Ultra Fine-Grained Image Semantic Embedding

Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, Yi-Ting Chen, Yaxi Gao,
Tom Duerig, Andrew Tomkins, Sujith Ravi

Google Research
Mountain View, CA
{dacheng,chunta,zhenli,futangpeng,altimofeev,yitingchen,gyxlucy,tduerig,tomkins,sravi}@google.com

ABSTRACT

"How to learn image embeddings that capture fine-grained semantics based on the instance of an image?" "Is it possible for such embeddings to further understand image semantics closer to humans' perception?" In this paper, we present, Graph-Regularized Image Semantic Embedding (Graph-RISE), a web-scale neural graph learning framework deployed at Google, which allows us to train image embeddings to discriminate an unprecedented $O(40M)$ ultra-fine-grained semantic labels. The proposed Graph-RISE outperforms state-of-the-art image embedding algorithms on several evaluation tasks, including kNN search and triplet ranking: the accuracy is improved by approximately 2X on the ImageNet dataset and by more than 5X on the iNaturalist dataset. Qualitatively, image retrieval from one billion images based on the proposed Graph-RISE effectively captures semantics and, compared to the state-of-the-art, differentiates nuances at levels that are closer to human perception.

CCS CONCEPTS

• **Computing methodologies** → **Image representations**; • **Information systems** → *Web searching and information discovery*;

KEYWORDS

Image embeddings, semantic understanding, graph regularization

ACM Reference Format:
Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, Yi-Ting Chen, Yaxi Gao, Tom Duerig, Andrew Tomkins, Sujith Ravi. 2020. Ultra Fine-Grained Image Semantic Embedding. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3336191.3371784

1 INTRODUCTION

Learning image embeddings that capture fine-grained semantics is the core of many modern image-related applications such as image search, either querying by traditional keywords or by an example query image [14]. Albeit its importance, learning such embeddings is a challenging task, partly due to the large variations seen among images that belong to the same category or class.

WSDM '20, February 3–7, 2020, Houston, TX, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6822-3/20/02.
https://doi.org/10.1145/3336191.3371784

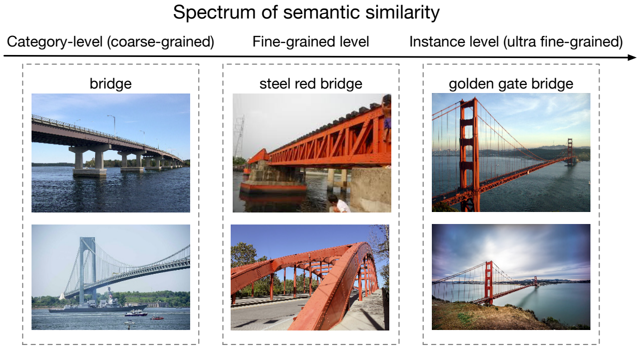

Spectrum of semantic similarity
Category-level (coarse-grained)
Fine-grained level
Instance level (ultra fine-grained)
bridge
steel red bridge
golden gate bridge

Figure 1: Spectrum of image semantic similarity. We provide six image examples (two for each granularity) to illustrate the difference from coarser (left) to ultra-fine granularity (right). We refer to ultra fine-grained as "instance-level" to contrast with category-level and fine-grained semantics.

Several previous works consider category-level image semantics [9, 26], in which two images are considered semantically similar if they belong to the same category. As illustrated in Figure 1, category-level similarity may not be sufficient for modern vision-based applications such as query-by-image, which often require the distinction of nuances among images within the same category.

Recently, deep ranking models [29, 30] have been proposed to learn fine-grained image similarity. These ranking models are trained with image triplets, where each entry contains {query image, positive image, negative image}. The goal is to rank (query, positive image) as more similar than (query, negative image); this training formulation can encode distinctions that are as fine-grained as the construction of the triplets allows. In practice, however, it becomes increasingly difficult to generate a large corpus of triplets that encode sufficiently fine-grained distinctions by ensuring the negative image is similar enough to the query image, but not too similar. Furthermore, since human raters need to be involved to provide the triplet ranking ground truth, collecting high quality image triplets for training is costly and labor-intensive. We instead propose moving from triplet learning to a classification framework that learns embeddings capable of associating an image to one of a large number of possible query strings.

Such an approach produces image embeddings that are predictive of queries that might lead to the image. In addition, we also obtain similarity data between the images themselves, encoding for example the fact that two images were both clicked in a particular setting. This relational data encodes important aspects of human image perception but is not easily encapsulated by labels (*i.e.*, image-label pairs for training). To incorporate image-image similarity into the training we employ neural graph learning, in which the model is trained by minimizing the supervised loss combined with a graph regularizer that drives the model to reduce embedding distance between similar image pairs.

To the best of our knowledge, this work brings the following contributions:

- **Effective embedding learning via large scale classification.** We formulate the problem of image embedding learning as an image classification task at an unprecedented scale, with label space (*i.e.*, total number of classes) in $O(40M)$ and the number of images in $O(260M)$. This is **the largest scale** in terms of number of classes, and one of the largest in terms of images used for learning image embeddings. Furthermore, the proposed model is one of the **largest vision models** in terms of the number of parameters (see Section 5.1 and Section 5.2). No previous literature has demonstrated the effectiveness of such a large-scale image classification for learning image representation.
- **Neural graph learning on image representation.** We propose a neural graph learning framework that leverages graph structure to regularize the training of deep neural networks. This is the first work deploying large-scale neural graph learning for image representation. We will describe below two techniques to construct image-image graphs based on “co-click” rate and “similar-image click” rate, designed to capture ultra-fine-grained notions of similarity that emerge from human perception of result sets.
- **Graph-RISE for instance-level semantics.** We present a deployed framework at Google: Graph-RISE, an image embedding that captures ultra-fine-grained, instance-level semantics. Graph-RISE outperforms the state-of-the-art algorithms for learning image embeddings on several evaluations based on k-Nearest-Neighbor (kNN) search and triplet ranking. Experimental results show that Graph-RISE improves the Top-1 accuracy of the kNN evaluation by approximately 2X on the ImageNet dataset and by more than 5X on the iNaturalist dataset. Case studies also show that, qualitatively, Graph-RISE outperforms the state of the art and captures instance-level semantics.

