Photo Recall: Using the Internet to Label Your Photos

Neeraj Kumar Steve Seitz
Computer Science and Engineering
University of Washington
Seattle, WA, USA
{*neeraj,seitz*}*@cs.washington.edu*

***Abstract*—We describe a system for searching your personal photos using an extremely wide range of text queries, including dates and holidays (*Halloween*), named and categorical places (*Empire State Building* or *park*), events and occasions (*Radiohead concert* or *wedding*), activities (*skiing*), object categories (*whales*), attributes (*outdoors*), and object instances (*Mona Lisa*), and any combination of these – all with no manual labeling required. We accomplish this by correlating information in your photos – the timestamps, GPS locations, and image pixels – to information on the Internet. This includes matching dates to holidays listed on Wikipedia, GPS coordinates to places listed on Wikimapia, places and dates to find named events using Google, and visual categories and object instances using classifiers either pre-trained on ImageNet or trained on-the-fly using results from Google Image Search. We tie all of these disparate sources of information together in a unified way, allowing for fast and accurate searches using whatever information you remember about a photo. We quantitatively evaluate several aspects of our system and show excellent performance in all respects. Please watch a video demonstrating our system in action on a large range of queries at http://youtu.be/Se3bemzhAiY**

***Keywords*-photo organization; image search; content-based image retrieval; gps; visual classifiers; natural language search; layered graph inference; events**

I. Introduction

We've all had the frustrating experience of trying – unsuccessfully – to find photos of a particular event or experience. With typical personal photo collections numbering in the tens of thousands, it's like finding a needle in a haystack. Current tools like Facebook, Picasa, and iPhoto only provide rudimentary search capabilities, and that only after a tedious manual labeling process. In contrast, you can type in just about anything you want on Google Image Search (for instance), and it will retrieve relevant photos. Why should searching your personal photos be any different? The challenge is the lack of descriptive text for indexing; people generally label very few of their photos.

The key insight in this paper is that a surprisingly broad range of personal photo search queries are enabled by **correlating information in your photos to information on the Internet**. For starters, we can find your photos from *Christmas* (Fig. 1a) by using lists of holidays and dates, or of *Hawaii* by analyzing GPS (aka *geotags*) and matching to online mapping databases. We introduce an extremely powerful version of location search that enables queries ranging from exact place names (*FAO Schwartz*, *Grand Canyon* [Fig. 1b]) to rough recollections (*park*, *skiing* [Fig. 1c]). These capabilities alone are very powerful and go beyond what's possible in leading photo tools like Facebook.

More interestingly, there's a broad class of important queries that are not expressed in terms of time or location, but which can be answered *using* photo time and location information, in conjunction with online data sources. For example, suppose you want to find the photos you took of the *Radiohead* concert (Fig. 1d). This query doesn't specify a location or a date; so to answer it, we find all your photos that are taken near performance venues (*e.g.*, stadiums, concert halls, arenas, major parks), search Google for events that occurred at those places on the dates when you took your photos (using a query like, "Key Arena, Seattle, April 9, 2012"), parse the results page (on which many mentions of "Radiohead" occur), and associate the resulting text to the corresponding photos in your collection. This enables searching for a wide range of events you've seen, like *Knicks game*, *Cirque du Soleil*, and *Obama's inauguration*. All of this is transparent to the user; they simply issue the query *Knicks game* and we figure out how to answer it.

An even broader range of queries is enabled by analyzing the pixels in photos and correlating them to other photos on the Internet. For example, when you type in *Mona Lisa* (Fig. 1f), we do a Google Image search for "Mona Lisa," download the resulting images, match them to your photos using interest points, and return the results – all in a few seconds. Whereas this is an example of a specific instance, we also support category-level queries using both pretrained and on-the-fly trained classifiers. For example, to find your *wedding* photos, we do a Google Image search for "wedding," download the results, train a visual classifier, and run the classifier on your photos – again, in just seconds. In summary, we support an extremely wide range of queries:

- **dates and holidays**: *August 2012*, *Thanksgiving*
- **named places**: *Grand Canyon*, *Sea World*, *FAO Schwartz*
- **categorical places**: *zoo*, *hotel*, *beach*
- **activities**: *skiing*, *cricket*, *paintball*
- **named events**: *Radiohead concert*, *Knicks game*
- **events by type**: *wedding*, *birthday*, *graduation*
- **categories of things**: *whales*, *green dress*, *convertible*

