Fine-Grained Image Search

Lingxi Xie, Jingdong Wang, Bo Zhang, and Qi Tian, Senior Member, IEEE

Abstract—Large-scale image search has been attracting lots of attention from both academic and commercial fields. Conventional Bag-of-Visual-Words (BoVW) model with inverted index is verified efficient at retrieving near-duplicate images, but it is less capable of discovering fine-grained concepts in the query and returning semantically matched search results.

In this paper, we suggest that instance search should return not only near-duplicate images, but also fine-grained results, which is usually the actual intention of a user. We propose a new and interesting problem named fine-grained image search, which means that we prefer those images containing the same fine-grained concept with the query. We formulate the problem by constructing a hierarchical database and defining an evaluation method. We thereafter introduce a baseline system using fine-grained classification scores to represent and co-index images so that the semantic attributes are better incorporated in the online querying stage. Large-scale experiments reveal that promising search results are achieved with reasonable time and memory consumption. We hope this paper will be the foundation for the future work on image search. We also expect more follow-up efforts along this research topic and look forward to commercial fine-grained image search engines.

Index Terms—Fine-Grained Image Search, Problem Formulation, Evaluation, Semantic Indexing, Applications

I. INTRODUCTION

Recent years have witnessed the development of Web-scale content based image retrieval (CBIR). Based on the Bag-of-Visual-Words (BoVW) model and the inverted index structure, the state-of-the-art image search engines are capable of indexing billions of images and returning results in milliseconds. Although different approaches have been proposed to improve near-duplicate search performance, we still see few search engines which retrieve images based on the analysis of fine-grained concepts in the query.

In fact, it is a common requirement that an image search engine should return fine-grained results, i.e., those candidates containing exactly the same semantic concept as the query image. For example, when a user uploads a bird or flower photo he/she just took, retrieving those images containing exactly matched bird or flower species helps a lot to his/her understanding; a user could also benefit from the search result of a car if it contains cars of the same brand/prototype with the query image.

L. Xie and Bo Zhang are with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), and Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China (e-mail: 198808x-c@gmail.com, dcszb@mail.tsinghua.edu.cn). This work is done when Lingxi Xie is a research intern at Microsoft Research.
J. Wang is with Microsoft Research, Beijing 100080, China (e-mail: jingdw@microsoft.com).
Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

Fig. 1. Above: a typical fine-grained search query and three types of candidate images (semantically matched, semantically similar, and irrelevant). Below: the comparison between our results and those of Google/Baidu reveals that our approach might deal with fine-grained concepts in the query which is not captured by conventional search engines.

Based on the observations above, we propose a new research topic, fine-grained image search, which differs from conventional cases in that candidate images might have multiple levels of relevance to the query. We take the query shown in Figure 1 (groove billed ani, a bird species) as an example. We desire to retrieve other images also containing the same species, i.e., the groove billed ani. If we fail to find the exactly matched images, we still prefer those candidates with similar semantics, i.e., containing another bird species such as a Brewer blackbird, to the irrelevant ones containing, say, dogs, flowers or clutters.

It is worth noting that fine-grained image search is very challenging especially in a large database. As shown in Figure 1, commercial search engines such as Google and Baidu often fail in such cases, due to the lack of semantic analysis but simply parsing an image as a set of local or global features. Here, we are not claiming that our algorithm works better than Google or Baidu. Actually, the comparison is not fair since we only focus on a handful of basic-level categories, but real-world commercial search engines must deal with many other
concepts in all the Web images. We are just demonstrating the
difference between fine-grained and conventional image search
tasks, which distinguishes this work from the previous ones
not considering fine-grained concepts in the search process.

In this paper, we formulate the fine-grained image search
problem by constructing a hierarchical database, and defining
an evaluation method to judge the search quality. We propose
a baseline algorithm which uses fine-grained classification
scores to represent images’ semantic attributes, and co-indexes
images into an inverted file according to their distance in
the attribute space. An efficient online querying stage is
also designed to lookup the indexed structure in real-time.
Experimental results reveal the promising performance of our
approach. We will also release our benchmark database and
expect more research efforts to be triggered in the future.

The main contribution of this paper could be summarized
in three aspects:

- **New problem.** We formulate fine-grained image search,
  which is a new problem in the research field of multime-
dia information retrieval. To the best of our knowledge,
  this problem has not been well studied in the previous
  literatures. We expect it could become a new research di-
  rection which brings new insights and novel applications.

- **New dataset.** We construct a new database based on
  several basic image classification and object retrieval
datasets. Both fine-grained and near-duplicate concepts
  are contained, making it difficult for conventional algo-
rithms to return satisfying search results. We will make
  the database publicly available, and gradually add new
  images to enlarge the database.

- **New framework.** Based on state-of-the-art image classi-
fication and object retrieval techniques, we design a new
framework for fine-grained image search. The flowcharts
of both offline indexing and online querying are signifi-
cantly different with conventional algorithms. Our frame-
work serves as a very first trial towards this challenging
problem, and also provides a baseline performance which
is convenient to be compared with future works.

