PRECISE IMAGE EXPLORATION WITH CLUSTER ANALYSIS

An Undergraduate Research Scholars Thesis

by

SAGAR PATEL

Submitted to the Undergraduate Research Scholars Program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. James Caverlee

May 2020

Major: Computer Science

TABLE OF CONTENTS

Page

ABSTRACT 1

ACKNOWLEDGMENTS 2

CHAPTER

I. INTRODUCTION 3

II. RELATED WORK 8

- Image Exploration 8
- Defining and Representing Subject 9
- Defining and Representing Style 10
- Diversifying Search Results 10
- Personalized Image Recommendation 11

III. METHODS 12

- Problem Statement and Overview 12
- Exploration 12
- Deep Metric Learning 15

IV. EXPERIMENTS 20

- Dataset 20
- Experimental Setup 21
- Results 23

V. CONCLUSION 27

REFERENCES 28

ABSTRACT

Precise Image Exploration with Cluster Analysis

Sagar Patel
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. James Caverlee
Department of Computer Science and Engineering
Texas A&M University

Since the rise of digital multimedia in our present age, when looking for an image that closely matches their needs and preferences, the number of images a user must sort through has become more and more unmanageable. Even when searching for a narrow topic, it can be nearly impossible to find an image that meets a specific preference by going through all the possible images.

To combat this growing problem, we describe an exploration system built on deep neural networks to empower the users to quickly sort through all the possible images by quickly narrowing down to their preferred images. By design, our exploration system goes around the need to match the user's query directly to a small group of images to serve users images that would traditionally be too difficult to group together and match to a query. We propose to use deep metric learning and clustering to group the images, which we will see cleverly manages problems that hold back traditional neural networks in this problem—unseen image groups and shifting definitions.

ACKNOWLEDGMENTS

I would like to thank my research advisor, Dr. James Caverlee and Infolab for guidance and support throughout my research project, and for providing a platform to test my research. From Infolab, I would also like to specifically thank Yin Zhang for taking time out of her own work to help me conduct my research smoothly.

CHAPTER I

INTRODUCTION

As the web has grown more and more connected to our lives, social media has become a vital means for us to connect to the world. As a result, image-focused social media platforms like Flickr and Pinterest have exploded in size, and with them so have the problems related to managing and serving a large catalog. For reference, in December 2019, in a blog about their company, Flickr reported having more than 100 million accounts and tens of billions of photos, and Pinterest reported having more than 320 million users and 200 billion posts (referred to as "ideas" on the platform) [1, 2]. At this size, even with a narrow topic in mind, we cannot explore all the content that these large-scale platforms offer.

To deal with such large collections, social media platforms, not unlike other web platforms, have continuously worked on improving search. Search in this context has been the focus of decades of research, and has grown to incorporate more and more features to present each user a ranked list of images that fit their need [3]. Most of this research has worked to improve on metrics focused on a natural measure of quality of the search results—relevance. Relevance in this context is a judgement on whether a result answers the information need behind a query or not. For example, in a query for "New York", an image of one of its boroughs, Manhattan, would be relevant while an image of Seattle would not be.

However, even with the advances in search, complex and specific queries are still particularly difficult to handle because they require an extraordinary level of understanding of both the images and the query. As a result, when we search for a specific query, we often get back only a few results, requiring us to broaden our search and going through the images ourselves. For example, when we search directly for "Ueno Park long exposure" on Flickr as shown in **Figure 1** and **Figure 2**, we get only 38 search results that fit our need. However, when we search for a generalized version of our query, "Ueno Park", we get about 85,000 results, of which, many fulfill

our need that we didn't see when we search directly for them. From the users' perspective, this is a big problem because to them, the fact that the first search didn't fulfill their need means that they must personally go through the large collection of images themselves to find images that fit their need, mixed in with a lot images that aren't relevant to their need. Finding those specific types of images in an ever growing list increasingly feels like finding a needle in a hay stack.

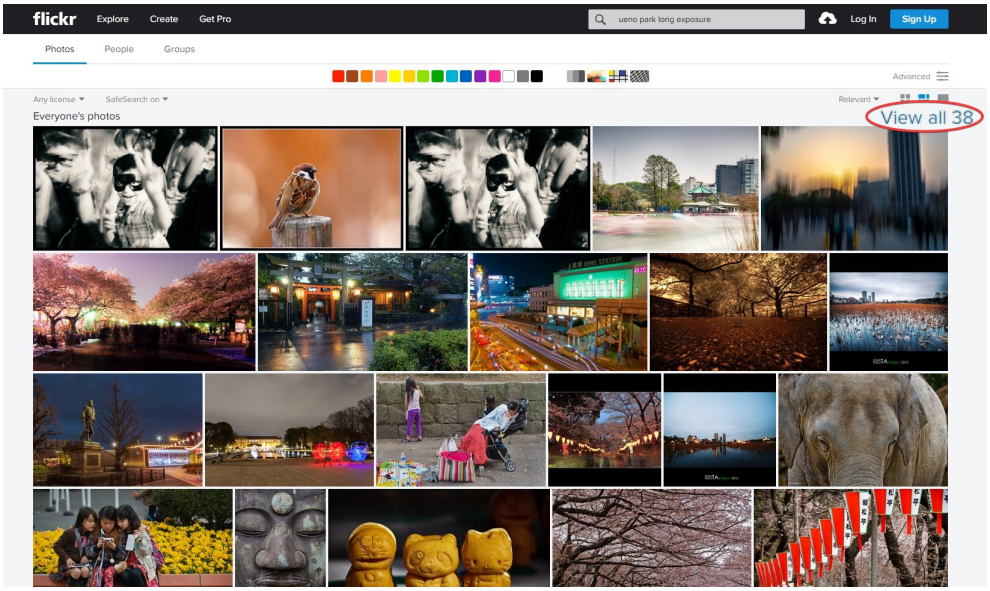

Figure 1: Flickr search results for the term "ueno park long exposure"