The remainder of this paper is organized as follows. Section 2 provides related work on learning image embeddings. Section 3 formulates the problem and provides the details of training datasets. Section 4 explains the proposed learning algorithms, followed by Section 5 with the details of network architecture and training infrastructure. Section 6 shows the experimental results and Section 7 concludes this paper.

2 RELATED WORK

There are several prior works on learning image similarity [8, 26, 28]. Most of them focus on category-level image similarity, in which two images are considered to be similar if they belong to the same category. In general, visual and semantic similarities tend to be consistent with each other across category boundaries [6]. Visual variability within a semantically-defined category still exists, especially for broadly defined categories such as “animal,” or “plant,” as a result of the broad semantic distinctions within such classes. As classes become finer grained, however, the visual distinctions within a class due to natural variations in image capture (angle, lighting, background, etc) become larger relative to the fine distinctions between classes that are semantically closer; hence, new techniques are required.

For learning fine-grained image similarity, local distance learning [7] and OASIS [5] developed ranking models based on hand-crafted features, trained with triplets wherein each entry contains {query image, positive image, negative image} that characterizes the ranking orders based on relative similarity. In [29], a DeepRanking model that integrates the deep learning and ranking model is proposed to learn a fine-grained image similarity ranking model directly from images, rather than from hand-crafted features. As discussed above, while these ranking models have been widely used for learning image embeddings, the model performance relies heavily on the quality of triplet samples, which involves pair-wise comparisons that can be costly and labor-intensive to collect. As we will show later in Section 3 and Section 6, Graph-RISE does not require models to be trained by triplets and outperforms the state-of-the-art on capturing image semantics for several evaluation tasks.

There has also been a significant amount of work on improving image classification to near-human levels [18] by increasing the representational capacity and the depth of network architectures. See, *e.g.*, VGG-19 [20], Inception [25], and ResNet [10]. To support learning such deep networks with millions of parameters, large-scale datasets such as ImageNet [13], iNaturalist [27] and YouTube-8M [2] have played a crucial role. For example, the authors of [15] demonstrate that the rich mid-level image features learned by Convolutional Neural Networks (CNNs) on ImageNet can be efficiently transferred to other visual recognition tasks with limited amount of training data. Their study suggests that the number of images and the coverage of classes for training in the source task are important for the performance in the target task. In [22], the authors reveal a logarithmic relationship between the performance on vision tasks and the amount of training data used for representation learning. In this paper, we share the same observation and further show that when increasing the number of classes to $O(40M)$ with sufficient amount of training data, the purposed Graph-RISE is able to capture instance-level, ultra-fine-grained semantics.

3 PROBLEM FORMULATION

In this section, we formulate the task of learning an image embedding model, and then provide the details of training dataset used for this task.

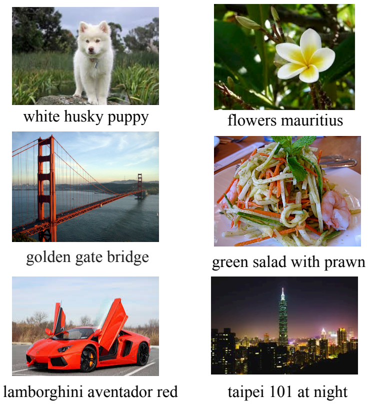
Problem Formulation. Given the following inputs:

- A labeled set $\mathcal{D}_L$ that contains image-label pairs (x, y), where label y provides ultra-fine-grained semantics to the corresponding image x.
- An unlabeled set $\mathcal{D}_U$ that contains images without labels.

The objective is to find an image embedding model that achieves instance-level semantic understanding. Specifically, let $\phi(\cdot)$ represent a function that projects an image to a dense vector representing an embedding; given two images x_1 and x_2, the similarity in the

Table 1: List of symbols.

Symbol	Definition and description
(x, y)	Image-label pair
$\mathcal{D}_L$, $\mathcal{D}_U$	Labeled dataset, unlabeled dataset
$\mathcal{K}$	Total number of classes
$\mathcal{E}$	Set of edges
$w_{u,v}$	Edge weight
η	Margin for triplet distance
$\phi(\cdot)$	Embedding mapping function
$d(\cdot)$	Distance function
$p(\cdot)$, $q(\cdot)$	Probability and ground-truth distribution
$\mathcal{L}(\cdot)$	Loss function
$\Omega(\cdot)$	Graph regularizer
$\boldsymbol{\theta}$	Model parameters


white husky puppy

flowers mauritius

golden gate bridge

green salad with prawn

lamborghini aventador red

taipei 101 at night

Figure 2: Six potential samples of image-query pairs. Each image is labeled with the corresponding textual search query.

image embedding space is defined according to a distance metric: $d(\phi(x_1), \phi(x_2))$, where $d(\cdot)$ is a distance function (*e*.g., Euclidean distance or cosine distance). If x_1 and x_2 belong to the same class, an ideal $\phi(\cdot)$ minimizes the distance between $\phi(x_1)$ and $\phi(x_2)$, indicating these two images to be semantically similar. Table 1 provides the symbols and the corresponding definitions used throughout this paper.

