Searching Personal Photos on the Phone with Instant Visual Query Suggestion and Joint Text-Image Hashing

Zhaoyang Zeng[†], Jianlong Fu[‡], Hongyang Chao[†], Tao Mei[‡]

[†]School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China
[‡]Microsoft Research, Beijing, China
zengzhy5@mail2.sysu.edu.cn; {jianf, tmei}@microsoft.com; isschhy@mail.sysu.edu.cn

ABSTRACT

The ubiquitous mobile devices have led to the unprecedented growing of personal photo collections on the phone. One significant pain point of today's mobile users is instantly finding specific photos of what they want. Existing applications (e.g., Google Photo and OneDrive) have predominantly focused on cloud-based solutions, while leaving the client-side challenges (e.g., query formulation, photo tagging and search, etc.) unsolved. This considerably hinders user experience on the phone. In this paper, we present an innovative personal photo search system on the phone, which enables instant and accurate photo search by visual query suggestion and joint text-image hashing. Specifically, the system is characterized by several distinctive properties: 1) visual query suggestion (VQS) to facilitate the formulation of queries in a joint text-image form, 2) light-weight convolutional and sequential deep neural networks to extract representations for both photos and queries, and 3) joint text-image hashing (with compact binary codes) to facilitate binary image search and VQS. It is worth noting that all the components run on the phone with client optimization by deep learning techniques. We have collected 270 photo albums taken by 30 mobile users (corresponding to 37,000 personal photos) and conducted a series of field studies. We show that our system significantly outperforms the existing client-based solutions by 10× in terms of search efficiency, and 92.3% precision in terms of search accuracy, leading to a remarkably better user experience of photo discovery on the phone.

CCS CONCEPTS

• **Computing methodologies** → **Visual content-based indexing and retrieval**; **Image representations**;

KEYWORDS

Visual query suggestion, deep learning, image search, hashing, mobile.

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123446

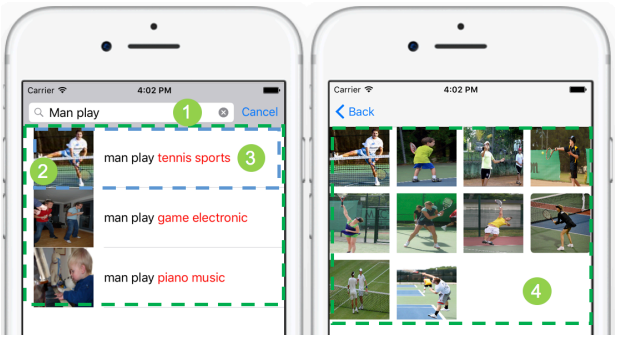


Figure 1: The procedure of image search with visual query suggestion. User can (1) enter a few characters; (2) browse the instantly suggested items; (3) select one suggestion to specify search intent; (4) perform image search based on the new query and the compact hashing technique. It is worth noting that all these components run on the client.

1 INTRODUCTION

Recent years have witnessed the ubiquity of mobile devices (e.g., smart phones, digital cameras, tablets, etc.). This has led to an unprecedented growth in the number of personal photos on the phone. People are taking photos using their smart devices every day and everywhere. One significant pain point of today's mobile users derives from the challenge of instantly finding specific photos of what they want. Specifically, it is often not practical for a mobile user to search a desired photo by browsing at least thousands of photos, due to the limited screen size of mobile devices. Although a great number of photo management systems on iOS and Android platforms focus on using timestamp and GPS [2, 7, 16, 21] to categorize personal photos for relieving the heavy user memory load, the search experience is far from satisfactory for photo search. Because time and location are less representative than semantic words which often used in Web search engines, and thus considerably hinder friendly user experience on the phone.

Extensive research on both academic and industrial fields have been made by proposing cloud-based solutions [5, 12–14, 39, 45], which usually consists of image data transmitting from client to cloud, image semantic tagging and online image search on cloud.

∗ This work was performed when Zhaoyang Zeng was visiting Microsoft Research as a research intern.

Typical commercial products include Google Photos[1] and Microsoft OneDrive[2], both enable effective photo indexing and search not only by time and location, but also by face grouping and object/scene understanding, and thus photos can be found by entering human-friendly tags (e.g., "beach," "sunset") in the search box. However, the cloud-based solution requires sending each photo to a remote server on the cloud, which can hardly guarantee instant photo indexing and search on the phone, due to network latency. With the increasing computing capacity on mobile clients, designing effective client-based photo search systems on the phone becomes a fundamental challenge for better user experience. Further improvement has been observed from the recently-released photo management system on iOS 10, which can provide efficient photo grouping by face, tag, time and location on the phone. Once photos have been tagged and indexed on the phone, the system further supports semantic photo search by typing a single word from a predefined vocabulary. However, the system usually suffers from low recall due to the limited vocabulary of queries that can be searched. Therefore, designing a system with the capability of instantly finding arbitrary photos of what users want still remains largely unexplored.

In this paper, to solve the above challenges on the clients, we propose to consider the following several challenges for developing a personal photo search system on the phone. 1) Query formulation: due to the inconvenience for typing complete queries on mobile screens, query auto-completion by visual query suggestion is significantly crucial for mobile users, which can enable fast search on the phone. 2) Photo indexing: as the limited number of tags are inadequate to present the rich image appearances, existing systems are difficult to discover the desired photos to arbitrary user queries. Therefore learning an embedding space across visual and textual domains with rich semantics that can represent diverse images are indispensable for client-side applications. 3) Computational cost: as the computational capacity (e.g., CPU and memory) of mobile devices are not comparable with cloud servers, the state-of-the-art image and query feature extraction models (e.g., VGG-19 [38], ResNet-152 [17] for images or LSTM [15] for queries) with large memory or heavy computational cost are not suitable for mobile clients. 4) Instant response: since user experience should be greatly emphasized for mobile user scenarios, the search process is expected to be instant. Specifically, the query suggestions and search results should be returned instantly or even progressively during the user querying process.