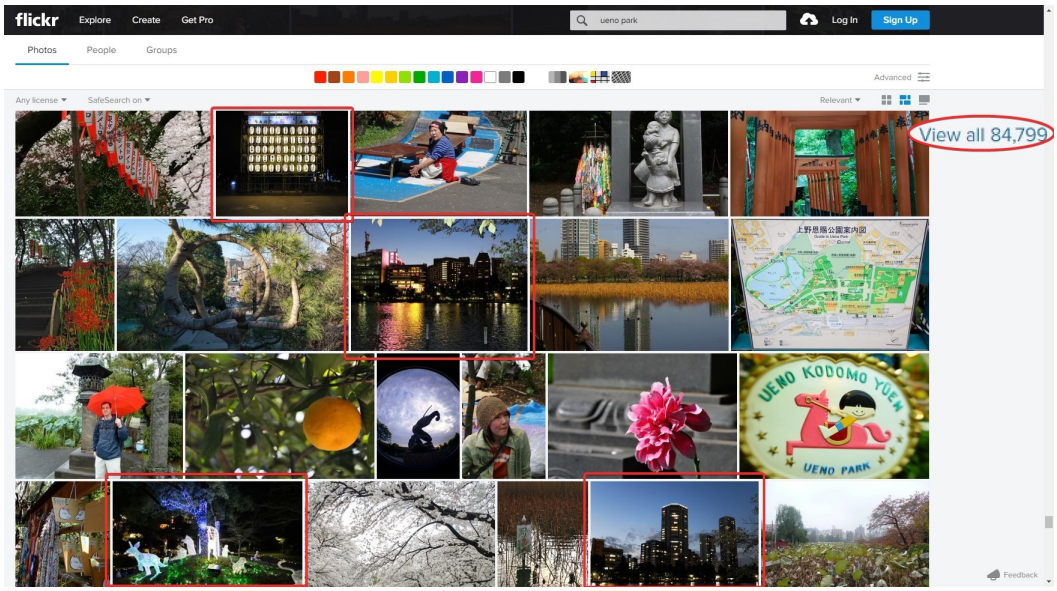

Figure 2: Flickr search results for the term “ueno park”

So, to better connect users to the specific type of image they want, we propose to build upon image exploration [4]. Image exploration works to connect users to specific images by allowing users to explore all the facets of their query and narrow down their search. We propose to expand this system by presenting narrow facets over both aesthetic and subjective categories. The system would first present the users with facets of images formed from an initial category. Then, selecting a facet would narrow down all the images to only those belonging to the facet, and run the images through cluster analysis of a different category, or simply display them if the categories convey enough information or if the images have already been narrowed down enough.

Since our precise image explorer does not rely on properly identifying images such that they can be matched directly to users’ queries, we fundamentally do not have to rely on breaking down the intricacies of the query to serve users images. As such, we can fulfill narrow queries without addressing all the problems traditional search platforms have to do the same.

In this view of the problem, there are four main challenges:

- *Defining and identifying what separates two images.* Even when we're given 2 groups of images, it's hard to generalize what separates them. This problem becomes much more difficult over hundreds of classes, which can easily happen in our problem.

- *Handling unseen images.* In practice, we must deal with types of images that our model has never seen (it matches none of the groups we trained over).

- *Managing shifting definitions and preferences.* While in an offline lab setting, our dataset doesn't change. However, when our model is used in a real application, all the information around it is constantly changing. What defines a particular aesthetic style changes in the world of photography, and our model has to keep up

- *Staying computationally simple.* The methods we use should be quick enough to satisfy the users interacting with it.

So, to help bridge the gap between the gap between what's easily available to the users and what could potentially be, we must find ways to connect users to the images of the subject they want in the style they want. Overall, the contributions of our work are the following:

- First, *we describe a new approach to this problem: precise exploration*. We propose to use exploration infused with deep metric learning to connect users to specific images in a fundamentally different way than search.

- Second, *we explore using hierarchical loss* to deal with groups with few images in deep metric learning use cases similar to ours. Analyzing the subject of the images in deep metric learning raises a unique problem, as we will talk about later, and we find the effects of using hierarchical loss to address it.

- Third, *we introduce a new dataset, places190, simulating a real world application of precise exploration*. This dataset allows us to design and conduct experiments that can specifically aim to improve deep metric learning in exploration.

For the remainder of the thesis, we will contextualize our research, properly define our work's assumptions, detail our methodology, provide our results and conclusions, and provide potential ways to expand our work.

CHAPTER II

RELATED WORK

The central research problem of this work, connecting users to specific types of images, is a very well researched problem and there are many facets of this problem that have already been studied. In particular, there has already been considerable work using image exploration, identifying and defining subject and styles, diversifying search, and personalized image recommendations in this setting. At the core, we apply a different method than these works, and so alongside the works mentioned here, have more relevant research that we will discuss later when we discuss our methodology.

Image Exploration

Image exploration is an application of faceted search in the context of images [5]. Faceted search groups images based on an inherent hierarchical manner, much like how it works in shopping catalogs, where products are listed in a hierarchical manner, going from large categories like "Health & Beauty" to "Hair" and then to "Shampoo". The research in this topic largely dates back to the time before neural networks caught up in the area of Computer Vision. Exploration was seen as a way to help alleviate the problem associated with content based image retrieval and a way to evaluate and build upon search results [4, 6]. Without a proper way to identify the content of an image, exploration worked to group images in a hierarchical way and present facets of information to the user. This was primarily done on a fairly broad scale, typically by classifying images into human-identified categories based on simple knowledge graphs. For example, [4] finds all the attractions in the given place and using already known information like the location of the taken image and the attraction to match images to categories. However, as advances in machine learning and neural networks were made and problems content-based image retrieval that it targeted were solved, exploration has become less emphasized.

In this thesis, we build upon the basic idea of image exploration introduced in these works,

and apply deep metric learning to take exploration from a broad human-identified categorization to very fine-grained automatic image categorization that acts as a medium to connect users to specific images they want. At the core, we similarly work to alleviate the problems associated with content-based image retrieval, but on a much finer scale.