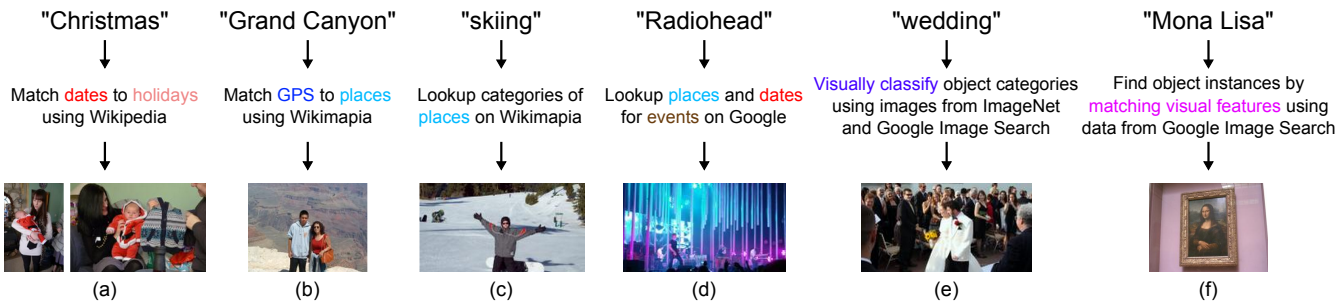


Figure 1: Our system allows users to search their personal photos using queries like (a) "Christmas" or other **holidays**, (b) "Grand Canyon" or other **places**, (c) "skiing" or other **activities**, (d) "Radiohead" or other **events**, (e) "wedding" or other **visual categories**, (f) "Mona Lisa" or other **object instances**, and arbitrary combinations of these – all with **no manual labeling.** We associate images with labels by correlating information in your photos to information online using a variety of techniques, ranging from computer vision, to GPS and map databases, to on-the-fly internet search and machine learning.

- **attributes**: *portrait*, *black-and-white*, *blurry*
- **instances**: *Mona Lisa*, *Eiffel Tower*, *Mickey Mouse*

Furthermore, these types of queries can be combined, *e.g.*, *wedding in New York*, to provide even richer queries and more specific results. The use of complementary types of data also greatly improves both the robustness and flexibility of our system; the former because uncertain estimates of one modality can be compensated for by others, and the latter because there are often many possible ways to arrive at the same image, allowing the user to search by whatever pieces of information she remembers about the image. Fundamentally, the use of Internet data enables an enormous shift in user experience, where *the user chooses the search terms* rather than these being limited to a predefined set of options (as is the norm for virtually all prior work in CBIR and object recognition), or requiring manual labeling. More specifically, ours is the first published work to include the following new capabilities, without requiring any labeling:

- Named event personal photo search (*e.g.*, *Knicks game*).
- Visual category search for a wide range of user-defined queries, by extending the on-the-fly-training work of [1] (proposed previously for faces only) to general queries.
- Object instance search in its full generality, *i.e.*, matching your photos to arbitrary named objects on the Internet.
- Far more extensive location-based query support (including named and categorical places at all levels of granularity) than any other work, by leveraging Wikimapia.

We represent all information in our system as a hierarchical knowledge graph, which provides a unified representation of all data and lets us efficiently perform inference operations via propagations through the graph, including search, auto-complete, and query-dependent description of matched images. Finally, we quantitatively evaluate the key aspects of our system by testing search performance on manually labeled images. Specifically, we test our coverage of places (both named and categorical), our ability to find named events, and our computer vision-based visual category classifiers. For a qualitative look at many more examples of search results, please see our supplementary video: http://youtu.be/Se3bemzhAiY

II. Related Work

Our work is inspired by Google Image Search and other Internet search engines aimed at producing relevant content for *any* user-specified query. We seek to provide similar functionality in the domain of personal photos. But while Internet image search engines exploit co-occurances of images and text on web pages, most personal photo collections have scarce textual information to use as a ranking signal; hence the latter domain is much more challenging.