Motivated by the above observations, we have developed an innovative personal photo search system on the phone, which enables instant and accurate photo search by client-based visual query suggestion (VQS) and joint text-image hashing. The system consists of offline image feature extraction with compact binary codes, and online VQS and instant photo search. Specifically, image features are offline extracted on the client by a light-weight convolutional neural network (CNN) [26] and a novel joint text-image hashing technique. The hash function is optimized in a text-image embedding space by contrastive losses and orthogonal constrains, which ensures powerful yet compact image representations over rich semantic space. The learned hash codes are further indexed into a concept hierarchical tree, from which different categories are considered as diversified query suggestion phrases to clarify the initial user input.

The online stage is composed of instant query understanding by recurrent neural network (RNN) [44], query reformulation by VQS, and fast photo retrieval by progressive binary search. In particularly, an initial user query is parsed into binary codes by a light-weight Gated Recurrent Unit (GRU) [6] and the joint text-image hashing. Once the initial query has been understood, an instant search on hashing space is conducted to discover the top-N nearest photos. To facilitate the reformulation of queries, we propose the photos that belong to different categories in the concept tree together with their corresponding categories as joint text-image query suggestions. Once users select one of the suggestions, a more precise photo search can be conducted progressively to rerank the top-N photos by using the user-selected suggestion as query example and binary image search. The screenshot of the proposed system is shown in Figure. 1.

To the best of our knowledge, this work represents the one of first attempts for developing an instant personal photo search system with client-side integration by deep learning techniques. Our contributions can be summarized as follows:

- We have designed a fully-client image search system towards instant and progressive personal photo search by leveraging light-weight computing capacity of mobile devices, which is quite distinct from existing cloud-based solutions.
- We propose a novel joint text-image hashing network that can instantly and accurately produce visual query suggestions and visual search results, and thus real-time query suggestion and fast image search can be achieved.
- We conduct comprehensive field studies from a real user collection of 30 mobile users, 270 albums and 37,000 photos, and show superior performance of the developed system on both objective and subjective evaluation.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 introduces the proposed system. Section 4 provides the evaluation, followed by the conclusion in Section 5.

2 RELATED WORK

2.1 Query Suggestion for Image Search

Query suggestion (QS) aims to help users formulate precise queries to clearly express their search intents by suggesting a list of complete queries based on users' initial inputs. This technique has been widely-used in commercial search engines, such as Google and Bing. Most earlier works focus on suggesting relevant queries while ignoring the diversity in the suggestions, which potentially provide inferior experiences for users [8, 34]. Although promising works have been further proposed to prevent semantically redundant suggestions by Markov random walk and hitting time analysis [20, 31], which focus on text-only query suggestion. Significant progress has been made by introducing VQS [52, 53], which can generate joint text/image suggestions for initial queries. Because images can carry more vivid information and help users better specify their search intents, VQS provides users with friendly experiences.

[1] https://photos.google.com/
[2] https://onedrive.live.com/

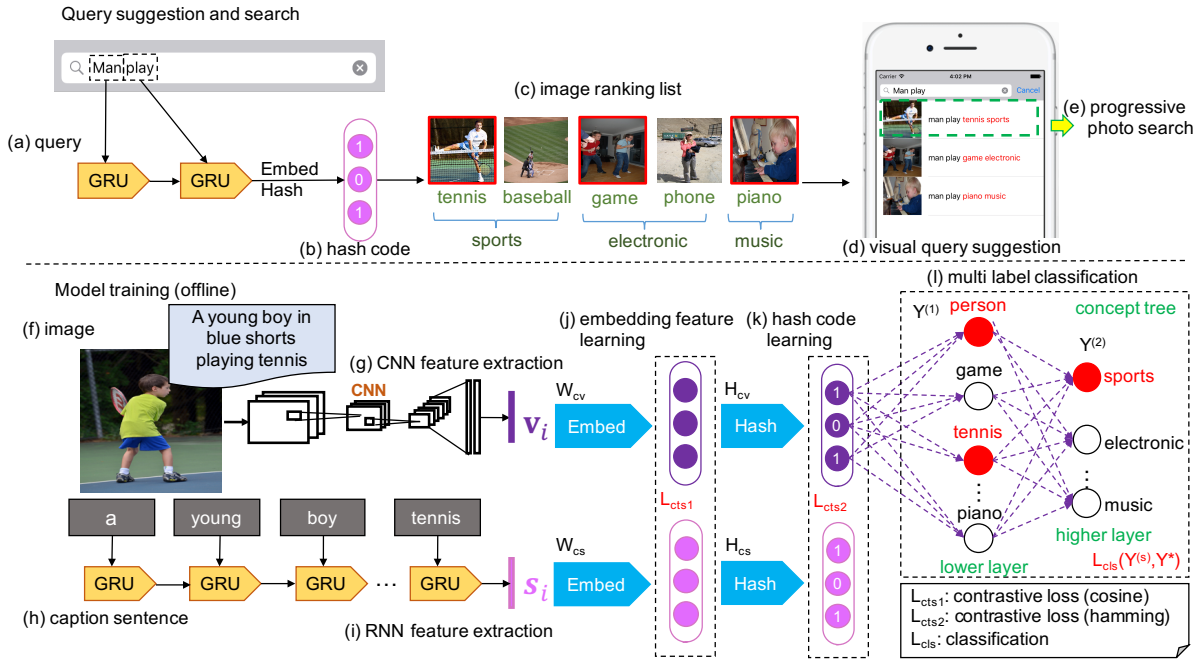


Figure 2: The overview of the proposed innovative personal photo search system on the phone. The system consists of two closely-related modules, i.e., online visual query suggestion and fast image search (a-e), and offline model training (f-l). Light-weight CNN and RNN models are proposed to use in (g) and (i), and a novel joint text-image hashing network is designed in (j) and (k) with embedding matrixes W_{cv}/W_{cs} and hashing functions H_{cv}/H_{cs} for instant query suggestion and fast image search. The hashing network is optimized by both two contrastive loss L_{cts1}, L_{cts2}, and one classification loss L_{cls}. $Y^{(s)}$ and Y^* denote prediction and ground truth labels. s can be $\{1, 2\}$ and denotes two levels in the concept tree. [Best viewed in color]

However, existing approaches predominantly solve this problem on predefined queries of single word, while neglecting the facts that queries can be sequences of words with arbitrary length [18, 36, 42]. Due to the success of natural language understanding and image captioning by using recurrent neural networks (RNN) [15, 30, 44, 48], some pioneer works have been proposed for understanding long queries on textual query suggestion [35, 42] and object retrieval [18], whereas few works have discussed the accessibility on VQS. Besides, designing efficient VQS algorithms on clients also remains largely unexplored.