Defining and Representing Subject

As we understand it, subject is what a picture is taken of. A subject, according to our definition here, is the focus of the picture and is implicitly different from objects that simply exist in the image. However, despite the technical distinction in definition, in practice, it's better to think of subjects as just central objects or what multiple objects make up. As such, all the work done on object detection and specialized versions of it is relevant to ours. This topic is one of the biggest in field of Computer Vision, and there is an enormous amount of work dealing with this problem. So, we will only mention only a few key works directly related and used in this thesis. ImageNet, comprising of ILSVRC challenges and the data, was a major spark in research around object detection, providing hand-labeled 10,000,000+ images with 10,000+ object classes. With significant contribution to making deep neural networks computationally possible with GPUs, research in [7] proposed AlexNet, a deep CNN architecture, to classify the over 1000 classes in ILSVRC 2012 challenge. Following the computational advancement AlexNet made, [8] introduced GoogLeNet, a CNN architecture that goes even deeper and wider than AlexNet, while still being computationally costing similar. A year later, [9] introduced ResNet, a deep residual neural network architecture that, unlike previous work, used residual connections with explicit reference to layer inputs so that networks with large depths that they couldn't even be trained before could be trained in even lower computational cost than the then state of the art neural networks. Since then, ResNets have become the default choice of neural architecture in image recognition tasks for many problems, and we will also use ResNet in this work as a baseline, and build on it to allow us to better identify subjects, the central objects of our pictures.

Defining and Representing Style

Despite lacking an established definition, we understand style as it refers to the aesthetic perception of an image, a quality that measures how an image was taken, rather than what it was taken of. Concretely, in the photography world, style can be identified by classifying the way the camera was used to take the picture, down to labels such "long exposure" or "shallow depth of field". Since these qualities are rather subjective, capturing these qualities in manner that a computer can understand is in and of itself very difficult. To try to do so, [10] introduces the AVA dataset, which along with other aesthetic classifications, names 14 styles (as taken from a popular photography magazine) with about 1200 images each, derived from identifying key words mentioned description of the images. [11] instead defines 20 concrete styles by the 4000 images per style they gained by mining Flickr groups associated to that style, and then trains AlexNet (the CNN architecture trained on ImageNet we mentioned earlier) to classify those styles. In this work, we will use this dataset and build our models around it.

Diversifying Search Results

The main idea behind diversifying search results is that diversity is desired in search results to queries that are ambiguous. Diversity in this sense refers to a measure that identifies how dissimilar relevant results of the query are to each other.

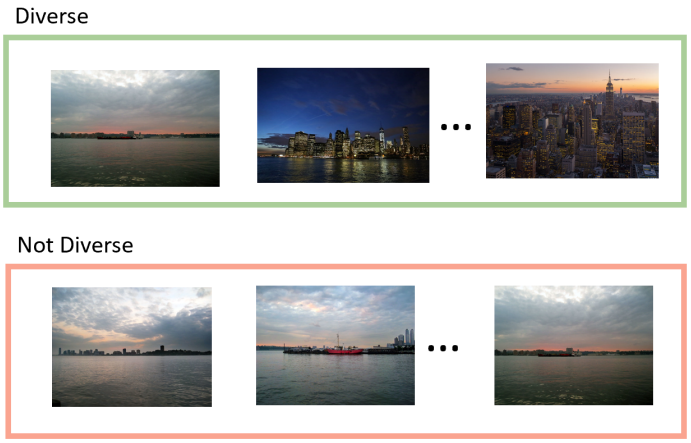

Figure 3: Example of diversity in search results

For example, in **Figure 3**, we can say that results on top are more diverse than the ones on the bottom because while they both have the correct images, the ones on the top are subjectively more dissimilar to each other. Diversifying search results is a major topic in Information Retrieval and as such, has an immense amounts of research and tasks dedicated to it [12, 13, 14, 15, 16, 17, 18]. In image retrieval specifically, [19, 20] introduced the MediaEval2014 and Div400 datasets with hand-labeled data that provides an ideal ranking with diversity in mind, and a way to measure deviance from those ideal ranking. However, we will not review the specific work produced by these tasks because despite the fact that the idea behind diversity is still very valuable, the specific methods these works use are no longer very relevant in the modern image retrieval space with deep neural networks. This is because as [21] reviewed in a presentation at multiple major IR conferences, neural networks have become increasing popular in Learning to Rank applications because they have been so successful at modeling an immense amounts of features and information that previous work that rely on them just can't reliably outperform them.

Personalized Image Recommendation

Recommendations can be thought of as the system's results to an implicit query of user just passively existing. So far, we have only focused on research as it relates to search on these social media platforms. However, it's important to also look at this problem from the recommendation perspective, as it's also a major part of the user experience. Since advancements in object recognition and the rise in popularity of neural networks, research in recommendation system in this space has grown to model more and more detailed features with increasing accuracy. [22] introduced the idea of content-based personalization in image recommendation in social media by analyzing click-through data, using image attributes and descriptions to understand the image. Using that idea in the neural network space, [23] incorporates spatial, temporal, and visual features into recommendations by combining matrix factorization and Bayesian Personalized Ranking. [24] incorporates social relationships and association with groups into recommendations.

While we won't directly build upon the work done here, we will introduce methods that can be adapted to improve this area as well.

CHAPTER III

METHODS

Our goal is to ultimately connect users to the specific images they want more easily.

Problem Statement and Overview

In a traditional search environment, to connect users to the specific images they want, we must take their query q and break it down into f facets, filter by any qualifiers in q, and then produce a list of images ranked by some score that measures their quality and match to q. For example, in the query "Empire State Building at night", the only subject or facet is "Empire State Building" with the qualifier "night". So, the answer to the query should be a ranked list of quality images that show the Empire State Building at night.

However, we of course can't directly follow our ideology and have a computer analyze the query exactly how we would. We must approximate that understanding. As such, the quality of our best approximation of the understanding of the query is extremely important to providing quality search results. This is because even if we perfectly analyze all of our images and their dynamics, we can't provide users with what they want if we cannot match their language to our analysis. While we won't mention all the problems that makes it so, in long and complicated queries, the intricacies in language make it difficult to approximate understanding all its facets and qualifiers.

To deal with this difficulty, instead of proposing an improvement to the way we analyze the query, we adopt an exploration based approach:

Exploration

Concretely, we propose to expand upon the exploration system proposed in [4] that presents the users with clusters of images formed from the selected category (like subject) as a way to navigate through the images by using multiple layers of categorization. Selecting a cluster would narrow the images down to only those, and run the images through cluster analysis of a different

category, or simply display them if the categories convey enough information or if the images have already been narrowed down enough.