The remainder of this paper is organized as follows. The
related works are briefly reviewed in Section II. We then
formulate the fine-grained image search problem in Section III,
and propose the search framework in Section IV. After experi-
mental results and search examples are provided in Section V,
we draw the conclusions in Section VI.

II. RELATED WORKS

Our work is closely related to several popular research
topics, including, large-scale image search, fine-grained object
recognition, semantic co-indexing and semantic ontology.

A. Large-Scale Image Search

Large-scale image search is aimed at retrieving query-
relevant candidates from a large image corpus. One of the
most popular and efficient approaches is based on the Bag-of-
Visual-Words (BoVW) model and the inverted index struc-
ture [1]. Local descriptors such as SIFT [2][3] and SUR-
F [4] are extracted on the detected interest points [2][5][6].

A large visual vocabulary (codebook) is trained using the
hierarchical [7] or approximate [8] versions of K-Means,
and the descriptors are quantized onto the codebook as visual
words. Codebook training-free approaches [9][10] have also
been proposed for quantization. The inverted index is then
built as an efficient data structure representing the relationship
between images and visual words.

Naive online querying stage ranks the candidates according
to their number of feature matches to the query image.
Due to the fact that the quantization loss greatly reduces
the discriminative power of local features, the initial image
search results usually suffer from low precision and/or recall.
Various post-processing techniques are therefore designed
to improve the initial search quality, including false match
filtering [8][11], Hamming embedding [12], soft assignmen-
t on visual words [13], query expansion [14][15], query
adaptation [16], prototype extraction [17], selecting high-
quality features [18][19], contextual weighting [20][21][22],
sparsity-constrained measurement [23], distance learning [24],
visual phrases [25][26], feature similarity adaption [27], and
diffusion-based methods [28][29][30]. It is also suggested to
improve image retrieval with unlabeled training data [31].

B. Fine-Grained Object Recognition

Object recognition tasks often require capturing global prop-
erty of images, therefore the Bag-of-Visual-Words (BoVW)
model with feature pooling strategy [32] is often adopted.
Local descriptors such as SIFT [2][33] and HOG [34] are
extracted on small patches, and a codebook is trained using
K-Means [35] or GMM clustering [36]. Sparse coding [37][35] or
Fisher vector encoding [36] algorithms are used to quantize the
descriptors onto the feature space, and spatial pooling strate-
gies [38][39][40] are adopted to summarize local features into
long vectors [41]. Generalized machine learning algorithms
such as SVM are used for training and testing.

In the fine-grained classification tasks [42][43], the main
difficulty arises from the surprising inter-class similarity, there-
fore it becomes more important to uncover really useful
patches on the objects [44]. One can see many state-of-
the-art algorithms targeting at using visual attributes [45],
pose estimation [46], template matching [47], pair-wise [48]
and/or part-based [49][50] pooling models, and geometric
information [51][52] for part segmentation.

C. Semantic Co-Indexing

In the fine-grained search tasks, it is not always true that
the candidate images share a number of common visual words
with the query. In this case, conventional BoVW model with
inverted index might fail to retrieve really relevant candidates.
Consequently, it requires other higher-level clues to help
describing the visual properties of images. Intuitive solutions
come from summarizing local features into global representa-
tions [53], leveraging recognition scores [54] or referring to
privileged information [55].

Semantic co-indexing is aimed at integrating other types of
clues [56] into the inverted index structure, so that the online
querying stage could benefit from these additional information
to improve the search accuracy. It is suggested in [54] to use Hashing strategy for retrieving the approximate nearest neighbors in the attribute space efficiently, and [57] proposes to store the semantically similar pairs in the inverted index beforehand. Some authors also suggest to use feature fusion methods [58][59] for integrating multiple types of information together.

### D. Semantic Ontology

In computer science and information science, an ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse. An ontology compartmentalizes the variables needed for some set of computations and establishes the relationships between them.

The field of computer vision creates ontologies to limit complexity and organize information. Especially, ontology is widely adopted to describe the category of a namable object. For example, the ImageNet [60] database is constructed based on the WordNet project [61] and its related works [62][63]. Each leaf node of the WordNet concept tree corresponds to a fine-grained object category, such as groove-billed ani (a bird species) or golden retriever (a dog breed), and basic-level concepts correspond to the nodes with higher levels.

Ontology can also apply to problem solving [64]. Semantic ontology could be used for bridging the semantic or intent gap between query and candidates, and many researchers have been adopting this idea to define and/or improve the performance of visual information retrieval [65][66][67]. A survey on this topic could be found in [68].

### III. Problem Formulation

This section formulates fine-grained image search. Since this is a new problem which is less studied before, we first discuss the goal of the problem, and then construct a new database and suggest a hierarchical scoring function to evaluate different search engines.