2.2 Image Hashing

Joint text-image hashing, (i.e., cross-modal hashing), was a popular topic in machine learning [1] and multimedia retrieval [11, 29]. Cross-modal hashing can be divided into unsupervised and supervised methods. Unsupervised hashing methods usually adopt the paradigm of minimizing construction/quantization errors and preserving image similarity in Hamming space with unsupervised training data [11, 41, 47]. As supervised hashing can leverage semantic information to enhance the cross-modal correlation and reduce the semantic gap, they can achieve superior performance than unsupervised methods for cross-modal image retrieval [3, 19, 29, 46, 51]. Significant progress has been proposed by effectively exploiting deep learning networks, which unifies convolutional neural network (CNN) and recurrent neural network (RNN) for image and text modeling, respectively. The most relevant work to ours comes from Cao et.al. [3], which presents a deep visual-semantic hashing model to generate compact representations of images and sentences in an end-to-end deep learning architecture for cross-modal image retrieval, as a result of the highly efficient hash codes. However, how to implement an effective visual query suggestion framework in terms of algorithm and application design by cross-modal hashing is still unexplored. In this paper, we formulate VQS as a joint cross-modal retrieval and classification problem, and generate relevant/diversified visual query suggestions by hash codes.

3 SYSTEM

To develop an instant and accurate photo search system on the phone, we propose to leverage VQS and joint text-image hashing techniques. VQS can help conduct instant user query formulation, while the joint hashing can benefit fast cross-modality image search. Specifically, the system is designed with two closely-related modules, i.e., online visual query suggestion, progressive photo search (Figure 2 (a-e)), and offline model training (Figure 2 (f-l)).

In online stage, given an initial user input in (a), we embed and quantize the query into semantic hash codes by the learned GRU model and joint text-image hash functions from (i-k). Once initial query has been projected to binary codes in (b), an instant search on hashing space is conducted to discover the top-N nearest photos in (c). Since photos have been indexed into a concept hierarchical tree in (l), different categories (e.g., "tennis sports," "piano music") from the concept tree can be instantly generated and associated to

the top-ranked images. Different categories with their corresponding photos are further proposed as joint text-image query suggestions to facilitate the reformulation of queries in (d), and thus relevant yet diversified textual/visual query suggestions can be generated. Once users select a suggestion in (d), a progressive photo search can be conducted to re-rank the top-N photos by using it as a query example and binary image search. In offline stage, the model is trained by taking a collection of images in (f) with corresponding descriptive sentences in (h), and learning the latent relationship between visual representations in semantic embedding and hashing spaces in (j) and (k). This cross-modal embedding can benefit the deep understanding between queries and photos, and thus enables accurate image search.

3.1 Visual-Semantic Representation

Image representation: since CNN has been widely used for image classification/captioning tasks with great success [10, 27], we adopt CNN to extract deep image representations. Specifically, given an image I, we compute its visual representation as follows:

$$V = W_v[CNN(I)] + b_v, \tag{1}$$

where $CNN(I)$ transforms an image into a 1024-dimensional vector. The matrix W_v has dimension $d_v \times 1024$, in which d_v is the size of the embedding space. In our experiment, we deploy GoogLeNet [43] as the CNN architecture on clients to extract image features, due to its less memory consumption. We adopt $d_v = 512$ as in [22], and thus each image can be represented in a d_v-dimensional vector. In order to further reduce the feature extraction time and battery consumption for GoogLeNet on mobile devices, we compress the CNN model using the methods in [23]. We apply Trucker Decomposition on convolution kernel tensors and fine-tune the model to recover the accumulated loss. The computational complexity can be reduced by half, compared with un-compressed models [23].

Sentence representation: since the user queries often contain phrases of two or more words, it is important to learn the word dependency for catching the user intent. Therefore, we use Recurrent Neural Networks (RNN) to learn a language model, due to its success in sequence learning. Long-short time memory (LSTM) [15] is the state-of-the-art RNN architecture, however it is not suitable for mobile clients because of the heavy computational complexity and time cost. Gated Recurrent Unit (GRU) [6] is a light-weight RNN architecture proposed in recent years. The performance of GRU is comparable with LSTM, while GRU has much lower computational cost. Therefore, we adopt GRU for sentence representation, in which the computational complexity can be reduced by half, compared with LSTM models.

Particularly, we consider the index $t = \{1 \dots N\}$ as the position of a word in a sentence. Given an index t, GRU receives inputs from 1) current input x_t, 2) previous hidden state h_{t-1}, given by:

$$\begin{aligned} x_t &= W_w \mathbb{1}_t, \\ z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z), \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r), \\ \widetilde{h} &= tanh(W_h x_t + (U_h r_t \odot h_{t-1}) + b_h), \\ h_t &= z_t \widetilde{h} + (1 - z_t) h_{t-1}, \end{aligned} \tag{2}$$

where $\mathbb{1}$ is an indicator column vector that has a single one at the index of t^{th} word in a word vocabulary. The weight W_w specifies a word embedding matrix, which is initialized with pre-trained parameters. $W_z, W_r, W_h, U_z, U_r, U_h$ are weight matrixes, and b_z, b_r, b_h are bias vectors in GRU. σ is the sigmoid activation function with the form of $\sigma(x) = 1/(1 + exp(-x))$. $\odot$ denotes the product with a gate value, as defined in [6]. We further consider the bidirectional framework to encode the state along timesteps from the past and future. We rewrite the GRU in Eqn. (2) into a compact form:

$$(z_t, r_t, s_t, \widetilde{h}, h_t) = GRU(x_t, h_{t-1}; W, U, b), \tag{3}$$

and the components of BiGRU can be written as:

$$\begin{aligned} (z_t^f, r_t^f, s_t^f, \widetilde{h}^f, h_t^f) &= GRU(x_t, h_{t-1}^f; W^f, U^f, b^f), \\ (z_t^b, r_t^b, s_t^b, \widetilde{h}^b, h_t^b) &= GRU(x_t, h_{t+1}^b; W^b, U^b, b^b), \\ h_t &= [h_t^f, h_{(N-t+1)}^b], \end{aligned} \tag{4}$$

where the superscript f indicates the forward pass and b denotes the backward pass. We consider $S = [h_N^{uni}, h_N^{bi}]$ as the representation of a sentence, where h_N^{uni} and h_N^{bi} are the outputs from the last cell in GRU and BiGRU. We set the dimensions of the hidden layers of GRU and BiGRU are both $d_s/2$, thus each sentence can be represented in a d_s-dimensional vector. To learn the joint embedding spaces with images, we set $d_s = 1200$ to make sure that the model is able to learn the sentences representation well and is small enough to run on mobile devices. Besides, we set the word embedding size as 620 like [25].

3.2 Joint Text-Image Hashing

To support real time visual query suggestion and search, we propose to learn a joint cross-modal hashing embedding. In particular, both image and sentence representations from CNN and RNN are encoded into the same Hamming space with hash functions, where the Hamming distance between aligned image-sentence pairs are expected to be small. We first embed both images and sentences representations into m-dimension, which is given by:

$$c(x) = W_c^T x + b_c, \tag{5}$$

where x denotes the input vector with the dimension of d, and matrixs W_c denotes weights in hash layer with the dimension of $d \times m$, where m denotes the number of hash bits. When $c(x)$ takes images features as input, $d = d_v$, and otherwise $d = d_s$ for sentence inputs. The hash function can be defined as as:

$$h(x) = \text{sign}(c), \tag{6}$$

where $\text{sign}(x)$ is a sign function, where $\text{sign}(x_i) = 1$ if $x_i > 0$ and otherwise $\text{sign}(x_i) = 0$. Due to the discontinuous sign function, we relax $h(x)$ to:

$$h(x) = 2\sigma(c) - 1, \tag{7}$$

where $\sigma(\cdot)$ denotes the logistic function.

For query suggestion, we aim to consider both semantical relevance and diversity at the same time. We can easily ensure the relevance by minimizing the Hamming distance between the initial and suggested queries. To maximize the diversity, we propose to index the learned hash codes into a two-layered concept hierarchial tree, which is shown in Figure 2 (l).

We consider the problem as a multi-label classification task, and the concept tree can be represented as two fully-connected layers, which is given by:

$$[p^{(l)}, p^{(h)}] = g(x), \tag{8}$$

where the specific form of $g(\cdot)$ can be represented by two-stacked fully-connected layers with sigmoid activation, and $p^{(l)}, p^{(h)}$ denotes the outputs of the first (the lower) and second (the higher) sigmoid activation layers. We consider the categories corresponding to outputs of $argmax(p^{(l)})$ and $argmax(p^{(h)})$ as the low-level and high-level predictions in the concept tree. We combine the predicted categories from low (e.g., "tennis") and high (e.g., "sports") layers to form the suggested phrases (e.g., "tennis, sports") for visual query suggestion. We think this concept tree can directly benefit hash codes learning, so we concatenate it behind the hash layer insead of CNN or RNN layers.

3.3 Loss Function

To learn the deep correspondence between aligned text and image pairs, we propose to jointly optimize the semantic embedding and hashing space by contrastive losses. Specifically, given an image k and a sentence l, we adopt the dot product $h_i(V_k)^T \cdot h_s(S_l)$ to approximate the similarity measurement in Hamming space [11]. This similarity is further normalized by the number of hash bits, which is given by:

$$H_{kl} = \frac{h(V_k)^T \cdot h(S_l)}{m}. \tag{9}$$

To better enhance the semantic correlation between images and sentences, we propose to take the non-quantized feature similarity before hashing into consideration. Cosine distance is adopted to measure this semantic correlation, due to its good performance shown in previous research [4], which is given by:

$$C_{kl} = \frac{c(V_k)^T \cdot c(S_l)}{\|c(V_k)\|_2 \cdot \|c(S_l)\|_2}. \tag{10}$$

To learn the metric in embedding spaces, we encourage the aligned image-sentence pairs in training set to have a higher similarity score than misaligned pairs by a margin, and thus a contrastive loss [54] is adopted. Particularly, the loss function is given by:

$$\begin{aligned} L_{cts} = \sum_k [\sum_l & max(0, H_{kl} - H_{kk} + 0.1) \\ & + \sum_l max(0, H_{lk} - H_{kk} + 0.1) \\ & + \sum_l max(0, C_{kl} - C_{kk} + 0.1) \\ & + \sum_l max(0, C_{lk} - C_{kk} + 0.1)], \end{aligned} \tag{11}$$

where L_{cts} denotes the contrastive loss, 0.1 is empirically set to be a margin, and $k = l$ denotes a corresponding image and sentence pair. The loss function is symmetric with respect to images and sentences. As observed by [37], this formulation can benefit the learning of predition by conditioning on both images and sentences with superior performance than asymmetric models.

Algorithm 1 Query suggestion algorithm (detail in Sec. 3.4)

Input: Images ranking list L
Output: Candidate suggested images list C

```
Initialize set S = ∅, C = ∅
for each i ∈ [1, N] do
    y = argmax(p^(h)(L_i))
    if |S| < k and y ∉ S then
        S = S ⋃ {y}; C = C ⋃ {L_i}
    end if
end for
if |S| == 1 then
    set S = ∅, C = ∅
    for each i ∈ [1, N] do
        y = argmax(p^(l)(L_i))
        if |S| < k and y ∉ S then
            S = S ⋃ {y}; C = C ⋃ {L_i}
        end if
    end for
end if
return C
```

The classification task can be solved by minimizing the cross entropy loss [32], which is given by:

$$L_{cls} = \sum_s [L_{cls}(\mathbf{Y}^*, \mathbf{Y}^{(s)})], \tag{12}$$

where L_{cls} denotes the classification loss, $\mathbf{Y}^*$ indicates the ground truth and $\mathbf{Y}^{(s)}$ is the label prediction. s can be $\{1, 2\}$ and denotes two levels in the concept tree.