This sort of exploration can be understood as search without classification. If we identify all the clusters we see along the way, we can directly connect users to that specific cluster by matching it to their query. However, by using exploration, we are separating our ability to serve images and our ability to match all the clusters to users' need. After doing so, we can present users with the specific images they want without having to fully understand the intricacies of the query. A user can simply query a simpler version of their query in the explorer, be presented with different facets that they'd be interested in, and quickly arrive at the cluster they were looking for.

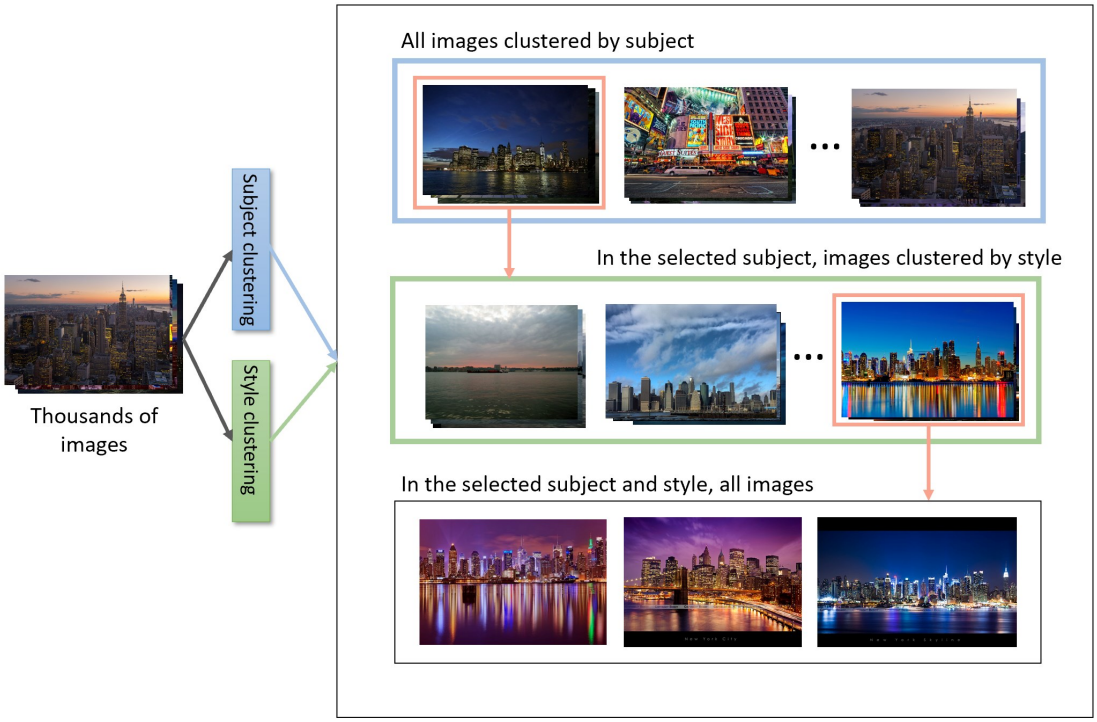

Figure 4: An explorer with subject and style analysis

In **Figure 4**, we can see an example of an implementation of this exploration approach with subject and style as the categories to cluster around. We chose subject and style to cluster around because that's what we felt was enough to narrow down the images we encountered (leaving it up to the internal search to provide all the images at least close to a "general" query). However, if in another application we are particularly interested in an additional category (ultra fine-grained subject, for example), we can simply cluster around the new category and display images by *subject* $\rightarrow$ *ultra fine-grained subject* $\rightarrow$ *style* instead.

Formally, we define this sort of exploration as the way to serve the most diverse set of images in a given category over all the categories. To naturally accomplish this, we cluster the images around each category, and return the predicted cluster labels (one for each category) for each image to the exploration viewer to serve. Given the categories C and the set of query results I, for $i \in \mathbb{R}^I$, we must predict the cluster label $l \in \mathbb{R}^L$ for each category $c \in \mathbb{R}^C$ (where L, the total number of clusters in category c, varies for each c and each query).

When given this problem and the two categories subject and style, the obvious answer would be to just use a fully supervised neural network to classify every class. To do so, we can view changing L as a variable bounded by L', the theoretical maximum number classes that all the potential classes of all possible queries would produce for a category c, and just substitute the constant L' in for L to get a $\mathbb{R}^{L'}$ classification problem for each image, for each category $c \in C$ (note that L' is still different for each category). Style identification, as defined by [11], can already be done by a supervised CNN, with L' set to 20 (the number of style classes in their dataset). Subject identification can also be formulated as fully supervised classification by defining L' as the sum of all distinct clusters of all the representative queries on the platform.

However, while this formulation could perform well in an offline lab-based setting, in a real-world platform with hundreds of millions of images, it's crippled by two of the main hurdles we mentioned in the introduction: unseen images and shifting definitions. These two hurdles mandate that we know about every cluster and that we keep our knowledge absolutely up-to-date. Because if we don't know about every cluster or if we only know about the category from 10 days

ago, we can't cluster properly. For example, in subject identification, if we trained over Chicago and New York, running images of Seattle through our model would tell us that none of the images belong to any class with reasonable confidence if Seattle was, in fact, different from Chicago and New York. And even if we did know about the cluster, if our training dataset didn't closely match today's understanding of styles and subjects, we would get our cluster labels incorrect (only this time confidently so).

To deal with these hurdles, we would have to constantly train and re-train our model, learning about unseen images and re-learning definitions of old clusters. Both of which, are very costly. So, to reduce our need to constantly train and re-train, we propose to use deep metric learning for this problem instead of traditional neural network methods.

Deep Metric Learning

Deep metric learning (also often referred to as distance metric learning) is a semi-supervised learning method to learn information-rich representations or features in a low-dimension space using neural networks by first training over a very large general task, and then re-training only the last layer (the layer which provides the representations) on the specific task [25]. It's often used in fine-grained image recognition tasks like facial recognition, reverse image search, and fine-grained image search [26, 27, 28]. In image classification tasks, the neural network is naturally first trained on ImageNet, and then retrained on a small dataset with a loss function that attempts adjusts the wights so that we maximize distance between two dissimilar images and minimize the distance between two images similar images.