Dataset. In order to achieve instance-level semantic understanding, the classes should be ultra-fine-grained. Thus, we created a training dataset $\mathcal{D}_L$ derived from Google Image Search. It contains approximately 260 million images; selection data are used to characterize how frequently anonymous users selected an image when that image appeared in the results of a particular textual search query. After the characterization, a search query is then treated as the "label" annotated to the corresponding image to provide semantics as labeled samples. Each image is labeled with one or more queries (2.55 queries per image on average), and the total number of unique queries (used as classes) is around forty million. Figure 2 illustrates six potential image-query pairs[1]. To the best of our knowledge, this is the largest scale of training data for learning image embedding in terms of the number of classes[2], and one of the largest in terms of the number of training images [19, 22, 29].

Unlabeled dataset $\mathcal{D}_U$ contains approximately 20 million images, without any annotation or labeling. Unlabeled dataset is mainly for constructing similarity graphs (see Section 4.2) and for evaluation purposes (see Section 6.1).

4 LEARNING ALGORITHMS

In this section, we first introduce the algorithm that takes image-query pairs as training data in a supervised learning manner. Then we elaborate on neural graph learning, a methodology incorporating graph signals into the learning algorithm.

4.1 Discriminative Embedding Learning

From triplet loss to softmax loss. In order to train image embedding models, metric learning objectives (such as contrastive loss [9, 23] and triplet loss [29]), and classification objectives (such as logistic loss and softmax loss) have been widely explored. When using metric learning objectives, collecting high quality samples (*e*.g., triplets) is often challenging. Furthermore, optimizing metric learning objectives suffers from slow convergence or poor local optima if sampling techniques are inappropriately applied [21, 30]. In order to achieve instance-level semantic understanding, we employ softmax loss for training the image embedding model. When the size of classes is sufficiently large, $O(40M)$ in our case, classification training (with softmax loss) works better than triplet loss [29].

For each training example x, the probability of each label $k \in \{1, \ldots, \mathcal{K}\}$ in our model is computed via softmax:

$$p(k|x) = \frac{exp\{z_k\}}{\sum_{i=1}^{\mathcal{K}} exp\{z_i\}} \tag{1}$$

where z_i are the logits or unnormalized log probabilities. Here, the z_i are computed by adding a fully connected layer on top of the image embeddings, i.e., $z_i = \mathbf{W}_i^T \phi(x) + b_i$, where $\mathbf{W}_i$ and b_i are weights and bias for target label, respectively. Let $q(k|x)$ denote the ground-truth distribution over classes for this training example such that $\sum_{i=k}^{\mathcal{K}} q(k|x) = 1$. As one image may have multiple ground-truth labels, $q(k|x)$ is uniformly distributed to the ground-truth labels. The cross-entropy loss for the example is computed as:

$$\ell = -\sum_{k=1}^{\mathcal{K}} log(p(k|x))q(k|x) \tag{2}$$

While softmax loss works well when the number of classes is not large (say 10K or 100K), several challenges arise if the number of classes is increased to millions or even billions. First, the computational complexity involved in computing the normalization constant of the target class probability $p(k|x)$ is prohibitively expensive [3, 11]. Second, as the training objective encourages the logits corresponding to the ground-truth labels to be larger than all other logits, the model may learn to assign full probability to the ground-truth labels for each training example. This would result in over-fitting and make the model fail to generalize [25].

[1] In this paper, "image-query pairs" and "image-label pairs" are used interchangeably since queries are used as labels.

[2] In prior arts (such as [29]), the training dataset contains up to $O(15M)$ samples with $O(100K)$ classes.

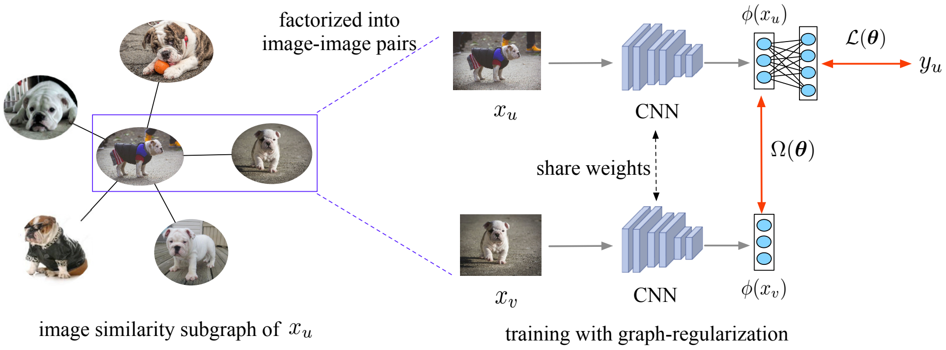

factorized into image-image pairs
$\phi(x_u)$
$\mathcal{L}(\boldsymbol{\theta})$
y_u
x_u
CNN
$\Omega(\boldsymbol{\theta})$
share weights
CNN
x_v
$\phi(x_v)$
image similarity subgraph of x_u
training with graph-regularization

Figure 3: An illustration of a graph-regularized neural network. The image similarity subgraph of a training image x_u (with the ground-truth labels y_u) is factorized into image-image pairs, where the neighbor image x_v is semantically similar to x_u. The training objective consists of both the supervised loss $\mathcal{L}$ and the graph regularization Ω; minimizing Ω drives the distance between the embeddings of similar images—$\phi(x_u)$ and $\phi(x_v)$—to be minimized, which means the neural network is trained to encode the local structure of a graph.

