ball of our covering will end up completely inscribed within small radii cells of $\mathcal{T}$. For each ball we upper bound their contribution to the total fraction of cells that contain points within $\mathcal{S}$.

Consider any ball $B_i \in \mathcal{B}$ in the covering. By Lemma 3 we know this ball is inscribed within a cell of radius R'. Our goal is to show that this cell must be several levels down in the tree. By our regularity conditions we know that at each subsequent level of a tree, the cell size decreases by at most a factor of γ. So to achieve the reduction in size from W to R', the cell must lie at or below level $\log_\gamma(W/R')$. At worst, every child of our cell contains a point within $\mathcal{S}$. Because $\mathcal{T}$ is balanced, the ratio of cell children to total cells of the tree is at most $2^{-log_\gamma(W/R')}$. At worst each ball of the cover, $\mathcal{B}$, is in a separate branch of the tree so combining these contributions yields $|\mathcal{B}|2^{-log_\gamma(W/R')}$.

■