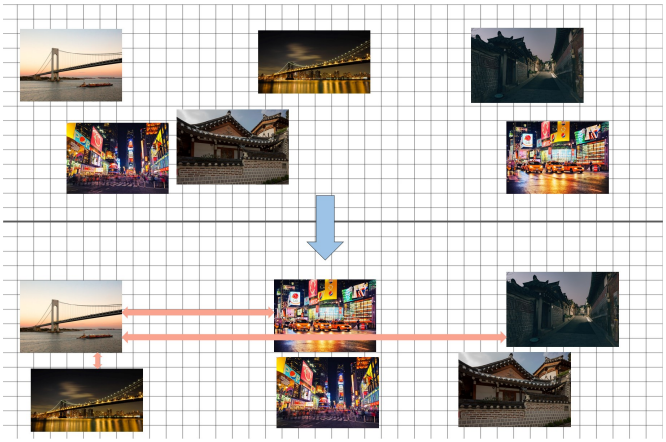

Figure 5: Demonstration of deep metric learning's objective

In the demonstration given in **Figure 5**, we can see what we're trying to accomplish with deep metric learning in our toy dataset for subject diversification. We start in a space that doesn't separate our data very well, and want to end up in a space that separates our data as best as possible. Mathematically, we can find this transformation by calculating the loss in our current representations (as represented by a metric that measures the euclidean distance between images that should be similar and images that shouldn't be) and back propagating to end up in our desired space over many iterations.

More concretely, in training, given the class labels for all the training images, we arbitrarily separate them into mini-batches (Note that batch selection also matters, and can be thought of as part of sampling), calculate their representations by feeding them forward our neural network, and then sample n images, calculate the loss based on the samples, and back propagate the loss. We repeat this process for all the remaining mini-batches and ultimately repeat again for all the epochs we need to converge. Technically, our the loss functions are defined as the loss over all combinations of the images in the mini-batch (all n^3 triplets if using triplet-based loss for example). However, in practice, we sample to estimate this loss because it's computationally hard to go through all the combinations and many of the combinations become 0 after just a few iterations.

The main benefit of deep metric learning is its ability to produce feature-rich representations

of unseen images. In fact, in typical deep metric learning tasks, the test cases consist only of images belonging to classes never seen before [29, 30]. This is because by learning the transformation that separates our training images into their respective class as much as possible, we implicitly learn the intricacies between all the classes and use that knowledge to evaluate where an unseen image fits in the transformed space.

Most of the research into deep metric learning has focused on improving the two main components that impact the learning process: sampling and loss. [31] introduced a fundamental way of thinking about loss functions in deep metric learning by introducing triplet loss, the loss function. Triplet loss, given an image, finds a positive image (an image that's in the same class as the original) and a negative image (an image not in the same class as the original), and calculates loss such that the distance between the given and positive image is small, and the distance between the negative is large. Building on this basis, [32] introduces Proxy-NCA, which incorporates the idea of neighborhood component analysis into triplet loss, and instead of forming triplets with single positive and negative images, forms triplets with proxies that represent an entire neighborhood of images. Unlike the other papers, [33] looks at the effect of sampling in deep metric learning, and finds sampling is just as important as the loss function, and that properly weighted sampling with a simple loss function can outperform many state of the art methods. In our work, we use their distance weighted sampling function, and explore working with a hierarchical loss function to work alongside it.

Deep Metric Learning in Exploration

Exploration, as we mentioned earlier, needs clusters that represent the most diverse sets of images. To get those with deep metric learning, we need to train two models per category: a neural network doing deep metric learning and a clustering model. We naturally train the neural network on classes for the given category, and use a simple k-means clustering method to cluster the resulting embeddings for the images. This separation of our feature learning and clustering is a major benefit to us, and is the reason why deep metric learning handles shifting preferences much better than a normal neural network classifier. With feature-rich embeddings, it's much

easier for us to monitor the shift in our data (which can be the emergence of a new class or the shifting definition of an existing one), and work to incorporate these changes in our final clustering model—all without re-training our deep metric learning model.

That said, it's important to note how subject clustering in the deep metric learning approach has a unique problem. When we train our neural network, we train over all the different different classes of all the queries (exactly how deep metric learning models are traditionally trained). This means that 5 classes formed from the query "New York", which are intuitively more similar to each other, are treated the same as 5 classes from 5 different queries in our training process. Of course, over enough samples, this predisposed intuition would be naturally captured by our neural network. However, classes from a query that doesn't return many images would particularly suffer because they would be overpowered by the noise introduced by treating intra-query classes and inter-query classes the same, and fail to converge at an equilibrium where all its classes are appropriately separated.

To help deal with this problem, we explore using hierarchical loss alongside proper distance-weighted sampling introduced by [33].

Hierarchical Loss

When using hierarchical loss, our main objective is to exploit predisposed information we have about the relationships between classes to guide our deep metric learning model to reflect these relationships. The assumption of that objective being that if only few images are preset for a set of classes, our model can compromise and leave the classes to be similar to each other, instead of trying to forcefully separate them and ending up with very noisy embeddings.

To implement this idea, we make a very simple modification to the margin loss introduced by [33] by adding a dynamic variable δ.

$$\ell^{hierarchical}(i,j) := (\alpha + y_{ij}(D_{ij} - (\beta \times \delta_{ij})))_+ \qquad \text{(Eq. 1)}$$

In **Eq. 1**, the i are the image we picked, and j is either a positive or a negative sample for the image we're trying to find loss for, α is a hyper parameter that controls margin of separation,

$y_{ij} \in \{-1, 1\}$, D_{ij} is the distance between the two images, and β is the class-specific learned parameter that controls the separation of boundary between positive and negative classes. The variable we added, δ_{ij} adds predisposed information about the class relationships to the learned boundary from the data when comparing to negative samples. In positive samples, i.e the samples of the class the image belongs to, δ_{ij} is 1 because our predisposed information is available only between two different classes. δ_{ij} is a class-specific variable that incorporates intuitive distance between samples of two different classes, as defined by a tree that represents the training dataset. This tree can either be generated automatically according to some metric, or be hand-made based on prior information about the dataset (ex. all SUVs made by different manufacturers belong to the higher order topic "SUV", and can be grouped as such).