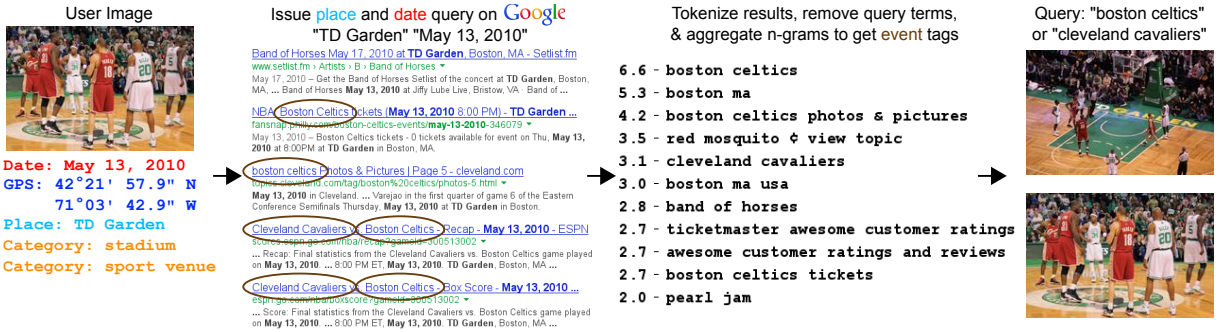
In contrast, consumer photo organization tools like Picasa and Facebook provide only rudimentary search capabilities, almost completely based on manual labeling. Since most users label few if any of their photos, search is largely ineffective. In their latest release, iPhoto introduced the ability to search for place names by "matching terms such as *Seattle* or *Milan*, to a mapping database."[1] While no other technical details have been published, the feature seems to provide similar capabilities to our system with regard to matching place names, but not place categories. In June 2013, Google enabled a new auto-labeling feature that leverages deep learning to classify users' photos using $2,000$ pre-trained visual classifiers (again, few technical details are provided).[2] However, none of these systems support searches for named events (*e.g.*, *Burning Man*), on-the-fly training of arbitrary visual classifiers (*e.g.*, *green dress*), or matching object instances (*e.g.*, *The Last Supper*). While researchers have explored the use of manual tags, we focus here on purely automated approaches.

In the research community, there is a large body of work on content-based image retrieval (CBIR). See Datta et al. [2] for a survey of this field. While much of this literature involves novel browsing interfaces or visual search techniques (*e.g.*, query-by-example, similar images), we focus specifically on related work aimed at *text-based search* in particular, which requires indexing based on semantic

[1] quoted from http://support.apple.com/kb/PH2381
[2] http://goo.gl/xtV5mc

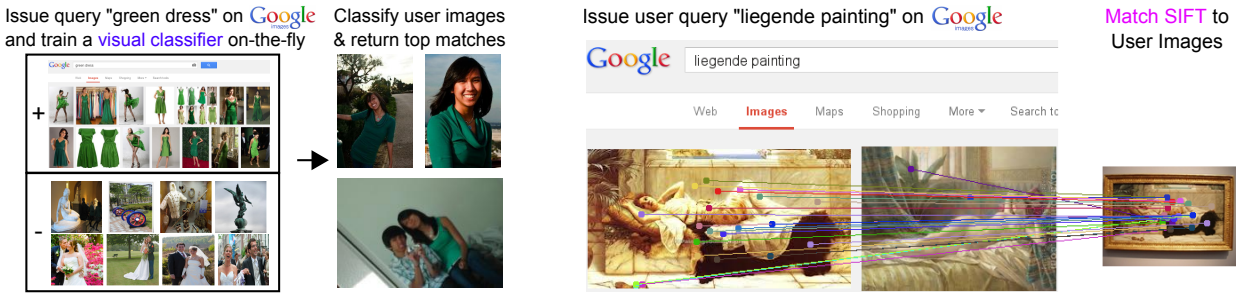
(a) Indexing holidays using Wikipedia

(b) Indexing place names and categories using Wikimapia



(c) Indexing events using Google

(d) Indexing visual categories using classifiers trained on ImageNet



(e) On-the-fly visual classification using Google Images

(f) On-the-fly visual SIFT matching using Google Images

Figure 2: Our system associates images with labels by matching different types of image data to various online sources, either in an initial indexing step (a-d), or on-the-fly when the user issues a query (e-f). (a) We match the datestamps stored in photos to a list of holidays from Wikipedia, allowing for queries like *"Saint Patrick's Day"*. (b) We lookup GPS coordinates from photo metadata on Wikimapia to get place names and categories, allowing for searches like *"FAO Schwarz"* or *"toy shop"*. (c) We issue searches on Google for pairs of {date, place name} to find what event took place there. We parse the results and accumulate n-grams to get event tags, like *"boston celtics"*. (d) We pretrain thousands of binary visual classifiers using categories from ImageNet, such as *"wedding"* or *"whale"*. (e) For things not covered in ImageNet, we issue queries on Google Images and train a binary visual classifier on-the-fly, such as *"green dress"*. (f) For finding object instances, we can also match SIFT descriptors on-the-fly from Google Image search results, such as for a photo of the *"Liegende"* painting. Despite several sources of noise in the data and matching processes, we are able to return accurate results.

content, and discuss the most relevant techniques. Naaman [3] first proposed using a database of named geographic locations to automatically label geo-tagged photos, although their capabilities were limited to cities, states, and parks, and they did not support search. More recent work has looked at combining personal tags with community tags, using GPS-tagged photos on flickr to find nearby tags [4], [5]. Many authors have used date and time information to organize photo collections and group them into events, *e.g.*, [6]. Image content has also been used to improve event clustering [7]. There is also work specific to finding faces in photos [8]. For assigning labels to images based on visual content, existing work assumes a labeled dataset is accessible beforehand to learn image annotation models using sophisticated optimization methods [9], [10]. In contrast, we use standard off-the-shelf vision algorithms applied to results from Google Images, allowing for more generality.