To enhance the representative ability of hash bits, we enforce an orthogonality constraint in the learning of hash functions. As relaxed in [49], the loss of orthogonality constraint is given by:

$$L_{otg} = \frac{1}{m}(\|W_{ci}^T W_{ci} - I\|_F^2 + \|W_{cs}^T W_{cs} - I\|_F^2), \tag{13}$$

where L_{otg} denotes the orthogonality constraint, W_{ci} and W_{cs} are two types of W_c in Eqn. (5), and $\|\cdot\|_F$ represents the Frobenius norm, and I is an identity matrix. The final optimization function of the proposed joint text-image hashing is formulated as follows:

$$L = L_{cts} + \lambda_1 L_{cls} + \lambda_2 L_{otg}, \tag{14}$$

where λ_1 and λ_2 are hyper parameters. To evaluate the effects, we set the range of λ_1 and λ_2 from 0 to 1, and finally set $\lambda_1 = 1$ and $\lambda_2 = 0.1$ since this setting can help achieve the best performance in our experiment.

3.4 Visual Query Suggestion and Search

Once the joint hashing model has been optimized, we apply it to instant query suggestion and photo search systems. Given a collection of user photos on clients, we first calculate the hash codes of each photo by using the learned hash functions. Note that these hash codes can be used in both query suggestion and photo search.

Visual query suggestion: the designed procedure of the system for VQS is shown as follows: given an initial query from users, we first extract its hash code from Figure 2 (a,b). Based on Hamming distance, we select the top-N closest images in a image ranking list from Figure 2 (c), which is denoted as list L. We call this step

Table 1: The training/validation pairs, vocabulary size, and node numbers in the concept tree.

Datasets	# trn	# val	#vocab	#nodes(low/high)
Flickr30K [50]	154k	5k	7.4k	200/12
MSCOCO [28]	414k	203k	8.8k	500/12

as "initial search." We apply Algorithm 1 to obtain the candidate suggested images list C. The images in C as well as their category terms in the concept hierarchial tree are provided to users as the joint text-visual suggestion results. In our experiment, we find that the ranges from three to five can be the suitable values for k, i.e., the number of categories to be selected for VQS, and we usually take $N = 100$.

Progressive photo search: for initial search, the retrieval list of images is produced by sorting the Hamming distances of hash codes between the query sentence and images in search pool. Once users have selected one of the suggestions, we further propose a progressive image search to rerank the images from top-N image search list. Since the top-N images are retrieved by the initial search and the number of N would not be large, the progressive search can be accurate and efficient.

3.5 Training

Initialization: for the CNN model, we pre-train it by ImageNet [9], and fine-tune by a concept vocabulary $c_i, i = \{1, 2, \cdots C\}$ in MSCOCO dataset [28], where each concept c_i is a single word. Details of the mining of these highly-frequent concepts can be found in Section 4.1. For the RNN model, we initialize the word embedding matrix W_w with skip-thought vector [25], which is trained on a large book-collection corpus.

Optimization: in Eqn. (11), we need to calculate the similarity between each pair of images and sentences, the computational complexity is too high. To accelerate the training, we iterate the aligned and misaligned pairs in a training batch, and the images and sentences in each batch are randomly selected in each epoch.

4 EXPERIMENT

In this section, we conduct quantitative comparisons and extensive user studies on both standard cross-modality image search datasets and a collection of real-world photos from mobile users to evaluate the usability for the proposed system.

4.1 Datasets and Baselines

To conduct quantitative evaluation for the proposed joint hashing approach, we select **MSCOCO** [28] and **Flickr30K** [50] for both training and testing. Because the two image captioning datasets are so challenging that are widely-used for cross-modality embedding and hashing evaluations. Detailed statistics can be found in Table 1. To make a fair comparison, we follow the same settings as [3, 22] to randomly select 5,000/1,000 images from validation set as testing images for MSCOCO and Flickr30K, respectively.

To make comprehensive user study, we have collected a real-world personal photo datasets of 270 photo albums (corresponding to 37,000 personal photos) from 30 volunteers, including 20 students and 10 staffs. All of them are experienced image search engines users with more than 500 photos on their phones. The age distribution, career distribution, categories of searched key words,

Table 2: Search results of different variations of our model with 128 hash bits on Flickr30K dataset.

Method	Med r	R@1	R@5	R@10
L_{cts}	13	17.2	41.1	53.9
$L_{cts} + 0.1L_{otg}$	13	17.9	41.1	53.2
$L_{cts} + L_{cls} + 0.1L_{otg}$	11	17.3	42.6	55.5

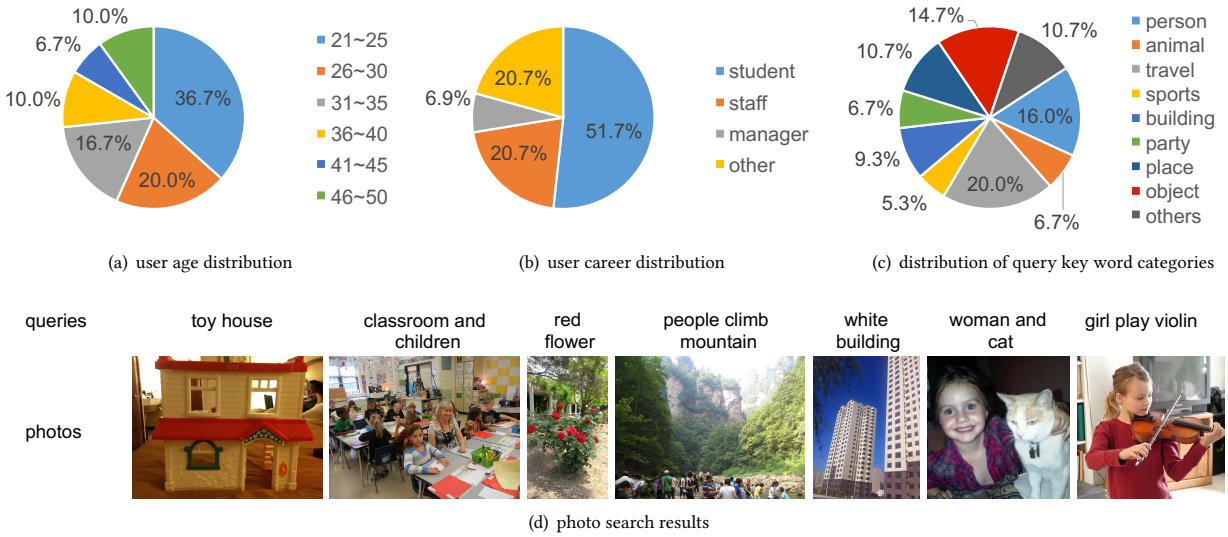
Table 3: Cross-modality image retrieval results by binary search. R@K and Med r represents Recall@K (higher is better) and median rank (lower is better).