Instead of computing the normalization constant for all the classes, we sample a small subset of the target vocabulary $L \in \{1, \ldots, \mathcal{K}\}$ to compute the normalization constant in $p(k|x)$ for each parameter update. Then, the target label probability can be computed as:

$$p'(k|x) = \frac{exp\{z_k\}}{\sum_{i \in L} exp\{z_i\}} \tag{3}$$

which leads to much lower computational complexity and allows us to efficiently use Tensor Processing Units (TPUs) [12] to train this deep image embedding model with sampled softmax.

Furthermore, to discourage the model from assigning full probability to the ground-truth labels (and therefore becoming prone to over-fitting), we follow [25] to "smooth" the label distribution by replacing the label distribution with a mixture of the original ground-truth distribution $q(k|x)$ and the fixed distribution $u(k)$:

$$q'(k|x) = (1 - \epsilon)q(k|x) + \epsilon u(k) \tag{4}$$

where $u(k)$ is a uniform distribution over the sampled vocabulary $u(k) = \frac{1}{|L|}$, and ϵ is a smoothing parameter.

Finally, the discriminative objective for training the neural network can be defined as the cross-entropy of the target label probability on the sampled subset and the smoothed ground-truth distribution:

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{x_i \in \mathcal{D}_L} \sum_{k \in L_i} log(p'(k|x_i))q'(k|x_i) \tag{5}$$

where $\boldsymbol{\theta}$ denotes the neural network parameters. The ground-truth labels of x_i are always selected within the sampled labels L_i. In our experiments, we randomly sample 100K classes for each training instance and ϵ is selected to be 0.1.

4.2 Neural Graph Learning

While the discriminative objective indeed paves the way to learning an image representation that captures fine-grained semantics, there is more information available in human interactions with images. Many of these additional data sources can be represented as graphs (such as image-image co-occurrence), and yet current vision models (*e.g.*, ResNet) cannot consume such graphs as inputs. Thus, we propose to train the network using graph structure about the relationships among images. In particular, images that are more strongly connected in the graph should reflect stronger semantic similarity based on user feedback (see Section 4.3 for the details of graph construction), and should be closer in the embedding space. To achieve this goal, we deploy graph regularization [4] to train the neural network for encouraging neighboring images (from the graph) to lie closer in the embedding space. The final graph-regularized objective is the sum of the discriminative loss and the graph regularization:

$$\mathcal{R}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \underbrace{\alpha \sum_{(u,v) \in \mathcal{E}} w_{u,v} d\Big(\phi(x_u), \phi(x_v)\Big)}_{\Omega(\boldsymbol{\theta})} \tag{6}$$

where $\Omega(\boldsymbol{\theta})$ denotes the graph regularizer, $\mathcal{E}$ represents a set of edges between images, $w_{u,v}$ represents the edge weight between image u and v, $\phi(\cdot)$ is the representation extracted from the embedding layer[3], $d(\cdot)$ is the distance metric function, and $\alpha \geq 0$ is the multiplier (applied on regularization) that controls the trade-off between the discriminative loss and the regularization induced from the graph structure. An illustration of the graph-regularized neural network is given in Figure 3.

The multiplier α (applied on regularization) controls the balance between the discriminative information (*i.e.*, predictive power) and the contributions of the graph structure (*i.e.*, encoding power). In other words, the neural network is trained to both (a) make accurate classification (or prediction), and (b) encode the graph structure. When $\alpha = 0$, the proposed objective ignores the graph regularization and degenerates to a neural network with only supervised objective in Eq. (5). On the other hand, when $p'(x) = \phi(x)$, where $p'(x)$ is the predicted label distribution, we have a label propagation objective as in [17] by training with the objective using $p'(x)$ directly without parameters $\boldsymbol{\theta}$ (*i.e.*, no neural network involved),

[3] In general, the embedding layer refers to the layer right before the softmax layer.

and letting the distance function $d(\cdot)$ and the loss function $\ell(\cdot)$ to be mean squared errors (MSE). The label propagation objective encourages the learned label distribution $p'(x)$ of similar images to be close. Furthermore, the label distribution of a sample is aggregated from its neighbors to adjust its own distribution. Thus, the proposed objective in Eq. (6) could be viewed as a "graph-regularized version of neural network objective" or as a "non-linear version of label propagation."

4.3 Graph Construction

In this section, we provide the details for constructing graphs used by the regularizer $\Omega(\boldsymbol{\theta})$ in Eq. (6). In addition to image-query pairs (described in Section 3) where images are annotated by queries for obtaining semantics, we propose to find "image-image" pairs where the semantics shared by these two images are closer to human perception and beyond textual search queries. Each image is treated as a "vertex" and an image-image pair is treated as an "edge," together forming a graph that can be included as ancillary training data.

Specifically, each image-image pair contains one source vertex $x_u \in \mathcal{D}_L$ and one target vertex $x_u \in \{\mathcal{D}_L \bigcup \mathcal{D}_U\}$. In this work, we introduce two methods to construct edges: (a) based on co-click rate of the image pair, and (b) based on similar-image click rate of the image pair. The co-click rate of the image pair characterizes how often users select both the source image x_u and the target image x_v in response to both x_u and x_v being concurrently identified by search results from a textual search query. This type of image-image relationship sheds light on the question: "Given that one image is selected from the resulting images, what other images that are sufficiently similar will also be selected?" If the co-click rate between x_u and x_v is higher than a pre-defined threshold, x_u and x_v are considered to be sufficiently similar and an edge between them is constructed; the edge weight $w_{u,v}$ is calculated based on the co-click rate. Then x_u and x_v will be used for calculating the graph regularization $\Omega(\boldsymbol{\theta})$ in Eq. (6).