We are also inspired by, and strongly leverage, the recent progress in the computer vision community on object recognition. Modern techniques, mostly based on low-level features and discriminative classifiers [11], [12], [13], [14], perform increasingly well on computer vision bechmarks such as the Pascal Challenge [15] and ImageNet [16]. They have not been evaluated, however, in the context of unconstrained image search, where the user can type in *any* query term, although some have tried learning object classifiers for relatively "clean" images using Google Image search – but at far slower speeds [17]. Others have looked at using text from the web to learn better classifiers [18]. We present the first system that uses visual classifiers for unconstrained personal photo search, and show it to be remarkably effective, especially when combined with time and place cues.

Our technique of training classifiers on-the-fly adapts the recent work of Parkhi et al. [1] to the domain of personal photo collections. They also train classifiers based on Google Image Search results, but focus specifically on video footage, and finding celebrity faces. Others have also started exploiting Google Images for training classifiers directly, *e.g.* [19].

III. Data Sources

Our personal photo search system takes text queries as input and returns a ranked list of matching images; this requires associating text labels with images. In most existing systems, users must assign these labels directly to the photos, a manual and time-consuming process. To avoid this tedium, we find existing sources of text labels and associate these to the right photos. One of the keys to enabling the wide variety of queries we support is that we make much more extensive use of not just the image pixels, but also the image timestamps and GPS coordinates. To make these raw sensor values useful, we connect them to labels through the use of existing online data sources. Figure 2 visually summarizes all of our data sources and indexing methods.

A. Holidays

Time is one of the most important qualities about a photo. When you think of a particular event or memory from your life, you probably remember *when* it happened: yesterday, last year, during Halloween, *etc*. To match such holidays, we use the datestamp stored in each of the user's photos and see if they occur on or around any of the dates (within ± 2 days) listed in the article, "Public holidays in the United States." This allows for searches like *Christmas* (Fig. 1a) and *Saint Patrick's Day* (Fig. 2a).

B. Places

Another primary way of describing photos is by *where* they happened. As with time, we have to work our way up from the raw sensor data recorded in the image metadata, in this case the GPS coordinates of a photo (*e.g.*, $51^\circ\ 30'\ 2.2''$ N, $0^\circ\ 7'\ 28.6''$ W), to a description like, "the Big Ben clock tower in the Westminster area of London, England." We use Wikimapia[3], an online crowd-sourced database which focuses on geographic information and currently contains data on 20 million places (and growing rapidly). Each place includes the place name, a text description, and a list of categories describing its type and activities performed there (such as *park*, *hotel*, or *paintball*). We store the list of places around each photo's GPS location. This allows the user to search using queries such as *FAO Schwarz* or *toy shop* (Fig. 2b), *Grand Canyon* (Fig. 1b), *skiing* (Fig. 1c), *etc*.

C. Events

Putting time and place together yields *events* – the natural way by which people tend to group many of their photos. Several of our most cherished memories come from shared public events like concerts, sports, cultural activities, and business conferences. There is a record of most such public events on the Internet, whether in the form of large domain-specific databases such as last.fm for music and espn.com for sports, global aggregators of events like ticketmaster.com, or individual websites for specific events, such as london2012.com for the 2012 Olympics (held in London). Fortunately, people don't have to know what the relevant sites are for a particular event, because all of these websites – and billions more – are indexed by search engines such as Google. While searching for the name of an event will obviously bring up relevant websites (with information such as the date and venue of the event), *the reverse is also true*: searching for the date and venue on Google returns results containing the name of the event and often other relevant information. This additional information includes, for example, the names of performing artists at concerts, operas, and dances; and the names of participating teams at sporting events. Figure 2c shows an example for a basketball game: a search for *"TD Garden" "May 13, 2010"* (the name of the venue and the date