In this thesis, we will treat δ_{ij} as a class-specific variable, and work under a simple intuitive assumption that images belonging to the same query are naturally more similar to each other than the images from a different query. We primarily do this because while it's possible to more precisely find the intuitive distance between all possible cluster pairs, finding it would require us to already have a model that does something similar to deep metric learning.

As we see from **Eq. 1**, in negative samples, when $y_{ij} = -1$, the loss induced by the sample becomes 0 if $D_{ij} + \alpha = \beta \times \delta_{ij}$. So, if we intuitively believe that our anchor image, the one we picked samples for, should be closer to this specific negative class than other classes, we could set δ_{ij} to be lower than all the other classes so that this separation boundary induces 0 loss even while being smaller than the other classes.

CHAPTER IV

EXPERIMENTS

In this work, our main conceptual contribution is separating labeling and differentiation of image classes to serve users the specific images they want without a need to explicitly understand their need or the meaning behind the image classes. Our technical contribution, hierarchical loss that adds predisposed information about class relationships, we will evaluate on the places190 dataset that we introduce that separates images based on the subject category.

Dataset

We introduce the places190 dataset, which comprises of 30 queries or large topics (ex. New York), each broken down into 5-7 classes or small topics (ex. Manhattan Bridge). We picked the large topics by picking popular locations on Flickr, and then picked small topics to break the larger topic down into by going on Tripadvisor and finding the top attractions for the given location. When picking large topics, we first ensured that its general subject is not the same as any large topics we had already picked (this is mainly to avoid adding significant noise when we do negative sampling), and that it can be broken down into at least 5 small topics. When picking the small topics, we just ensured that there were at least 50 public images available on Flickr for the given attraction on Tripadvisor. After generating the dataset's topics, we directly use the Flickr API to mine 50 public images per small topic, for a total of 30 large topics, 190 smaller topics and $190 \times 50 = 9500$ images.

This mining isn't a perfect way to get images for the smaller topics because it introduces quite a bit of noise, relying on crowd sourced annotations for the topics and providing no guarantee of a subject other than the annotated one not being present. However, we use it because it very accurately resembles a normal use of exploration on Flickr. We further also limit our images to 50 per small class to force our model to be robust enough to perform well on worse than usual conditions.

Due to these constraints we have imposed on our dataset, it's fundamentally hard to perform well on. As such, we expect that a model that performs well on this dataset will perform even better in real-world applications of exploration.

Experimental Setup

To implement our hierarchical loss, we extend the work of a popular github on deep metric learning[1] that implements multiple baselines. We experiment with the state-specific version of our loss, and use the 2-depth natural hierarchy formed with larger topics and smaller topics as our states. All experiments for subject clustering were performed on a desktop with 8-core Intel i7-4820k 3.7Ghz, 65GB physical memory and Nvidia GeForce RTX 2080TI with 11GB graphic memory.

Sampling

Following the train-test split in other deep metric learning datasets, we split our dataset by classes, training over the first 95 small image classes, and testing over the remaining 95. To create mini-batches from these training images, we also follow [34]'s mini-batch construction and create mini-batches of size 112 with 4 positive samples per class.

We use [33]'s distance weighted sampling to estimate the loss inside each mini batch in our loss function. Due to the skew of the classes in the subject category we've already discussed, distance weighted sampling is vital to quality embeddings in our case.

Evaluation Metrics

To evaluate the quality of our deep metric learning model, we first pass forward all our testing images through the trained neural network, and then use k-means clustering with 95 classes. Since we can't directly measure precision from the resulting clusters of the k-means model, we follow other deep metric learning papers, [26, 27], and adopt normalized mutual information (NMI) and recall@k to evaluate the quality of our deep metric learning model.

[1] https://github.com/Confusezius/Deep-Metric-Learning-Baselines

NMI measures the "amount of information" stored in clusters, and be formally written as:

$$\text{NMI (X,Y)} = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}} \qquad \text{(Eq. 2)}$$

Here, $I(X;Y)$ is mutual information between the predicted cluster labels X ($X = \{0, .., n-1\}$) and ground truth cluster labels Y ($Y = \{0, .., n-1\}$), and $H(X)H(Y)$ is the entropy for the the two cluster labels respectively.

Recall@k measures the quality of nearest neighbor retrieval, adopted from [35]. It measures the quality of image retrieval by finding the proportion of query images (one of the images of the clusters) for which the k nearest neighbors represent the whole true cluster. It can still be represented in the original equation of recall as:

$$\text{Recall@k} = \frac{1}{N}\sum_{1}^{N}\frac{|true\_cluster[i] \cap pred(true\_cluster[i])@k|}{|true\_cluster[i]|} \qquad \text{(Eq. 3)}$$

Here, the predicted images for the given cluster i ($pred(true\_cluster[i])@k$) are the k-nearest-neighbors of the centroid (centroid $\in$ all predicted centroids $\{0, .., n-1\}$) most similar to a randomly picked image from the cluster.

Looking at exploration, NMI is a measure of how informative our clusters are in a given category, and recall is a measure of how many of the original images we are able to display on the last step. Here, a high NMI is essential to our deep metric learning method because without one, our exploration system lacks enough confidence to be usable. Of course recall also matters. However, with a low NMI score for example, when a user clicks on a subject cluster that resembles the Manhattan Bridge, we cannot assure the user that pictures of the Manhattan Bridge aren't hidden somewhere in a different subject cluster, or that the images inside are exclusively of the Manhattan Bridge. And without that reasonable assurance, we end up right at our central problem and force the user to explore all the images themselves.

Results

Since we do not fully understand the impact of changing our new variable δ, we begin by quantifying its impact on recall for our dataset. When changing δ, it's important that we never set it less than 1.0 because we directly multiply it with our learned parameter β, and β has regularization applied to it. So, if we multiply β by a number less than 1.0, we indirectly induce regularization loss as well. To avoid that, if we need to increase the gap, we simply increase the other delta instead.