#### A. Fine-Grained Image Search

Conventional image search problems usually require retrieving near-duplicate or partial-duplicate candidates, such as a landmark building (Oxford [8]), a logo (FlickrLogo-32 [69]) or a reoccurring object (UKBench [7]). In those cases, a candidate image is either relevant or irrelevant to the query. Fine-grained image search, however, queries the database with an image containing a fine-grained concept, such as groove-billed ani (a bird species) or a golden retriever (a dog breed). In this case, there might exist multiple levels of relevance between query and candidate images.

Throughout this paper, we use three levels to model the relevance between query and candidates. Two images are named semantically matched, if they contain exactly the same semantic concept, e.g., the same species of birds, the same instance of buildings, or logos of the same brand. Image are considered semantically similar if the concepts are the same only at the basic level, e.g., a groove billed ani vs. a Brewer blackbird (both are birds), or a Mona Lisa vs. a potato eater (both are paintings). Image are defined irrelevant if the concepts are not the same even at the basic level, e.g., a bird, a dog, a building and a painting are irrelevant to each other.

#### B. Database

The database for fine-grained image search contains three levels, as illustrated in Figure 2. In the first level, the database is partitioned into three subsets, i.e., fine-grained, near-duplicate, and distractor image sets. The composition of three subsets are summarized as follows:

1) **Three fine-grained concept groups:** birds (the Caltech-UCSD Bird-200-2011 dataset [42]), dogs (the Stanford Dog-120 dataset [43]), and flowers (the Oxford Flower-102 dataset [70]).

2) **Three near-duplicate instance groups:** buildings (the Oxford and Paris Buildings datasets with 22 named buildings [8]), logos (the CarLogo-51 dataset [30]), and paintings (the FamousPaint-26 dataset[^1]).

3) **Distractors.** We crawl one million irrelevant images from the Web to test the scalability of the algorithm.

![Fig. 2. The three-level hierarchical database for fine-grained image search.](image)

<table>
<thead>
<tr>
<th>Dataset Type</th>
<th>Dataset Name</th>
<th>Concept Groups</th>
<th>Training Images</th>
<th>Searching Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fine-Grained</td>
<td>Bird</td>
<td>200</td>
<td>5994</td>
<td>5794</td>
</tr>
<tr>
<td></td>
<td>Dog</td>
<td>120</td>
<td>12000</td>
<td>8580</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>102</td>
<td>2040</td>
<td>6149</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>422</td>
<td>20034</td>
<td>20523</td>
</tr>
<tr>
<td>Near-Dup.</td>
<td>Building</td>
<td>22</td>
<td>22000</td>
<td>11455</td>
</tr>
<tr>
<td></td>
<td>Logo</td>
<td>52</td>
<td>5200</td>
<td>11903</td>
</tr>
<tr>
<td></td>
<td>Point</td>
<td>26</td>
<td>2600</td>
<td>3148</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>100</td>
<td>100000</td>
<td>20506</td>
</tr>
<tr>
<td>Distractors</td>
<td>Web</td>
<td>–</td>
<td>–</td>
<td>1000000</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>0</td>
<td>33034</td>
<td>1047029</td>
</tr>
</tbody>
</table>

[^1]: This dataset is publicly available online. Type ‘FamousPaint-26 dataset’ for searching.
In all the fine-grained datasets, objects (birds, dogs and flowers) are labeled with their biology species. For example, the Bird-200 dataset contains 200 bird species and each species contains up to 60 images. All these labels, such as groove billed ani (a bird species) or golden retriever (a dog breed), could be found in the WordNet [61], a large dictionary of semantic ontologies. In all the near-duplicate datasets, however, a category is defined as a repeatable object instance, such as a building picture taken in different conditions, a registered car brand logo, or a famous paint and its copies. The name of each instance (e.g., Eiffel Tower, BMW logo or Mona Lisa paint) is not ambiguous.

We take a small portion of the database as training images. For the fine-grained sets, we use the fixed training/testing splits provided by the authors, and each training image is labeled with its fine-grained category name, such as groove billed ani, a bird species. We also randomly select some non-query images (100 per instance) from the near-duplicate subsets, and 3000 images from the distractor set for training. These images are simply labeled as not containing a fine-grained concept. All the images, except for the fine-grained training cases, compose the search database. The query set consists of all the fine-grained testing images and author-specified near-duplicate querying cases. The detailed statistics of the database is summarized in Table I.

There also exist some fine-grained datasets which are not included in the composed database, such as the Oxford Pet-37 dataset [71] (37 pet breeds, 7390 images), the Aircraft-100 dataset [72] (100 aircraft models, 10000 images), and the recently published Food-101 dataset [73] (101 food categories, 101000 images). The number of images in these unused datasets is comparable with the used datasets and less than the distractor set. As illustrated in Section IV-E, the scalability of our model makes it easy to deal with a larger number of fine-grained concepts.