Model	Med r	R@1	R@5	R@10
Flickr30K				
SDT-RNN [40]	16	8.9	29.8	41.1
LBL [24]	13	11.8	34.0	46.3
m-RNN [33]	16	12.6	31.2	41.5
BRNN [22]	13	18.5	42.1	56.8
DVSH [3]	9	17.1	39.9	51.9
Our Model (512-bit)	**9**	**18.5**	**43.9**	**57.0**
MSCOCO 1K test images				
BRNN [22]	4	31.8	67.1	80.1
DVSH [3]	3	29.6	64.8	78.1
Our Model (512-bit)	**3**	**32.7**	**67.1**	**81.0**
MSCOCO 5K test images				
BRNN [22]	11	14.1	38.1	50.9
DVSH [3]	11	13.3	35.6	48.6
Our Model (512-bit)	**11**	**14.6**	**38.0**	**51.3**

and exemplar search queries with results are shown in Figure 3 (a-d), respectively. We use the model trained on MSCOCO to conduct user studies for VQS and photo search, as MSCOCO has larger vocabulary size. To obtain the concept tree, we select the top 500 noun words (e.g., "tennis," "piano") with the highest word frequency from MSCOCO as the lower level concepts, and use the 12 general category terms (e.g., "sport," "music") provided by MSCOCO as the higher level concepts. Each image can be annotated with one or more concepts by the specific 500 nouns, if a noun occurs in its caption. Images are also categorized by the 12 high-level terms by the annotation from MSCOCO.

We compare with five approaches to evaluate the quantitative binary search for cross-modality image retrieval. We also compare with two VQS systems in research community, and two commercial image search engines for user study. The seven baselines in research community are listed as follows:

- **SDT-RNN** [40]: **s**emantic **d**ependency **t**ree **r**ecursive **n**eural **n**etworks uses words and the dependency tree as inputs to train with contrastive loss.
- **LBL** [24]: **l**og-**b**ilinear **l**anguage model proposes to use lstm to encode sentences.
- **m-RNN** [33]: **m**ultimodal **R**ecurrent **N**eural **N**etwork proposes to connect the language model and the deep CNN together by a one-layer representation.
- **BRNN** [22]: use **b**idirection **r**ecurrent **n**eural **n**etwork to compute the word representations.
- **DVSH** [3]: **d**eep **v**isual-**s**emantic **h**ashing propose to learn a joint hash embedding space of images and sentences.



(a) user age distribution (b) user career distribution (c) distribution of query key word categories

(d) photo search results

Figure 3: A dataset of real-world personal photos. The above figures show the age, career distribution of volunteers, and exemplar query terms with corresponding results.

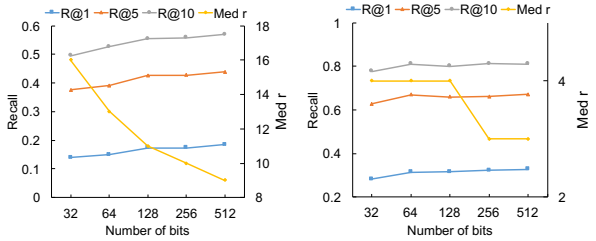
- **VQS** [53]: visual query suggestion provides the first attempt to help users to precisely express their search intents by joint text and image suggestions.
- **CAQS** [42]: context-aware query suggestion proposes a hierarchical recurrent encoder-decoder generative network for queries with arbitrary lengths.

4.2 Evaluation of Binary Search

To evaluate the proposed joint hashing approach, binary search is conducted by taking the captions as queries, and searching the corresponding images. We report the median rank of the closest ground truth result in the image ranking list, and Recall@K which measures the fraction of times a correct item was found among the top K results. For MSCOCO dataset, we report the results on a subset of 1,000 images and the full set of 5,000 test images, according to the same setting with [22]. We can observe from Table 3 that our hashing model with the same feature dimensions can achieve comparable results with the non-quantized approaches **BRNN** [22], and obtain even better results than the state-of-the-art joint hashing approaches **DVSH** [3]. Detailed comparison of different variations of our model and the performances of different hash bits can be found in Table 2 and Figure 4. Considering most phones can display 30 photos on average, we further calculate Recall@30 and obtain 92.3% accuracy on MSCOCO 1k test set, which ensures satisfying results for mobile users.

4.3 Time and Memory Consumption

System Latency and memory cost play key roles for a mobile application. We calculate the time and memory consumption for each module in Table 5. We test different modules on two typical mobile devices. Device I is iPhone 6S with a 2GB RAM and 64GB ROM, and device II is Samsung Galaxy S7 with a 4GB RAM and 64GB ROM. We store 10,000 photos on the two devices, respectively, and test the time cost on image and query embedding/hashing, query embedding/hashing, initial and progressive search. Since image processing can be conducted offline, we find that our system can act an instant response to return the suggestion and search result around 0.1s for both the two devices, which is adequate for real-time applications and can ensure friendly user experience. The peak memory is about 73.0 MB, which is moderate for most mobile clients, due to the light-weight GoogLeNet with model compression techniques and the GRU architecture used in our system.



(a) Flickr30K (b) MSCOCO

Figure 4: Experiment result on Flickr30K and MSCOCO datasets with different number of hash bits on 1K test images.