[3]http://wikimapia.org

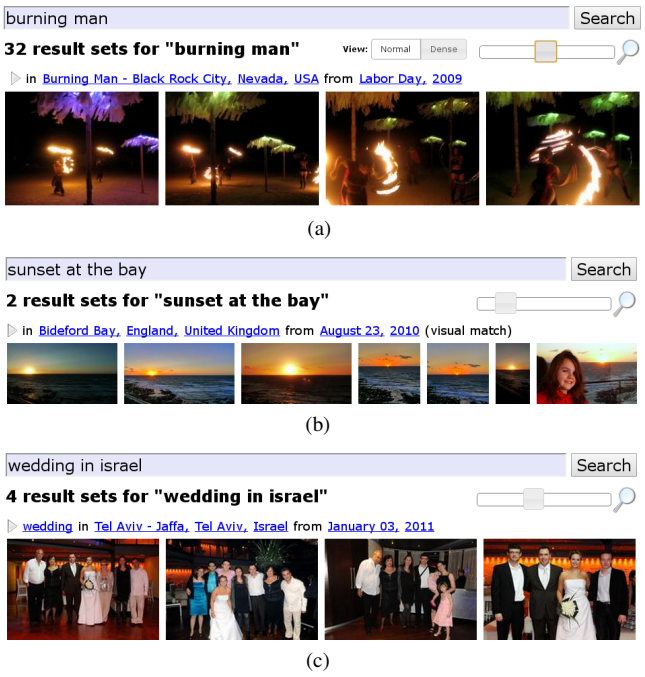


Figure 3: Various search results using our system.

of the event) brings up a page of results, many of which contain the terms *Boston Celtics* and *Cleveland Cavaliers*, the two teams playing that night. Note that we are talking about text on the Google result page itself – we do not need to follow any of the result links to their respective webpages.

We exploit this to automatically label events in a user's photos. First, from the timestamp stored in each image, we know the date the photo was taken on. Second, as described in the previous section, we have already mapped photos' GPS locations to place names via Wikimapia. Third, we also know the categories of each place returned from Wikimapia. We thus select those places that are likely to be venues for events – stadiums, theatres, parks, *etc*.– and find all photos taken at those places. Our system then submits queries on Google in the form "<*venue name*>" "<*date*>". We parse the text on the Google result page and accumulate the most commonly occurring n-grams across all results. We weight each instance by the length of the n-gram and the rank of the result it was found in. This gives us a ranked and scored list of candidate terms for each event, as shown in Fig. 2c (2nd from right). We associate the top 10 terms from this list to the image. Examples of events we can index in this way include sports, festivals such as *Burning Man* (Fig. 3a), concerts, parades, and many more.

D. Visual Categories and Object Instances

Many other photos you'd like to be able to find – from your sister's wedding, to portraits of your daughter, to the exotic flowers you saw in Brazil – correspond to visually distinctive categories. For example, wedding photos tend to have formal dresses, veils, flowers, and churches. Many of these characteristics are amenable to classification by modern computer vision techniques. We follow the standard supervised learning pipeline: features extracted from labeled images are used as positive and negative examples to train binary classifiers. This approach requires a source of labeled examples: images of the category we wish to learn as positive training data, and images of other categories as negative data.

Our first data source is ImageNet [16], which currently has 14 million images for $21,841$ different categories, the latter of which are organized into a hierarchy. Some of these categories are quite useful for our task, such as *wedding* or *whale* (Fig. 2d). Hence, we pre-trained linear SVM classifiers for nearly a fourth of ImageNet – $4,766$ categories (synsets) – using images of the synset as positive examples and images of other synsets (excluding ancestors or descendents) as negatives. Our features are histograms of color, gradients, and gist [20]. However, ImageNet is still missing many useful categories, from random omissions (like *fireworks*, or *graduation ceremony*), to instances of particular classes (*e.g.*, *2007 Toyota Camry* or *iPhone*), and it also does not cover adjectives or verbs.

Fortunately, there exists a much larger set of images covering all the types of queries users might want to do today or in the future: the Internet, as indexed by Google Image Search. By taking advantage of the rich structure of HTML webpages, links between pages, and text surrounding images, Google Image Search can return relevant image results for an extraordinary range of queries. We can piggyback on the work they have done by using their results as a source of labeled data, albeit a noisy one. When a user performs a search on *our* system, we issue the same query on Google Image Search, immediately. We then download their top results, and run the entire classifier training and evaluation pipeline, as described above (with some small tweaks for speed); the entire process takes under 10 seconds on a single machine in our prototype system. Note that our system uses the entire images for training; this means we can classify scenes and objects which cover most of the image but can't detect or localize small objects. Through Google, we can cover an extremely wide range of queries, including very specific visual categories like green dress (Fig. 2e), combinations of places and categories, like *wedding in Israel* (Fig. 3c); photos exhibiting attributes like *portraits*; *etc*.

Finally, if you're searching for a specific instance of an object (*e.g.*, *Mona Lisa* painting), we don't even need to train a classifier; instead, we simply match distinctive local features from a labeled image (using Google Image Search results) to those in your images and return the images which have the most consistent such feature matches [12]. Examples of queries we can support using this technique (again, in only a few seconds) include Figs. 1f and 2f.