Table 1: Impact of changing delta on recall@k

Delta		Recall			
close	far	1	2	4	8
3.0	1.0	0.6432	0.7263	0.8034	0.8556
1.5	**1.0**	**0.6562**	**0.7404**	**0.8074**	**0.8615**
1.0	1.5	0.6500	0.7375	0.8011	0.8526
1.0	3.0	0.6360	0.7234	0.7922	0.8446

In **Table 1**, "close" means that the two classes are siblings in the dataset tree, and "far" means that they are not. As we see from the table though, we got results that contradicted our initial conjecture that classes that are close to each other intuitively should be forced to be closer while training. Our conjecture rested on the assumption that if they weren't, the noise added by forcing them apart with a small dataset would overpower the benefit we get from potentially successfully separating our classes apart for clustering. However, even with 50 just images, it appears that our conjecture does not hold up. This is likely because our dataset, although very sparse, still has enough images to successfully separate class boundaries (although not mentioned here, this effect is also evident by the fact even when mean loss per epoch drastically changes by changing deltas, all of them still converge at a similar rate) with deep metric learning, and helping our method do this is what actually gives us performance increase. Outside of our conjecture, the table also shows that values too far from the normal also deteriorates our method, no matter which way it is. For

comparison, all β values per class are initialized as 1, and end up in range $\{0.94..1.4\}$.

Now, we compare the results of our hierarchical loss against state of the art deep metric learning models, using the best values of delta that we got from our previous experiment.

Table 2: NMI and recall@k on PLACES190

Model	NMI	Recall			
		1	2	4	8
Proxy NCA [32]	0.5411	0.5739	0.6615	0.7392	0.8114
LiftedStruct [26]	0.5654	0.6314	0.7192	0.7964	0.8503
N-Pair [36]	0.5784	0.6244	0.7183	0.7918	0.8469
Margin [33]	0.5967	0.6493	0.7349	0.8027	0.8575
Hierarchical	**0.5975**	**0.6562**	**0.7404**	**0.8074**	**0.8615**

As we can see, hierarchical loss outperforms all of our baselines. However, if we critically look at our dataset and our performance gains, we can see that predisposed information we were working with doesn't actually contribute much. In fact, the performance gain over the simple Margin Loss is minimal. Now considering that our dataset was comprised of only 30 large topics and had clear connections between siblings classes, we can see that our dataset was almost perfect for this loss function—which isn't the case for arbitrary applications and datasets like cars, cub200 or online products. Despite that, we only got minor performance gains.

Taking into account that we performed a grid search to find the most optimal delta, our results suggest that our predisposed assumption that 2 image classes belonging to the same query end up being closer to each other in the neural network doesn't hold up many times, even if it does on a grand scale. This means that to get reasonable performance gains from hierarchical loss, we must go 1 step further and calculate the intuitive distance between all class pairs. However, even assuming that calculating such distance would give us performance gain, in practice, calculating such distances isn't possible without extensive feature engineering and a method that already does something close to deep metric learning. Therefore, in its current state, hierarchical loss doesn't provide enough performance gains to use over loss functions that make no use of predisposed

information despite having potential and sound presumptions.

Example

Even after looking at the evaluation of our deep metric learning method though, it's hard to visualize how a model with these scores would perform in our exploration framework. So, we will now look at an example of the model handling images.

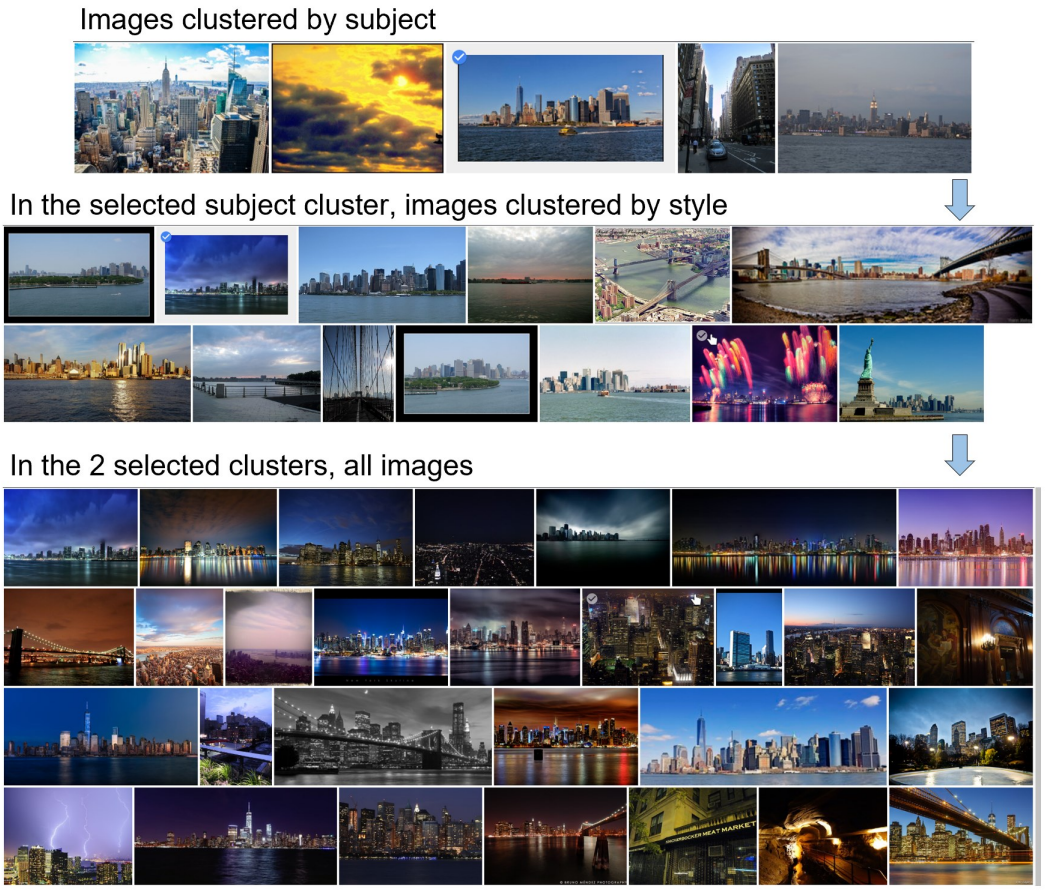

Figure 6: Explorer using DML with 5k images from Flickr with the query "New York"

In this example, we train the subject category over the places190 dataset mentioned above, and style category over the predictions generated by [11]'s CNN model for the places190 dataset. We train the models with hierarchical loss/distance sampling and N-pair loss/N-pair sampling for subject and style analysis respectively. After training, we pass forward over test images through the

network, run k-means clustering over a small subset of the images (10%) to determine the number of classes by silhouette score, and then run k-means clustering over the optimal number of classes.