4.4 Relevance and Diversity for VQS

We use the model trained on MSCOCO dataset to build an image search system, and ask 30 volunteers to try it. To evaluate the relevance between the suggestions and the initial queries and the diversity of query suggestions, all volunteer were asked to search ten times by different queries and report two scores from "0" to "3" for every generated suggestions, in which one is relevance score and another is diversity score. The relevance score indicates that there are no, one, two and three closely-related suggested items, and the diversity score denotes that all the related suggestion items have zero, one, two and three different meanings. More information could be found in Figure 6. We observe that long queries (consists of at least three words) have higher relevance scores than short queries, because long queries contain more user intentions, from which our RNN model can well-understand. However, as the

Table 4: Query Suggestions Comparisons between different search engines

Query="wood"					Query="man play"				
ours	VQS	CAQS	srh eng 1	srh eng 2	ours	VQS	CAQS	srh eng 1	srh engss 2
funiture	glass	window	texture	woodone	sports	fling	tennis	guitar	games
appliance	plastic	table	floor	wood job	electronic	meet gitar	locals	piano	guitar
outdoor	metal	room	background	woodman	music	sex	basketball	violin	golf
Query="cake and party"					Query="two men at canteen"				
ours	VQS	CAQS	srh eng 1	srh eng 2	ours	VQS	CAQS	srh eng 1	srh eng 2
food	-	birthday	birthday	pie	person	-	sitting	-	-
kitchen	-	people	cocktail	candle	food	-	bench	-	-
person	-	table	song	cafe	kitchen	-	store	-	-

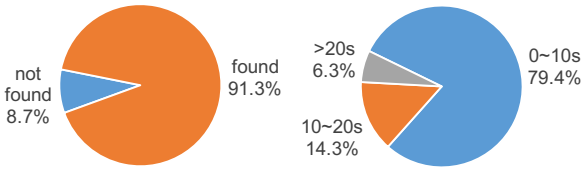


Figure 5: Accuracy and search time. a) the percent of the photos that can be searched; b) search time.

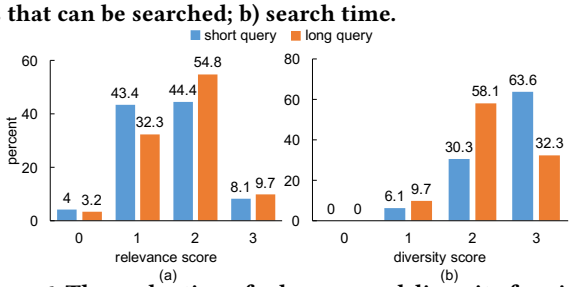


Figure 6: The evaluation of relevance and diversity for visual query suggestion results.

user intentions are clear, the diversity of query suggestions would be limited, which can be observed from Figure 6 (b).

4.5 Search Time

Search time is an important evaluation criteria for an instant photo search system. To obtain the practical search time, we design ten tasks for each volunteer. For examples in Figure 3 (d) , we may ask them to search "red flower," "a girl with a cat," and so on. Once volunteers input a query to search for a photo, our system returns the top 100 most related images in the backend (called "initial search"). Volunteers can re-organize their queries based on query suggestions to conduct the progressive search. we record the time duration from the beginning of entering queries to the moment of finding the desired photos. From Figure 5, we can observe that 91.3% photos could be found by using the proposed system, and about 80% photos could be retrieved in less than 10s. In our study, it usually costs more than 100s to find a specific photo by browsing about 1,000 photos one by one on a mobile phone.

4.6 Evaluation of Usability

Volunteers are invited to try several query suggestion services by using the same ten queries. For each query, they need to answer the question "which can provide the best query suggestions and user experience?" Table 4 shows some suggestions generated by different systems, and Figure 7 shows the evaluation results. We find that

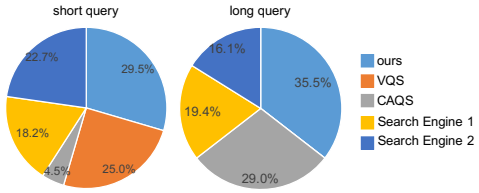


Figure 7: Comparisons between different query suggestion systems and search engines. We omit VQS in the right, as it cannot handle long queries.

Table 5: The computational time (ms) on each module and peak memory cost (MB) of the proposed system on clients.

Device ID	I	II
Image Embedding/Hashing (per image)	152	183
Query Embedding/Hashing (per query)	52	60
Initial Search	40	45
Progressive Search	0.40	0.45
Peak Memory	73.0	

the proposed VQS system outperforms existing image search engines and research prototypes, on both short or long queries. The superior results show the advantages of the effectiveness of query understanding by GRU models, and semantic-preserved hashing embedding learning, as well as the informative concept tree.

5 CONCLUSIONS

In this paper we investigated the possibility of designing fully-client systems for instant visual query suggestion (VQS) and photo search on the phone. We leverage VQS to significantly reduce user efforts and help formulate queries with clarified search intents. Then we use light-weight convolutional and sequential deep neural networks to extract representative features for both visual photos and arbitrary queries. Joint text-image binary search is used to enable the instant VQS and photo discovery. Ultimately we conduct extensive experimental comparisons and user studies, which prove a better user experience. In the future, we will conduct the research on two directions. First, we plan to conduct further research on deep model compression with client-side optimization in terms of both system latency and memory cost. Second, we will focus on highly-efficient hash code indexing and search algorithms to further support the instant query suggestion and photo search.

6 ACKNOWLEDGEMENT

The work is partially supported by NSF of China under Grant 61672548, U1611461, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

REFERENCES

[1] Michael M. Bronstein, Alexander M. Bronstein, Fabrice Michel, and Nikos Paragios. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing.. In *CVPR*. 3594–3601.

[2] Liangliang Cao, Jiebo Luo, and Thomas S Huang. 2008. Annotating photo collections by label propagation according to multiple similarity cues. In *ACM Multimedia*. 121–130.

[3] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S. Yu. 2016. Deep Visual-Semantic Hashing for Cross-Modal Retrieval.. In *KDD*. 1445–1454.

[4] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. 2016. Deep Quantization Network for Efficient Image Retrieval.. In *AAAI*. 3457–3463.