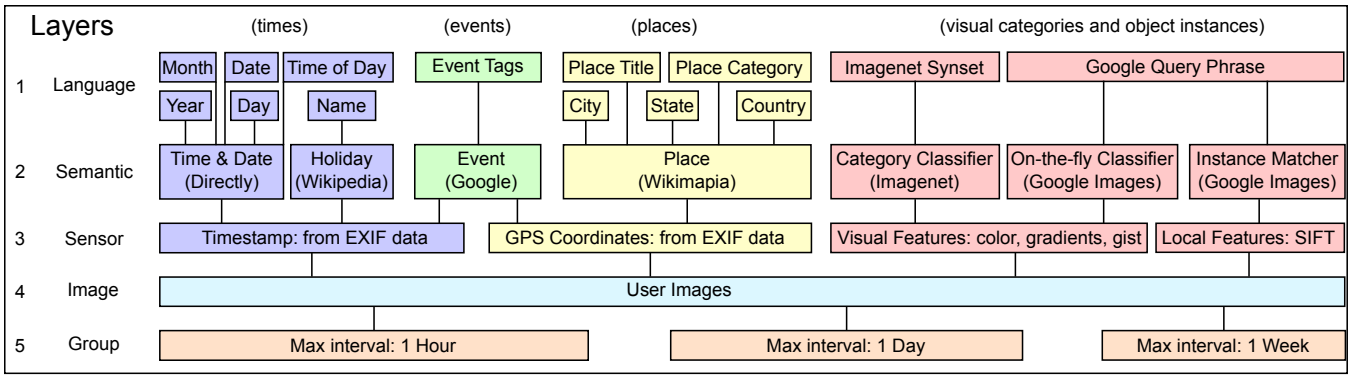


Figure 4: We represent all information in our system as nodes in a layered graph. Each colored box contains many nodes – individual bits of information – of a particular type (denoted by the name in the box). Lines between boxes indicate weighted connections between nodes of the two layers. Images are connected to their sensor values – timestamp and GPS, and low-level visual features. These are mapped into semantic concepts (*i.e.*, the things that people care about) through the use of Internet data sources, shown in parentheses. Finally, semantic nodes are exposed to search queries through the language layer, which contains text tags. By unifying all sources of information in this graph, we can easily incorporate new types of data to support novel types of queries, and perform fast and accurate search using any combination of terms.

IV. Layered Knowledge Graph

The previous section described several methods for obtaining labels for images; how should we store all of this data? A naive implementation might simply store a mapping from images directly to a list of {label, weight} pairs, and perhaps a reverse mapping to allow for fast searches. But this approach has several drawbacks – notably, it would be difficult to interpret search results due to lack of context for *why* a particular result was returned for a given query.

Instead, we store *all* data – not just labels and weights – in a hierarchical knowledge graph (see Fig. 4). The graph consists of nodes, each denoting a single conceptual piece of knowledge, connected via weighted edges, denoting the strength of the relation between two concepts. There are different types of nodes corresponding to different types of data, and each group of nodes of one type are assigned to a specific layer in the hierarchy. For example, a **place node** stores all the information about a given place from Wikimapia, which is connected above to **language nodes** denoting the place title, category, city, *etc*., and below to **GPS coordinate nodes** that are close to the given place.

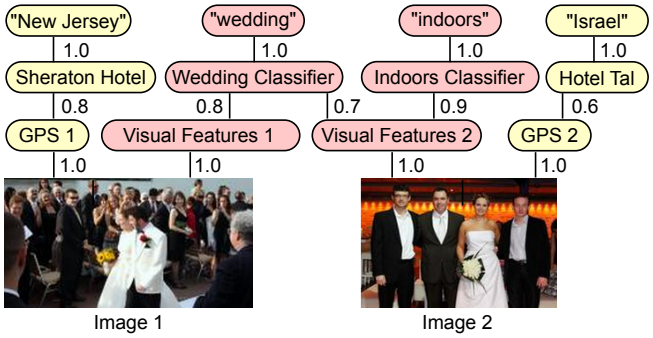
In Fig. 4, note that there is an additional layer below the image layer: groups. These groups are automatically created from images based on timestamps. By looking at the time intervals between successive photos, we build up a hierarchy of image groups (*e.g*., images taken within minutes of each other, or within days). Search results show groups instead than photos, so that rather than wading through a hundred nearly-identical shots of the party you attended, you see a small sampling of those images. Groups also function as a form of smoothing. As recognition is still an extremely challenging problem, classifiers might correctly label only a fraction of, say, *wedding* photos. By returning groups instead of images, users will be able to see photos of not only the highest classified (*i.e*., most prototypical) wedding photos – like the bride and groom at the altar – but also other, less wedding-like photos taken around the same time.