Different from the co-click rate, the similar-image click rate of the image pair characterizes how often users select the source image x_u in response to x_u being identified by a search result for a search query using the target image x_v (instead of a textual query). This type of image-image relationship sheds the light upon the question: "Given an image issued as the query, what other images that are sufficiently similar will be selected in response to the query image?" Similar with how edges are constructed based on the co-click rate, if the similar-image click between x_u and x_v is higher than a pre-defined threshold, x_u and x_v are considered to be sufficiently similar and an edge between them is constructed. Edge weight $w_{u,v}$ is calculated based on the similar-image click rate.

5 TRAINING FRAMEWORK

In this section, we provide the details of network architecture and training infrastructure along with the configurations we used in this work.

5.1 Network Architecture

Figure 4 illustrates the proposed network architecture. The main model is the 101-layer ResNet (referred as ResNet-101) [10]. Compared to Inception [24], ResNet-101 has larger model capacity, which yields more than 2% of performance improvement on our internal metric for embedding evaluation. While the major architecture of ResNet-101 remains unchanged, several detailed configurations have been modified. The input layer is modified to take enlarged input images from 224×224 to 289×289 pixels. The output 10×10×2K feature map is first avg pooled to 4×4×2K using a 4×4 kernel of stride 2, and then flattened and projected to 64-dimensional layer representing image embeddings. The activation function is ReLU-6[4]. Finally, a softmax layer is added to produce a multinomial distribution across 40 million classes (*e.g.*, queries).

During training, both training samples and their neighbors provided from graphs (described in Section 4.3) are fed into the model for enabling graph regularization; the 64-dimensional embedding layer is selected as the target layer to be regularized as described in Eq. (6). Furthermore, 100K out of 40 millions labels are sampled via an important sampling technique [3] for each parameter update Eq. (3). Finally, batch normalization is applied during training.

During the inference phase, a 64-dimensional L2 normalized embedding [5] is generated for each input image as a new, semantically-meaningful representation. Note that neighbors and graph regularization is not required when making inference (see the flow in red in Figure 4). In addition to the embedding, the queries with the top-k predicted probabilities are also outputted. Since the focus of this paper is image embedding, the output queries will largely be ignored in the rest of this paper.

5.2 Training Infrastructure

We implement the network architecture described in Section 5.1 using TensorFlow[1]. The details of training configurations are as follows. We select the batch size to be 24, and the momentum [16] as the optimizer; the initial learning rate is 0.001 and will be decayed with an exponential rate of 0.9 every 100,000 steps. The label smoothing ϵ is 0.1. The multiplier for applying L2 regularization (a.k.a. "weight decay") is 0.00004. For configurations related to graph regularization, the multiplier for applying graph regularization α is 1.0, and the distance function $d(\cdot)$ in Eq. (6) is selected to be cosine distance[6]. For constructing graphs, the threshold is set to 0.1, and we combine the edges (approximately 50 million edges, built from co-clicks and similar-image clicks) into one graph for calculating regularizer $\Omega(\boldsymbol{\theta})$ in Eq. 6.

Since our model is one of the largest vision models in terms of number of parameters (40M ×64 plus the parameters of ResNet-101 architecture), together with the scale of the training dataset, using TPUs [12] to train the model is the only feasible solution. The training is distributed to 8×8 TPU cores, and takes two weeks to converge from scratch after 5M steps. Training with graph regularization costs additional computation that grows with the number

[4] ReLU-6 computes Rectified Linear Unit as: min(max(x, 0), 6).

[5] L2 normalization is not applied on embedding during training as it makes the training hard to converge.

[6] We have also experimented with using the Euclidean (L2) distance as $d(\cdot)$; when using Euclidean distance with $\alpha = 0.01$, it achieves almost the same performance as using cosine distance with $\alpha = 1$

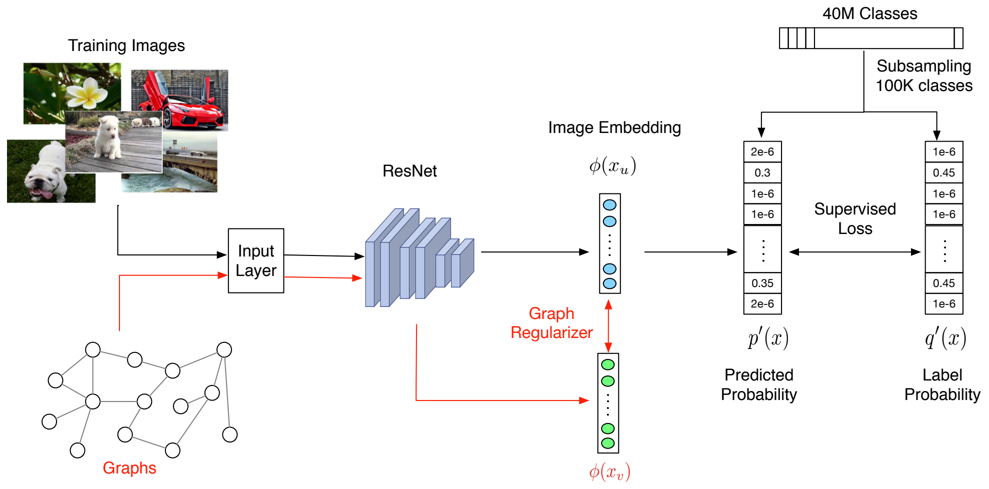


Figure 4: An illustration of the Graph-RISE framework. Flow in red is added to enable graph regularization and required only during training. In the input layer, a labeled image is associated with one of its neighbor images, which can be either labeled or unlabeled, and then fed into the ResNet together with its neighbor image. Then, the image embeddings generated from ResNet are used to both (a) compute the cross-entropy loss and (b) graph regularization.

of neighbors used (in this work, approximately 40% training time increase due to the use of neighbors). To reduce the computation, we compare the performance of two models: one is trained without graph regularization from scratch; the other is trained with graph regularization with α set to zero until it is converged, and then set $\alpha = 1$ to fine-tune the model. We find that the performance of the first model (w/o graph regularization) is slightly better in the beginning of the training, and two models achieves almost the same performance after 4M steps. With these setting, the fine-tuning model (the one with graph regularization) takes approximately 2 additional days to train for 500K extra steps until convergence.