### C. Evaluation

We start with defining the relevance between two images, say, A and B, which is natural in the hierarchical database: 
\[
\text{rel}(A,B) = \max\{ \text{LCA}(A,B) - 1, 0 \},
\]
where LCA(·,·) is the level index of the least common ancestor of two images. Obviously, two images are semantically matched if their relevance value is 2, semantically similar if the value is 1, and irrelevant if the value is 0.

Given a query image \( q \), we can obtain a set of \( P \) retrieved images: \( R_{q} = \{ I_{q,1}, I_{q,2}, \ldots, I_{q,P} \} \). Denote \( \text{rel}(I_{q,p}) = \text{rel}(q, I_{q,p}) \) for \( p = 1, 2, \ldots, P \), we can calculate the Discounted Cumulative Gain (DCG) [74] of \( R_{q} \) as:

\[
\text{DCG}(R_{q}) = \sum_{p=1}^{P} \frac{2^{\text{rel}(I_{q,p})} - 1}{\log_{2}(p + 1)}
\]

DCG is then normalized into nDCG by dividing it by the ideal DCG value at this query. The calculation process is illustrated in Figure 3. The nDCG values of all query images are averaged as the final score. It is proved that nDCG scores serve as a good measure for evaluating different ranking results [75].

### IV. OUR APPROACH

This section presents our approach for fine-grained image search. The overall framework is illustrated in Figure 4.

#### A. Fine-Grained Recognition

This module is aimed at training several classifiers for fine-grained concept discovery and recognition. We use the Bag-of-Visual-Words (BoVW) model with the SVM classifier.

The BoVW model consists of three stages, i.e., descriptor extraction, feature encoding and spatial pooling. We start from a raw image \( I = (a_{ij})_{W \times H} \), which is resized with the aspect ratio preserved so that the larger axis is 600 pixels. On a densely sampled set of patches, we use VLFeat [76] to extract 128-dimensional greyscale SIFT descriptors [2]. The spatial stride and window size are 8 and 16 for all images, respectively. The descriptors are reduced to 64 dimensions using PCA. Denote the set of reduced local descriptors with \( D_{a} = \{ a_{1}, a_{2}, \ldots, a_{M_{a}} \} \), where \( M_{a} \) is the total number of descriptors. To quantize the descriptors onto the feature space, we train a Gaussian Mixture Model (GMM) with 256 components. The numbers of descriptors collected for PCA and GMM training are around 2 million. The Improved Fisher Vectors (IFV) [36] are calculated for efficient feature encoding. Given a GMM with 256 components, each patch descriptor \( a_{m} \) is assigned a \( 2 \times 256 \times 64 \)-dimensional feature vector \( f_{a,m} \). Denote \( F_{a} \) as the set of encoded features: \( F_{a} = \{ f_{a,1}, f_{a,2}, \ldots, f_{a,M_{a}} \} \). We aggregate the visual phrases with sum-pooling for image-level representation: \( \tilde{x}_{a} = \sum_{m=1}^{M_{a}} f_{a,m} \). A 2-layer Spatial Pyramid Matching (SPM) [38] follows by dividing the image into 2 \( \times 2 \) subregions, and concatenating the individual pooled vectors as a long vector \( \tilde{x}_{a} \).

In practise, we also extract a set of 96-dimensional LCS descriptors [36] \( D_{0} \) on the densely sampled patches to capture color information. We follow the same flowchart to encode the descriptors into an image-level vector \( \tilde{x}_{a} \), and concatenate \( \tilde{x}_{a} \) and \( \tilde{x}_{a} \) as a single vector \( \tilde{x} \).

There also exist many advanced algorithms [47][51][52] aimed at detecting semantic object parts to capture more

![Fig. 3. The evaluation of a toy search results with two semantically matched, two semantically similar and two distractor images.](image-url)
powerful features for fine-grained recognition. Most of them are especially designed for a small number of fine-grained concepts. We do not adopt them so that our framework is capable of dealing with many other fine-grained datasets.

Based on the image-level vectors, we train two types of classification models. The first type, the fine-grained judge, is a 4-way classifier trained on 4 categories, i.e., birds, dogs, flowers and others (not containing any fine-grained concepts). We can use it to judge whether the query image contains a fine-grained concept, and which one it is. The second type is composed of three fine-grained classifiers, each trained on a fine-grained dataset. We denote the judge as $M^J$ (superscripts J for judgement), and other three classifiers as $M^C_B$, $M^C_D$ and $M^C_F$ (superscript C for classification, subscript B, D and F for birds, dogs and flowers), respectively. We can train more classification models if more fine-grained concepts are introduced. All the SVM classifiers are trained with LibLINEAR [77] using the tradeoff parameter $C = 10$.