A. Search

Search on the graph consists of assigning scores to each node in the graph, starting at the top (layer 1: language) and propagating them down to the bottom (layer 5: groups), and then returning the top-scoring groups in sorted order. Figure 5 shows an example for the query "*wedding in Israel*" on a simplified graph containing two images of weddings, the first in New Jersey, and the latter at the Hotel Tal in Israel. The query is tokenized into "wedding" and "Israel" and then matched via string similarity to all nodes in the language layer, giving scores between 0 and 1 (most will be 0). Propagation to the next layer is accomplished by multiplying scores with the edge weights, summing up scores at each target node. Formally: we represent the edge weights between layers i and j as matrix E^i_j. Given scores s_i at layer i, we compute the scores at layer j as $s_j = E^i_j s_i$. In practice, E^i_j tends to be sparse, as most nodes only connect to a few other nodes. This makes propagation fast.

We repeat this process for each layer, until every node in the graph has an assigned score. We call a complete assignment of scores a *flow*, in this case, the *search flow*, F_{search}. Scores for each node in this flow are shown in the 3rd column of Fig. 5b. Notice that the final scores for the two images are 0.8 and 1.3, respectively, which means that we would display both images in the results, but with I_2 first, as it has the higher score (exactly what we want).

Simply showing the resulting image groups without any context would be confusing to a user, especially if the results are not obviously correct – she might wonder, "why did I get



(a) Simplified graph with precomputed edge weights

Layer	Node	$F_{search}(\downarrow)$	$F_{I_2}(\uparrow)$	$F_{desc_{I_2}}$
1	"New Jersey"	0.0	0.0	0.0
1	"wedding"	1.0	0.7	**10.7**
1	"indoors"	0.0	0.9	0.9
1	"Israel"	1.0	0.6	**10.6**
2	Sheraton Hotel	0.0	0.0	0.0
2	Wedding Classifier	1.0	0.7	10.7
2	Indoors Classifier	0.0	0.9	0.9
2	Hotel Tal	1.0	0.6	10.6
3	GPS 1	0.0	0.0	0.0
3	Visual Features 1	0.8	0.0	8.0
3	Visual Features 2	0.7	1.0	8.0
3	GPS 2	0.6	1.0	7.0
4	Image 1	0.8	0.0	8.0
4	Image 2	**1.3**	1.0	14.0

(b) Computed flows for query *wedding in Israel*

Figure 5: Given the graph in (a), the flows generated by the query *wedding in Israel* are shown in (b). See text for details.

this result?" Therefore, we label each returned image group with a short description, akin to the snippets shown in search engines that highlight elements from the results matching the user's query. Concretely, we create these descriptions using the following template:

what at **place**, **city**, **country** on **date**, **year**

The goal is to fill in each bolded component with a label from our graph. Since our language nodes are already separated into different types (see Fig. 4), this reduces down to simply choosing one node within each type of language node. We want to pick labels that are relevant to the images in the group, *i.e.*, language nodes connected to the image nodes via non-zero edges, biasing towards the terms used in the query, when applicable, so that it is clear why the images matched. Notice that we already know which labels these might be – they are the ones with non-zero scores in the search flow. Formally, we can write this down as a *description flow for Image I*: $F_{desc_I} = F_I + \lambda F_{search}$, where F_I is the *image flow*, described next, and λ is a weight determining how much to favor the query terms in the generated description. F_I is the flow created by applying a score of 1 to the image node and propagating scores *up* through the graph until we get to the language nodes. This flow describes how relevant each label is for the given image. Generating the description is then simply a matter of picking the highest scoring node in the description flow for each component in the template. The final generated description for Image 2 in Fig. 5 is "*wedding at Hotel Tal, Tel Aviv, Israel on January 3, 2011*." (Note that not all nodes needed to generate this description are shown in the figure.)

V. Quantitative Evaluation

Places: As described in Sec. III-B, we use the online crowd-sourced website Wikimapia for locating places around the GPS coordinates of photographs. To measure what kind of coverage it offers, we gathered geotagged images, manually labeled them, and then compared these ground truth annotations with search results from our system. We used a subset of place categories from the Scene UNderstanding (SUN) database [21] as search queries to get geotagged images from flickr. We labeled 1183 images of 32 categories and found 73.0% of all places were successfully found by our system when searching by name and 28.9% when searching by category. This task is quite challenging, as the list of places from flickr is extremely diverse: it spans dozens of countries and includes several obscure places. As Wikimapia continues to expand, we expect that recall rates will also increase.