6 EXPERIMENTS

In this section, we explain the details of evaluation setup, and then show the experimental results for performance evaluation. We also provide the case studies for the qualitative analysis.

6.1 Evaluation Setup

We are interested in providing image embeddings with instance-level semantics, such that the similarity between embeddings approximates the relationships (of images) in terms of semantics. To evaluate the performance of the proposed image embedding model, we conduct both k-Nearest-Neighbor (kNN) search and triplet evaluation as metrics; these two are the most popular methods used for evaluating embedding models.

For kNN evaluation, we conduct experiments on ImageNet [13] and iNaturalist [27] datasets, and then report two metrics in the experiments: Top-1 and Top-5 accuracy, where Top-k accuracy calculates the percentage of query images that find at least 1 image—from the top k searched image results—carrying the exact same labels as the query images. For the ImageNet dataset, the images in the validation set are used as the query images, and the training set is used as the index set to search for top-k results. For the iNaturalist dataset, for each class two images are randomly sampled to construct the query set, and the remainder of the images in that class are used as the index set.

For triplet evaluation, we follow the evaluation strategy in [29] to sample triplets (A, P, N)—representing Anchor, Positive, Negative images—from Google Image Search and ask human raters to verify if P is more semantically closer to A than N. We sample the triplets in a way such that A and P have the same or a very similar instance-level concept, and N is slightly less relevant ("hard-negative"). Each triplet is independently ranked by three raters and only the triplets receiving unanimous scores are used for evaluation. Assume positive image P is rated to be more similar to anchor image A than negative image N, the prediction of a model is considered to be accurate if the following condition holds:

$$\eta + d(\phi(A), \phi(P)) - d(\phi(A), \phi(N)) < 0 \tag{7}$$

where η is the hyper-parameter that controls the margin between the distance of two image projections.

Two triplet datasets are created for calculating triplet evaluation metrics: (a) Product-Instance-Triplets (PIT) is a dataset designed to focus on evaluating the semantic concepts of images in the commercial product vertical, which consists of 10,000 triplets; (b) Generic-Instance-Triplets (GIT) is a dataset focusing on evaluating the semantic concepts of general images, including all possible image verticals from Google Image Search, consisting of 14,000 triplets.

6.2 Model Comparisons

We compare the proposed method with the following state-of-the-art models:

- DeepRanking model [29] that employs triplet loss on multi-scale Inception network architecture [25] with an online triplet sampling algorithm.
- Inception network architecture [25] that employs sampled softmax loss over 8 millions labels.

	ImageNet [13]		iNaturalist [27]	
	Top-1	Top-5	Top-1	Top-5
DeepRanking	35.20	60.93	6.03	13.71
Inception (8M)	61.92	84.77	17.30	34.58
ResNet (8M)	62.49	84.65	17.36	34.15
ResNet (40M)	66.20	86.41	27.05	47.44
Graph-RISE (40M)	68.29	87.75	31.12	52.76

Table 2: Performance comparisons (in %) via kNN search accuracy on publicly available datasets.

	PIT	GIT
DeepRanking	74.95	77.25
Inception (8M)	82.81	86.54
ResNet (8M)	85.46	87.72
ResNet (40M)	86.70	88.90
Graph-RISE (40M)	87.16	89.53

Table 3: Performance comparisons (in %) via triplet accuracy ($\eta = 0$) on the internal evaluation datasets.

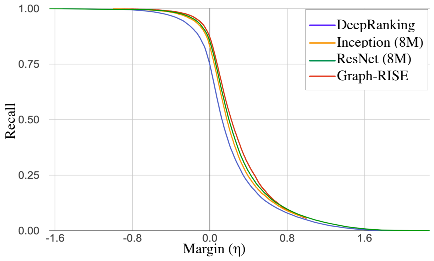


Figure 5: PIT triplet evaluation on Recall v.s. Margin.

- ResNet-101 network architecture [10] that employs sampled softmax loss over 8 millions and 40 millions labels (referred as ResNet (8M) and ResNet (40M) in Table 2, respectively).
- Graph-RISE model based on ResNet-101 network architecture proposed in Section 4.2.

The input layers in DeepRanking model and Inception model both use 224×224 image pixels, while the ResNet-101 and Graph-RISE use 289×289. Label smoothing is applied to all the classification-based models. When the graph regularization multiplier $\alpha = 0$, Graph-RISE is equivalent to the ResNet-101 model. In all the experiments, the Euclidean (L2) distance of the embedding vectors extracted from the penultimate layer—the layer before the final softmax or ranking layer—is used as similarity measure. To evaluate the effectiveness of image embeddings, no individual fine-tuning is performed for each dataset, and all the experiments are conducted directly based on the learned embeddings of the input images.

6.3 Performance Evaluation

Table 2 provides the performance comparisons (in terms of percentage) on kNN evaluations, and Table 3 shows the triplet evaluations (also in terms of percentage). From these results, we have several observations. First, Graph-RISE significantly outperforms the previous state-of-the-art [29] and other models without graph regularization in all the evaluation criteria. We attribute this to the fact that Graph-RISE leverages the graph structure via neural graph learning to drive the embeddings of similar images to be as close as possible. Notice that, compared to the previous state-of-the-art [29], Graph-RISE **improves the Top-1 accuracy by almost 2X** (from 35.2% to 68.29%) on ImageNet dataset, and by **more than 5X** (from 6.03% to 31.12%) on the iNaturalist dataset.