**B. Large-Scale Feature Indexing**

Feature indexing is a basic technique for large-scale image search, which is often conducted with the Bag-of-Vision-Words (BoVW) model and the inverted index structure. The images are also resized so that the larger axis size is 600 pixels. We extract SIFT descriptors [2] from the detected Hessian Affine regions [6]. The set of descriptors is denoted as $D_c = \{d_{c,1}, d_{c,2}, \ldots, d_{c,M_c}\}$. We train a large codebook containing one million codewords with the Approximate K-Means clustering [8]. Only descriptors on training images are collected for clustering. The descriptors are then assigned onto the nearest codeword, i.e., with hard quantization strategy. The set of visual words are denoted as $V_c = \{w_{c,1}, w_{c,2}, \ldots, w_{c,M_c}\}$. An inverted index [1] is constructed for efficient lookup. We filter the stop-words in the inverted index, defined as those occurring in more than 1% images. The visual words are then weighted by the $\ell_p$-norm IDF [78].

The candidate images are first ranked by counting the weighted feature occurrences. We then construct ImageWeb [30], a graph-based structure to model the image-level relationship between images, and refine the initial search results with affinity propagation on the graph. There also exist many complicated object retrieval systems producing very high accuracy [11][15]. However these methods could be time consuming at the online querying stage, therefore are not adopted in practise.

**C. Semantic-Aware Co-Indexing**

Semantic-aware co-indexing is aimed at incorporating semantic property of images into the inverted index, so that it is possible to lookup high-level semantic attributes at the online querying stage. We follow [57] to calculate semantic attributes from fine-grained object recognition scores, and perform two successive operations, i.e., semantic-isolated image deletion and semantic-nearest image insertion, to modify and expand the indexed image lists in the inverted file. The co-indexed structure is illustrated in Figure 5.

Semantic attributes are those properties that help to describe semantic concepts of an image [79]. We use the fine-grained classifiers $M^C_B, M^C_D$ and $M^C_F$ trained in Section IV-A to calculate the classification scores for each image. The output of $M^C_B$, for example, is a 200-dimensional vector $s_B$, in which the element $s_{B,k}$ stands for the confident score of an image containing the $k$-th bird species. We normalize the scores using the soft-max function: $\tilde{s}_{B,k} = \frac{\exp(s_{B,k})}{\sum_k \exp(s_{B,k})}$, so that we have $\sum_k \tilde{s}_{B,k} = 1$. The normalized vectors, $\bar{s}_B, \bar{s}_D$ and $\bar{s}_F$, are concatenated as the semantic attribute vector $\bar{s}$ with $200 + 120 + 102 = 422$ dimensions. The distance between two images $I_a$ and $I_b$ is then measured by the Total Variance.
Distance (TVD) score [57] between their attribute vectors:
\[
\text{TVD}(I_a, I_b) = \text{TVD}(\tilde{s}_a, \tilde{s}_b) = \sum_k |\tilde{s}_{a,k} - \tilde{s}_{b,k}|
\]  

After the TVD value is calculated for each pair of images, we modify the inverted index structure by adopting two steps successively, \text{i.e.}, semantic-isolated image deletion and semantic-nearest image insertion [57]. The semantic-isolated image deletion step traverses each entry in the inverted index and check the followed images. For those entries with no less than 3 images, it removes those images which are isolated from others, \text{i.e.}, the minimum TVD from this image and others is larger than a threshold \(\rho\). This process can effectively reduce the index size without impacting the retrieval precision in the search process. The semantic-nearest image insertion step, in opposite, adds images with large semantic similarity into the inverted index. It works by searching for a fixed number of nearest neighbors of each image according to the TVD scores, and add those nearest neighbors into the image’s entry if they are not in the visual word’s entry. In essential, this process could be considered as a query expansion operation based on the reliable candidate images.

It is obviously insufficient to represent an image’s semantic attribute with only three fine-grained datasets (422 categories), but this serves as a baseline method to incorporate fine-grained information into the inverted index structure. As more fine-grained datasets, such as \textit{cats} [71] or \textit{aircrafts} [72], are introduced into the search task, we can extract longer attribute vectors to capture richer semantic properties.

\textbf{D. Online Querying}

When a query image comes, we first follow the previous sections to extract all the required features, \text{i.e.}, an image-level representation vector, a set of visual words, and a semantic attribute vector. Then the flowchart of online querying is illustrated in the lower part of Figure 4. We use the 4-way classifier \(\mathcal{M}^J\) to judge if a fine-grained concept, \text{i.e.}, a \textit{bird}, a \textit{dog}, or a \textit{flower}, is contained in the query image, or if the query does not contain any of these concepts. We then search the database according to the judgement.

If the prediction says that the query image contains a fine-grained concept, say, a \textit{bird}, then we can calculate the 200-dimensional \textit{bird} attribute vector with model \(\mathcal{M}^C_B\), compare it to the images in the database, and sort them according to the TVD value \(\text{TVD}(\tilde{s}_q, I_b)\). Since we can index as many as one million images, calculating the TVD scores one-by-one can be very computationally expensive. We adopt the Locality-Sensitive Hashing (LSH) method [80] to index images into a small number of hashing tables and search only for those images sharing common bins with the query. In practise, LSH works much faster than brute force search and produces satisfying search results.