As we can see from **Figure 6**, the explorer performs reasonably well, and more importantly, is interpretable. The subject clusters can be reasonably labeled as sky scrapers, nature, bay, streets and another view of the bay respectively. The same is true with style clusters, where we can also reasonably guess the specific style each cluster belongs to. This interpretability allows our explorer to fulfill its job, and guide the users to the specific images they want very quickly.

Despite the fact that we didn't explicitly predict the meaning behind the cluster, because we use many small classes to train our deep metric learning model, our model could generate embeddings that could reflect the meaning. And so, with a dataset that mines more representative samples and deals well with noise, exploration can be even better.

CHAPTER V

CONCLUSION

In this thesis, we applied deep metric learning to image exploration to connect users to specific images when traditional search fails. The main benefit of exploration is its fundamental ability to inherently work around the need to map clusters to natural language. Because of this design advantage, we can effectively work around NLP problems around breaking down complex queries and still serve the images with complex filters quickly and effectively.

Alongside the concept, we also employed deep metric learning to handle exploration, managing unseen image classes and shifting definitions of the image clusters well. We provided a visualization of what exploration with deep metric learning can provide, and to specifically better deal with unseen or under-represented image classes in the subject category in our exploration implementation, we also explored using hierarchical loss. Hierarchical loss attempted to exploits the predisposed information about the structure of subject analysis in order to get better quality embeddings with few samples. However, hierarchical loss in our strict aggregate estimate approach wasn't able to perform well enough to warrant using it over traditional methods.

Despite that, there are still many avenues future work can explore based on our results. In particular, hierarchical loss still presents interesting problems worth investigating. Developing a way to automatically estimate class-wise intuitive distance without having to do deep metric learning or feature engineering would particularly help to make hierarchical loss viable. Outside of technical aspects, future work can also explore using our methods to connect users to images outside of exploration. With embeddings from this work, it would be interesting to try to tackle the problem in this work in the original form, and work to label clusters of images with meaning and break down the query to match it to a cluster. Presumably, the aesthetic style and subject analysis of images can also help to close the gap between user's inherent preferences and our approximations of them.

REFERENCES

[1] "The world's most-beloved, money-losing business needs your help." Web, 2019.

[2] "Introducing pinterest trends, showing the power of insights irl." Web, 2019.

[3] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, *et al.*, "Neural information retrieval: At the end of the early years," *Information Retrieval Journal*, vol. 21, no. 2-3, pp. 111–182, 2018.

[4] R. Van Zwol, B. Sigurbjornsson, R. Adapala, L. Garcia Pueyo, A. Katiyar, K. Kurapati, M. Muralidharan, S. Muthu, V. Murdock, P. Ng, *et al.*, "Faceted exploration of image search results," in *Proceedings of the 19th international conference on World wide web*, pp. 961–970, 2010.

[5] B. Zheng, W. Zhang, and X. F. B. Feng, "A survey of faceted search," *Journal of Web engineering*, vol. 12, no. 1&2, pp. 041–064, 2013.

[6] G. Strong, O. Hoeber, and M. Gong, "Visual image browsing and exploration (vibe): User evaluations of image search tasks," in *International Conference on Active Media Technology*, pp. 424–435, Springer, 2010.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[10] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2408–2415, IEEE, 2012.

[11] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, "Recognizing image style," *arXiv preprint arXiv:1311.3715*, 2013.

[12] K. Song, Y. Tian, W. Gao, and T. Huang, "Diversifying the image retrieval results," in *Proceedings of the 14th ACM international conference on Multimedia*, pp. 707–710, 2006.

[13] L. Qin, J. X. Yu, and L. Chang, "Diversifying top-k results," *arXiv preprint arXiv:1208.0076*, 2012.

[14] D. Rafiei, K. Bharat, and A. Shukla, "Diversifying web search results," in *Proceedings of the 19th international conference on World wide web*, pp. 781–790, 2010.

[15] F. Radlinski and S. Dumais, "Improving personalized web search using result diversification," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 691–692, 2006.

[16] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *Proceedings of the second ACM international conference on web search and data mining*, pp. 5–14, 2009.

[17] C. L. Clarke, N. Craswell, and I. Soboroff, "Overview of the trec 2009 web track," tech. rep., WATERLOO UNIV (ONTARIO), 2009.

[18] C. L. Clarke, N. Craswell, and E. M. Voorhees, "Overview of the trec 2012 web track," tech. rep., NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD, 2012.

[19] B. Ionescu, A.-L. Gînsca, B. Boteanu, A. Popescu, M. Lupu, and H. Müller, "Retrieving diverse social images at mediaeval 2014: Challenge, dataset and evaluation.,"

[20] B. Ionescu, A.-L. Radu, M. Menéndez, H. Müller, A. Popescu, and B. Loni, "Div400: a social image retrieval result diversification dataset," in *Proceedings of the 5th ACM Multimedia Systems Conference*, pp. 29–34, 2014.

[21] T. Kenter, A. Borisov, C. Van Gysel, M. Dehghani, M. de Rijke, and B. Mitra, "Neural networks for information retrieval," in *WSDM 2018*, ACM, 2018.

[22] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li, "Justclick: Personalized image recommendation via exploratory search from large-scale flickr images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 273–288, 2008.

[23] W. Niu, J. Caverlee, and H. Lu, "Neural personalized ranking for image recommendation," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 423–431, 2018.

[24] C.-H. Lai, S.-J. Lee, and H.-L. Huang, "A social recommendation method based on the integration of social relationship and product popularity," *International Journal of Human-Computer Studies*, vol. 121, pp. 42–57, 2019.

[25] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *CoRR*, vol. abs/1310.1531, 2013.

[26] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

[27] J. Lu, J. Hu, and J. Zhou, "Deep metric learning for visual understanding: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 76–84, 2017.

[28] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1875–1882, 2014.

[29] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, (Sydney, Australia), 2013.

[30] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[31] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*, pp. 84–92, Springer, 2015.

[32] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 360–368, 2017.

[33] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017.

[34] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[35] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.

[36] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in neural information processing systems*, pp. 1857–1865, 2016.