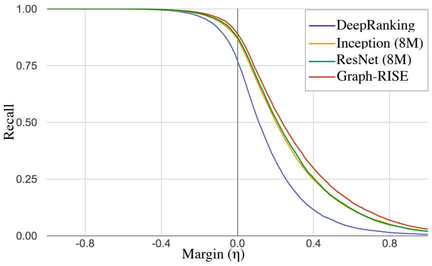


Figure 6: GIT triplet evaluation on Recall v.s. Margin.

Second, compared to the Inception network architecture, training image embeddings with ResNet-101 improves the performance for most datasets. This confirms the observation from [22] that to fully exploit $O(300M)$ images, a higher capacity model is required to achieve better performance. Furthermore, we confirm that sampled softmax is an effective technique to train image embedding model with datasets that have extremely-large label space; by comparing Inception (8M) and DeepRanking [29] (both based on Inception network), we observe that sampled softmax helps achieve better performance in triplet loss, even if DeepRanking directly aims at optimizing the triplet accuracy.

Moreover, increasing the number of labels (from 8M to 40M) significantly improves the kNN accuracy. We conjecture that the labels in 40M are more fine-grained than 8M, and therefore the learned image embeddings also need to capture fine-grained semantics in order to distinguish these 40M labels. In addition, we find that training ResNet-101 using larger input size (289×289) instead of smaller input size (224×224) also helps improve the model performance (from 85.13% to 86.7%, in terms of accuracy on PIT triplet evaluation), since larger input size encapsulates more detailed information from training images.

Figure 5 and Figure 6 depict the comparisons of triplet evaluation among four models: DeepRanking, Inception (8M), ResNet (8M) and Graph-RISE, on PIT dataset and GIT dataset respectively. Note that ResNet (40M) is ignored in the figures since the curves of ResNet (40M) and Graph-RISE are visually difficult to distinguish. In these two figures, x-axis is "Margin" and y-axis is "Racall" rate (the higher the better). "Margin" is the η in Eq. 7 representing the margin between the distance of "Anchor-Negative pair" and the distance of "Anchor-Positive pair." A large margin means "the Negative image is further away from the Anchor image than the Positive image." "Recall" rate represents the percentage of triplets that satisfy $\eta + d(\phi(A), \phi(P)) < d(\phi(A), \phi(N))$. From these two figures, the performance of Graph-RISE is consistently better the other models.