If the prediction suggests that the query image does not contain a fine-grained concept, we can adopt the semantic-aware online querying algorithm [57] to lookup the inverted index. Here, the search process is very similar to the general case of near-duplicate image search, but we make full use of the co-indexed structure to improve the quality of retrieved images. In a word, we use the samples found in the semantic-nearest image insertion step to update the matching score of each candidate. This process is equivalent to expanding the query in the attribute space using the top-ranked candidates, and will be surprisingly effective when the query image turns out to contain a fine-grained concept (see the lower part of Figure 7).
It is worth noting that we have made an early decision at the beginning of the querying stage. Although it limits the model flexibility and makes the search result highly unstable when the judgement is incorrect (see Figure 7), the online querying time is significantly reduced. We expect more advanced approaches to refine this process in the future research.

E. Scalability Issues

Currently, our algorithm only considers a handful of basic-level categories, i.e., birds, dogs and flowers. Here we discuss the scalability issues when we a larger number of visual concepts are introduced into the database.

When the number of basic-level categories increases, we first need to calculate a larger number of semantic attributes and store them in the inverted index. Let us assume that we have 100 basic-level categories and each of them has 200 fine-grained classes, which leads to a total number of 20000 categories, comparable with the scale of ImageNet [60]. In such cases, extracting all the 100 + 20000 attributes might be computationally expensive. We can adopt an approximate strategy, which first calculate 100 basic-level semantic attributes, then pick up the top-5 categories with the largest attribute scores, and calculate the fine-grained semantic attributes only for these 5 categories (5 × 200 fine-grained classes). For the remaining basic-level categories, we can simply set the same score for all the 200 fine-grained classes.

The above method significantly reduces the time/memory costs in the offline indexing process. The only information loss happens when the actual basic-level category is not contained in the top-5 recognition results. According to the performance of our classification algorithm in the Caltech101 dataset [81], which contains 101 basic-level classes (not including the background category) and the number of training images is much smaller (30 per category), the top-5 classification accuracy over 101 basic-level categories is often higher than 90%. Therefore, the time/memory costs could be reduced to around 5% with less than 10% information loss.

At the online querying stage, the difference comes from seeking the approximate nearest neighbor in the semantic attribute space. Since the expanded space might have tens of thousands of dimensions, even LSH requires heavy computation. To cope with, we could adopt another approximate strategy, which considers the semantic attribute space of each basic-level category individually. When the basic-level classification scores of the query image are calculated, we find the top-5 categories and retrieve the top-1000 nearest neighbors from each of these 5 fine-grained attribute spaces. Finally, the retrieved images are combined into a single list according to their total distance to query in these 5 spaces. In this way we prevent our algorithm from calculating a large number of fine-grained concepts and becoming computationally expensive.

We will provide a more detailed scalability analysis in Section V-D. Meanwhile, we admit that many other approaches might be adopted to deal with this problem, such as adopting a hierarchical structure and a top-down approach to categorize an image from-coarse-to-fine. We aim at suggesting a possible solution to this new and challenging problem, and inspiring more research efforts in the future.

F. Generalization Issues

Of course, three levels of relevance are not enough for describing the similarity between two images. A more generalized approach is to use the natural structure of ontologies [61] to calculate a multi-level similarity score. For example, [54] uses hierarchical semantic indexing to improve the image retrieval performance, and [82] learns a fine-grained similarity with deep learning techniques. Both the above works inspire us to formulate fine-grained image search on a larger number of relevance levels.

Our two-stage framework could be easily modified to fit the new problem setting. We inherit the module of basic-level concept discovery, but add the hierarchical loss function into fine-grained image classification process. By hierarchical loss we mean that the classification score of an image is no longer either 1 (correct) or 0 (incorrect), but could be a fractional value between 0 and 1 indicating the relevance of the predicted label and the ground-truth label. Such technique, which is widely adopted in large-scale image classification, penalizes heavier on those basic-level errors (e.g., a bird is classified as a building) than those fine-grained errors (e.g., a red-winged blackbird is classified as a brewer blackbird).

V. Experimental Results

A. Recognition and Retrieval Accuracy

First, we report the accuracy on fine-grained concept judgement and object recognition. The results are listed in Table II and III, respectively. One can see the satisfied performance when judging the fine-grained concept in an image: 96.20% birds, 97.98% dogs, 96.62% flowers and 98.59% others (not containing a fine-grained concept) queries are correctly classified. The fine-grained recognition accuracy is also nearly the state-of-the-art without using human annotation and complicated part detection models [48][51][52]. It guarantees that in most cases, the users’ intention could be correctly understood by the search engine.