Events: One of our major novel contributions is a general method for labeling photos of specific public events using a combination of place information from Wikimapia and simple NLP applied to the results of queries on Google (see Sec. III-C). We evaluate this capability using a methodology similar to that for our places evaluation described in the previous section. We search for different event categories on flickr and manually label the name of the event and/or key search terms shown in each image (*e.g.*, *New York Knicks* and *Boston Celtics* for a basketball game). We then search for these tags using our system. We successfully matched 17.3% of all tags, and 30.2% of all labeled images had at least one tag match. These measures distinguish between the situation where a venue is not found (in which case no event tags would be found), and that when the Google results are insufficient to get all of the right tags. The biggest problem was generally that the venue of the event was often not present on Wikimapia. Still, given the large variety of event types we tested, our performance on matching event tags is quite reasonable, especially given its generality.

Visual Categories: For evaluating visual classifiers, we labeled 5 personal photo collections and then queried our search engine with these labels, measuring performance using recall @ k – the fraction of the top k returned result groups that were correct (had a ground-truth annotation for that query). We feel that this is a fair metric for evaluating a system like ours, as a user is likely to be reasonably satisfied if she sees a relevant result on the first "page" of results. 42.4% of queries returned at least one relevant result in the top 5, and 49.6% in the top 10. This is quite remarkable because visual recognition is extremely challenging even in the standard "closed-world" regime (in which every image

belongs to exactly one of a fixed set of classes), whereas our system is "open-world," allowing a nearly infinite number of possibilities. Also, the best performing published methods are highly tuned and slow to train. In contrast, we download images, extract features, train a classifier, and run it on a user's collection in under 10 seconds – a very stringent operating scenario – and so we have favored speed over accuracy in our implementation.

VI. Conclusion

In this paper, we have described a system for searching personal photo collections in a flexible and intuitive way. By taking advantage of all types of sensor data associated with an image – timestamp, GPS coordinates, and visual features – we gather and generate a large set of semantic information that describes the image in the ways that people care about. This includes the *time* of the photo, specified as dates, holidays, or times of day; its *place*, in terms of names, categories, and common activities; *events* that this photo is a part of, such as concerts; *visual categories* exhibited in this photo, such as weddings, fireworks, or whales; and *object instances*, such as the Mona Lisa. We automatically label images by leveraging large online data sources, including Wikipedia for mapping dates to holidays, Wikimapia for mapping GPS locations to place information, Google for finding events, and ImageNet and Google Images for training visual classifiers and doing instance-level matching.

We believe that using the Internet to label personal photos is transformative: the user now gets to decide how to search, and doesn't need to spend time tediously labeling photos. Additionally, by allowing combinations of multiple query terms, we make it easy to find photos using whatever aspects of the photo she remembers. Handling people and faces is an obvious future work. Finally, we plan on deploying this system to real users soon, allowing us to better understand its effectiveness in practice.

Acknowledgments

This work was supported by funding from National Science Foundation grant IIS-1250793, Google, Adobe, Microsoft, Pixar, and the UW Animation Research Labs.

References

[1] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "On-the-fly specific person retrieval," in *International Workshop on Image Analysis for Multimedia Interactive Services*, 2012.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 5:1–5:60, May 2008.

[3] M. Naaman, Y. J. Song, A. Paepcke, and H. G. Molina, "Automatic organization for digital photographs with geographic coordinates," in *Joint Conf. on Digital Libraries*, June 2004.

[4] T. Quack, B. Leibe, and L. Van Gool, "World-scale mining of objects and events from community photo collections," in *Intl. Conf. Content-based image and video retrieval*, 2008.

[5] D. Joshi, J. Luo, J. Yu, P. Lei, and A. Gallagher, "Using geotags to derive rich tag-clouds for image annotation," in *Social Media Modeling and Computing*. Springer, 2011.

[6] D. Kirk, A. Sellen, C. Rother, and K. Wood, "Understanding photowork," in *SIGCHI*, 2006, pp. 761–770.

[7] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox, "Temporal event clustering for digital photo collections," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 1, no. 3, pp. 269–288, 2005.

[8] N. Kumar, P. N. Belhumeur, and S. K. Nayar, "Facetracer: A search engine for large collections of images with faces," in *ECCV*, October 2008.

[9] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma, "Image annotation by large-scale content-based image retrieval," in *ACM Multimedia*, 2006, pp. 607–610.

[10] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *IJCAI*, 2011.

[11] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Jnl. of Computer Vision (IJCV)*, 2004.

[12] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching," in *ICCV*, 2003.

[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[17] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *ICCV*, vol. 2, 2005, pp. 1816–1823.

[18] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What helps where–and why? semantic relatedness for knowledge transfer," in *CVPR*, 2010.

[19] X. Chen, A. Shrivastava, and A. Gupta, "NEIL: Extracting visual knowledge from web data," in *International Conference on Computer Vision*, vol. 3, 2013.

[20] A. Oliva and A. Torralba, "Modeling the shape of scene: A holistic representation of the spatial envelope," *IJCV*, 2001.

[21] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition," in *CVPR*, 2010.