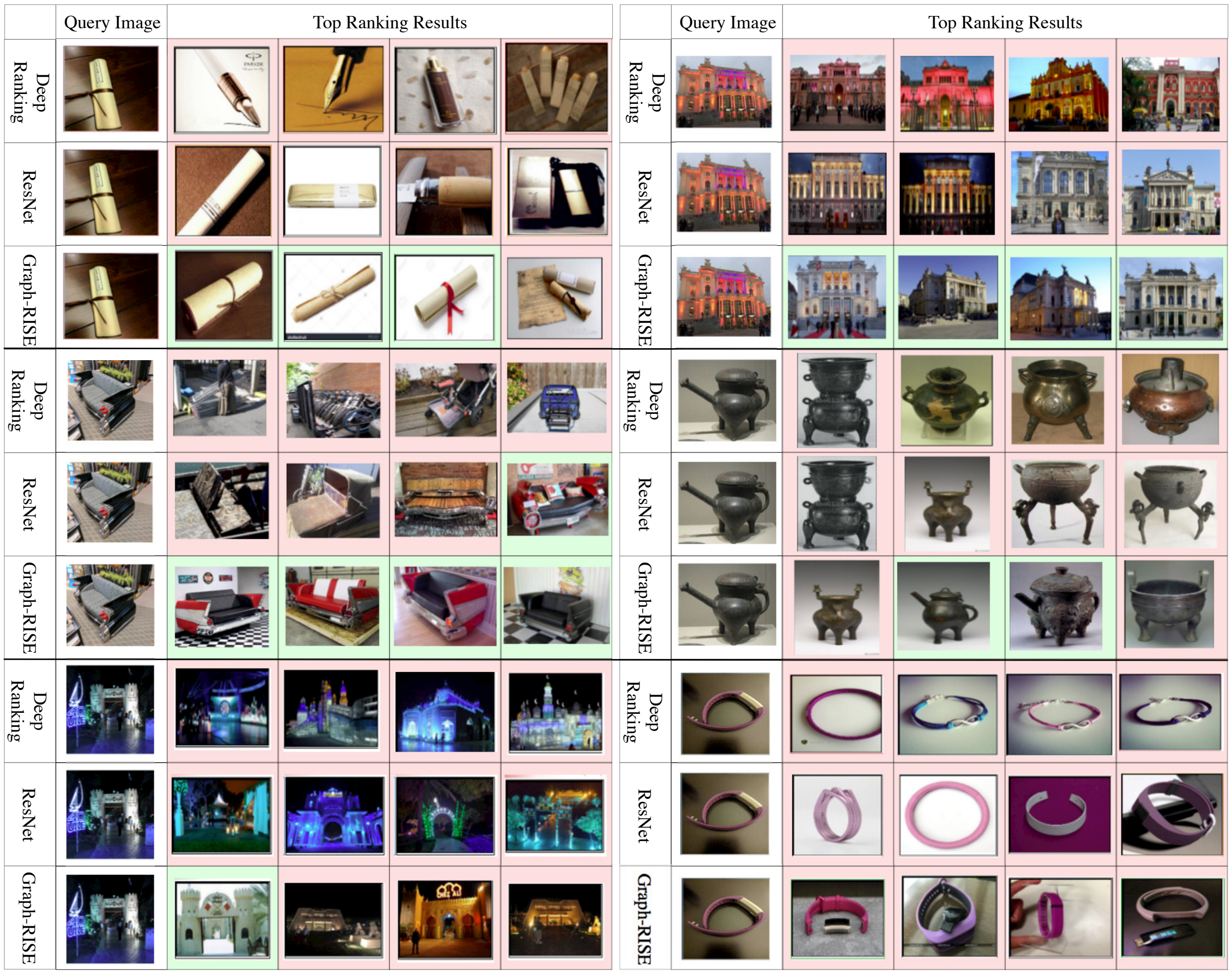


Figure 7: Retrieval results for 6 randomly-chosen query images. For each query image, we provide the Top-4 images retrieved (from 1 billion images) by DeepRanking, ResNet (40M), and Graph-RISE. Each retrieval result is rated and color-coded by human raters: retrieved images colored by green are rated to be strongly similar with the query image, whereas images colored by red are rated to be not (or somewhat) similar. Notice that images retrieved by Graph-RISE generally conform to experts' ratings.

6.4 Qualitative Analysis

Next, we evaluate the quality of images retrieved by DeepRanking [29], ResNet (40M), and Graph-RISE models. Given a randomly-selected query image, each method retrieves the most semantically similar images from an index containing one billion images. The top-ranked results are sent out to be rated by human experts. Figure 7 illustrates the retrieval results for 6 randomly-selected query images; for each query image we provide the Top-4 images retrieved. The images colored by green are rated to be strongly similar with the query image, whereas the images colored by red are rated to be not (or somewhat) similar. Compared to other models, images retrieved by Graph-RISE generally conform to experts' ratings, meaning that Graph-RISE captures the semantic meaning of images more effectively as judged by human raters. For example, given a query image of "scroll with ribbon" (top-left in Figure 7), the top three images retrieved by Graph-RISE are also "scroll with ribbon" (with similar colors, textures and shapes), and are rated as strongly similar by human experts. Another example is a query image of a landmark (top-right in Figure 7); Graph-RISE is able to retrieve images of the exact same landmark, while the other methods are only able to retrieve images of somewhat similar buildings.

In addition, we observe that the images retrieved by DeepRanking tend to be only visually similar to the query images, rather than semantically similar. This is probably because generating triplets that reflect the semantic concepts is very difficult, especially when the classes are ultra-fine-grained.

7 CONCLUSION

In this work, we present Graph-RISE to answer the motivational question: *"Is it possible to learn image content descriptors (a.k.a., embeddings) that capture image semantics and similarity close to human perception?"* Graph-RISE confirms that ultra-fine-grained, instance-level semantics can be captured by image embeddings extracted from training a sophisticated image classification model with large-scale data: $O(40M)$ classes and $O(260M)$ images. Graph-RISE is also the first image embedding model based on neural graph learning that leverages graph structures of similar images to capture semantics close to human image perception. We conduct extensive experiments on several evaluation tasks based on both kNN search and triplet ranking, and experimental results confirm that Graph-RISE consistently and significantly outperforms the state-of-the-art methods. Qualitative analysis of image retrieval tasks also demonstrates that Graph-RISE effectively captures instance-level semantics.

ACKNOWLEDGEMENT

We would like to thank Dr. Sergey Ioffe and Dr. Thomas Leung for the reviews and suggestions. We also thank the Google Expander, Image Understanding and several related teams for the technical support.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning.. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, Vol. 16. 265–283.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675* (2016).

[3] Yoshua Bengio and Jean-Sébastien Senécal. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks* 19, 4 (2008), 713–722.

[4] Thang D Bui, Sujith Ravi, and Vivek Ramavajjala. 2018. Neural Graph Learning: Training Neural Networks Using Graphs. In *ACM International Conference on Web Search and Data Mining (WSDM)*.

[5] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research* 11, Mar (2010), 1109–1135.

[6] Thomas Deselaers and Vittorio Ferrari. 2011. Visual and semantic similarity in imagenet. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 1777–1784.

[7] Andrea Frome, Yoram Singer, and Jitendra Malik. 2007. Image retrieval and classification using local distance functions. In *Advances in neural information processing systems (NeurIPS)*. 417–424.

[8] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV)*. IEEE, 309–316.

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1735–1742.

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

[11] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007* (2014).

[12] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 1–12.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NeurIPS)*. 1097–1105.

[14] Erik Murphy-Chutorian, Charles J Rosenberg, Nemanja Petrovic, Sergey Ioffe, and Sean O'malley. 2015. Visual content retrieval. US Patent 8,983,941.

[15] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 1717–1724.

[16] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks* 12, 1 (1999), 145–151.

[17] Sujith Ravi and Qiming Diao. 2016. Large scale distributed semi-supervised learning using streaming approximation. In *Artificial Intelligence and Statistics (AISTATS)*. 519–528.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 815–823.

[20] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[21] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1857–1865.

[22] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 843–852.

[23] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems (NeurIPS)*. 1988–1996.

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

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2818–2826.

[26] Graham W Taylor, Ian Spiro, Christoph Bregler, and Rob Fergus. 2011. Learning invariance through imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2729–2736.

[27] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. 2018. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 8769–8778.

[28] Gang Wang, Derek Hoiem, and David Forsyth. 2009. Learning image similarity from flickr groups using stochastic intersection kernel machines. In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV)*. IEEE, 428–435.

[29] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1386–1393.

[30] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. 2017. Sampling matters in deep embedding learning. In *Procceedings of the IEEE International Conference on Computer Vision (ICCV)*.