Next, we report the object retrieval accuracy on near-duplicate datasets with one million distractors in Table IV. Once again, our search accuracy is nearly the state-of-the-art,
given that we need to finish the search within 100ms thus some complicated post-processing approaches [11][15] are not used.

The good performance of separate modules helps us produce satisfying fine-grained search results. Please note that we are not aimed at comparing our classification or retrieval modules with the state-of-the-art algorithms. Most often, those complicated methods use some specified clues or tricks to boost the accuracy on one or two datasets. These algorithm are either too specific thus not generalizable onto a wide range of visual concepts, or very computationally expensive so that are not applicable onto large-scale image search.

<table>
<thead>
<tr>
<th></th>
<th>Building</th>
<th>Logo</th>
<th>Painting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our mAP score</td>
<td>0.7631</td>
<td>0.4842</td>
<td>0.5857</td>
</tr>
<tr>
<td>Best known, t ≤ 100ms</td>
<td>≈ 0.78</td>
<td>≈ 0.49</td>
<td>≈ 0.60</td>
</tr>
<tr>
<td>Best known</td>
<td>≈ 0.87</td>
<td>≈ 0.53</td>
<td>≈ 0.66</td>
</tr>
</tbody>
</table>

TABLE IV
THE NEAR-DUPLICATE SEARCH ACCURACY (WITH ONE MILLION DISTRACTORS) OF BUILDING, LOGO AND PAINTING QUERIES.

<table>
<thead>
<tr>
<th></th>
<th>Model-SA</th>
<th>Model-VW</th>
<th>Model-COM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird-200</td>
<td>0.6741</td>
<td>0.3215</td>
<td>0.6169</td>
</tr>
<tr>
<td>Dog-120</td>
<td>0.7102</td>
<td>0.3692</td>
<td>0.6727</td>
</tr>
<tr>
<td>Flower-102</td>
<td>0.7061</td>
<td>0.4250</td>
<td>0.7302</td>
</tr>
<tr>
<td>Building</td>
<td>0.5601</td>
<td>0.9205</td>
<td>0.8887</td>
</tr>
<tr>
<td>Logo</td>
<td>0.3406</td>
<td>0.5940</td>
<td>0.5432</td>
</tr>
<tr>
<td>Painting</td>
<td>0.4091</td>
<td>0.6703</td>
<td>0.6610</td>
</tr>
<tr>
<td>Fine-Grained</td>
<td>0.7268</td>
<td>0.3712</td>
<td>0.6733</td>
</tr>
<tr>
<td>Near-Duplicate</td>
<td>0.4366</td>
<td>0.7283</td>
<td>0.6976</td>
</tr>
<tr>
<td>Overall</td>
<td>0.5817</td>
<td>0.5501</td>
<td>0.6855</td>
</tr>
</tbody>
</table>

TABLE V
THE SEARCH PERFORMANCE OF THREE DIFFERENT MODELS.

The results are summarized in Table V. One can observe that on each separate query set, either fine-grained or near-duplicate, Model-COM does not produce the best performance among all three models. Figure 6 provides an intuitive comparison on both types of queries. When the query image contains a fine-grained concept, Model-SA better capture the image’s global property; however when the query is a near-duplicate instance, Model-VW produces higher accuracy. Model-COM becomes the best choice only when the scores are averaged, i.e., there might be both types of queries, as in the real-world applications. This suggests that analysing the search intention is probably more important than achieving higher accuracy in separate tasks.

Once again, we shall emphasize that this paper is aimed at proposing a new problem and suggesting a new framework. Although we do not use those complicated classification/retrieval algorithm in practise, one might notice that the proposed framework is highly modular, implying that it is easy to replace each module with a more efficient algorithm designed in the future. We believe that baseline accuracy of Model-SA and Model-VW would be thereafter improved. Since the combined model (Model-COM) absorbs the advantages of both Model-SA and Model-VW, we can forecast that better search performance will also be produced by Model-COM in such cases.

C. Time and Memory Costs

We report the time and memory costs at the online querying stage with about one million images.

We need to store two parts of information (see Figure 5). The first part, image ID for each occurrence of visual words, requires 4 bytes for each image-word pair. The average number of descriptors on each image is about 1000, thus we need 4 Kilobytes for a single image. According to [57], the memory consumption increases by about one half after semantic-aware co-indexing. The second part is the semantic attributes calculated from the fine-grained classification scores. For each image, we have to store 4 + 200 + 120 + 102 = 426 floating numbers in total, requiring about 2 Kilobytes. Therefore, the total memory overhead is less than 4 × 150% + 2 = 8 Kilobytes for one image, and about 8 Gigabytes for the whole database with one million images. ImageWeb [30] requires only about 160 Megabytes for one million images.

At the online querying stage, the time cost consists of three parts, i.e., descriptor extraction and quantization, concept judgement, and image search, which follows either fine-grained (using semantic attributes) or near-duplicate (using co-indexed file) flowchart, depending on the concept judgement result. The SIFT descriptor extraction takes 1000ms (650ms for dense sampling, 350ms for interest point detection and description), quantization takes 400ms (300ms for Fisher vectors, and 100ms for hard quantization). The concept judgement stage takes about 50ms. If the query is considered as containing a fine-grained concept, then the fine-grained recognition takes about 100ms, and LSH takes about 400ms; otherwise the near-duplicate searching in the semantically co-indexed file takes about 550ms. It takes less than 100ms for ImageWeb to post-process the search results. Overall, the total time required for a single query would not exceed 2 seconds. All the time is recorded on a single 3.0GHz CPU.

D. Scalability Analysis

Let us still assume that there are 100 basic-level categories and each category has 200 fine-grained concepts. For each image, we first need to store the IDs and scores of the top-5 basic-level recognition results. Next, 5 × 200 fine-grained semantic attributes are calculated. In total, we need...
We present some representative queries and corresponding search results in Figure 7. One can see that, our approach works well for those correctly judged queries, suggesting that understanding the users’ intention is crucial to improve the search quality. Meanwhile, we also present four wrongly judged cases, two fine-grained and two near-duplicate. For the first two cases, we can still obtain pretty good search results thanks to the semantic-aware co-indexing, however the latter cases show dramatically unstable results due to the ignorance of local features which are key to near-duplicate retrieval. This once again suggests to improve the concept judgement accuracy in the future research.

VI. Conclusions

This paper proposes fine-grained image search, a challenging topic which is less studied in the multimedia community. We argue that it is a common requirement that instance search returns not only near-duplicate but also fine-grained results. We formulate the fine-grained image search problem by constructing a new database and defining an evaluation method. A baseline framework is also proposed to incorporate semantic attributes into the inverted index, leading to an efficient search engine which produces promising search results in large-scale experiments. Since our framework is highly modular, it might cooperate well with many other classification/retrieval algorithms. The scalability of our algorithm also makes it easy to be transplanted onto commercial search engines.

In the future, we shall release the database used in the experiments, and gradually add more images, such as those fine-grained sets in ImageNet [60], into the database. The methods discussed in Section IV-E and IV-F will be implemented for scaling up and generalization. We hope this topic will reveal
Fig. 7. Representative queries and search results. Above the dashed line are the queries correctly classified in the fine-grained concept judgement. Below, however, the two birds, one logo and one painting images are wrongly recognized as: not containing a fine-grained concept, a flower, a bird, and a flower, respectively. The text below each image indicates its ground-truth label. One can see that, our system produces satisfied results when the users’ intention is correctly understood. Moreover, when a fine-grained query is misclassified as a near-duplicate query, we can also return fair quality search results thanks to the semantic-aware co-indexing; however the search accuracy will become dramatically unstable when a near-duplicate query is misclassified.
an exciting research direction which brings new insights and novel applications. We also expect more research efforts to be triggered on this interesting yet less explored problem.

ACKNOWLEDGMENTS
This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403, 2012CB316301, and 2014CB347600), the National Natural Science Foundation of China (Grant Nos. 61332007, 61273023 and 61429201), and the Tsinghua University Initiative Scientific Research Program (Grant No. 20121088071). This work was also supported in part to Dr. Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America.

REFERENCES


Lingxi Xie received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2010. He is currently a Ph.D. candidate at the same university. He was a research intern in Microsoft Research Asia from August 2013 to January 2014, and from September 2014 to present. He was a visiting researcher at the Department of Computer Science at the University of Texas at San Antonio (UTSA) from February to July 2014. His research interests include computer vision, multimedia information retrieval and machine learning.

Jingdong Wang received the B.Sc. and M.Sc. degrees in the Department of Automation from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in the Department of Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is currently a Lead Researcher at the Visual Computing Group, Microsoft Research. His areas of interest include computer vision, machine learning, and multimedia search. At present, he is mainly working on the Big Media project, including large-scale indexing and clustering, and Web image search and mining. He is an editorial board member of Multimedia Tools and Applications.

Bo Zhang received the B.E. degree from the Department of Automatic Control, Tsinghua University, China, in 1958. From 1960 to 1982, he visited University of Illinois at Urbana-Champaign, USA, as a scholar. He is currently a Professor in the Department of Computer Science and Technology at Tsinghua University. His research interests include artificial intelligence, machine learning, pattern recognition, knowledge engineering, intelligent robotics and intelligent control. He won the Prize of European Artificial Intelligence in 1984. He won the 1st-class and Science and Technology Progress Prize three times in 1987, 1993 and 1998, and the 3rd-class Prize two times in 1995 and 1999, respectively. He is a member of Chinese Academy of Sciences.

Qi Tian (M96-SM03) received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University in 1996 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign in 2002. He is currently a Professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA). He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008-2009. His research interests include multimedia information retrieval and computer vision.