[5] Yan Ying Chen, Winston H. Hsu, and Hong Yuan Mark Liao. 2012. Discovering informative social subgraphs and predicting pairwise relationships from group photos. In *ACM Multimedia*. 669–678.

[6] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Computer Science* (2014).

[7] Matthew Cooper, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox. 2003. Temporal event clustering for digital photo collections. In *ACM Multimedia*. 364–373.

[8] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2003. Query Expansion by Mining User Logs. *TKDE* 15, 4 (2003), 829–839.

[9] Jia Deng, Wei Dong, R. Socher, Li Jia Li, Kai Li, and Fei Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[10] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. 2010. Every Picture Tells a Story: Generating Sentences from Images. In *ECCV*. 15–29.

[11] Fangxiang Feng, Xiaojie Wang, and Ruifan Li. 2014. Cross-modal Retrieval with Correspondence Autoencoder.. In *ACM Multimedia*. 7–16.

[12] Jianlong Fu, Tao Mei, Kuiyuan Yang, Hanqing Lu, and Yong Rui. 2015. Tagging Personal Photos with Transfer Deep Learning. In *WWW*. 344–354.

[13] Jianlong Fu, Jinqiao Wang, Yong Rui, Xin-Jing Wang, Tao Mei, and Hanqing Lu. 2015. Image Tag Refinement with View-Dependent Concept Representations. *IEEE T-CSVT* 25, 28 (2015), 1409–1422.

[14] Jianlong Fu, Yue Wu, Tao Mei, Jinqiao Wang, Hanqing Lu, and Yong Rui. 2015. Relaxing from Vocabulary: Robust Weakly-Supervised Deep Learning for Vocabulary-Free Image Tagging. In *ICCV*.

[15] Alex Graves. 1997. Long Short-Term Memory. *Neural Computation* 9, 8, 1735–1780.

[16] Amarnath Gupta and Ramesh Jain. 2013. Social life networks: a multimedia problem?. In *ACM Multimedia*. 203–212.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[18] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural Language Object Retrieval. In *CVPR*. 4555–4564.

[19] Yao Hu, Zhongming Jin, Hongyi Ren, Deng Cai, and Xiaofei He. 2014. Iterative Multi-View Hashing for Cross Media Indexing.. In *ACM Multimedia*. 527–536.

[20] Di Jiang, Kenneth Wai-Ting Leung, Jan Vosecky, and Wilfred Ng. 2014. Personalized Query Suggestion With Diversity Awareness. In *ICDE*. 400–411.

[21] Xin Jin, Jiebo Luo, Jie Yu, Gang Wang, Dhiraj Joshi, and Jiawei Han. 2010. iRIN:image retrieval in image-rich information networks. In *WWW*. 1261–1264.

[22] Andrej Karpathy and Fei Fei Li. 2015. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *CVPR*. 3128–3137.

[23] Yong Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. 2015. Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. *Computer Science* 71, 2 (2015), 576–584.

[24] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539* (2014).

[25] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. In *NIPS*. 3294–3302.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*. 1097–1105.

[27] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. 2013. BabyTalk: Understanding and Generating Simple Image Descriptions. In *CVPR*. 1601–1608.

[28] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*. 740–755.

[29] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. 2015. Semantics-preserving hashing for cross-view retrieval. In *CVPR*. 3864–3872.

[30] Yu Liu, Jianlong Fu, Tao Mei, and Changwen Chen. 2017. Let Your Photos Talk: Generating Narrative Paragraph for Photo Stream via Bidirectional Attention Recurrent Neural Networks. In *AAAI*. 1445–1452.

[31] Hao Ma, Michael R. Lyu, and Irwin King. 2010. Diversifying Query Suggestion Results.. In *AAAI*.

[32] Shie Mannor, Dori Peleg, and Reuven Rubinstein. 2005. The cross entropy method for classification. In *ICML*. 561–568.

[33] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. 2014. Explain Images with Multimodal Recurrent Neural Networks. *arXiv preprint arXiv:1410.1090*.

[34] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. 2008. Query Suggestion Using Hitting Time. In *CIKM*. 469–478.

[35] Tao Mei, Yong Rui, Shipeng Li, and Qi Tian. 2014. Multimedia search reranking: A literature survey. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 38.

[36] Liqiang Nie, Shuicheng Yan, Meng Wang, Richang Hong, and Tat-Seng Chua. 2012. Harvesting visual concepts for image search with complex queries. In *ACM Multimedia*. 59–68.

[37] Scott Reed, Zeynep Akata, Bernt Schiele, and Honglak Lee. 2016. Learning Deep Representations of Fine-Grained Visual Descriptions. In *CVPR*. 49–58.

[38] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*. 1409–1556.

[39] Pinaki Sinha, Sharad Mehrotra, and Ramesh Jain. 2011. Effective summarization of large collections of personal photos. In *WWW*. 127–128.

[40] Richard Socher. 2014. Grounded compositional semantics for finding and describing images with sentences. *TACL* 2, 207–218.

[41] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources.. In *SIGMOD Conference*. 785–796.

[42] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *CIKM*. 553–562.

[43] 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 *CVPR*. 1–9.

[44] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*. 3156–3164.

[45] Jingwen Wang, Jianlong Fu, Yong Xu, and Tao Mei. 2016. Beyond Object Recognition: Visual Sentiment Analysis with Deep Coupled Adjective and Noun Neural Networks.. In *IJCAI*. 3484–3490.

[46] Ying Wei, Yangqiu Song, Yi Zhen, Bo Liu, and Qiang Yang. 2014. Scalable heterogeneous translated hashing.. In *KDD*. 791–800.

[47] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. 2015. Quantized Correlation Hashing for Fast Cross-Modal Search.. In *IJCAI*. 3946–3952.

[48] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*. 2048–2057.

[49] Ting Yao, Fuchen Long, Tao Mei, and Yong Rui. 2016. Deep Semantic-Preserving and Ranking-Based Hashing for Image Retrieval. In *IJCAI*. 3931–3937.

[50] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL* 2, 67–78.

[51] Zhou Yu, Fei Wu, Yi Yang, Qi Tian, Jiebo Luo, and Yueting Zhuang. 2014. Discriminative coupled dictionary hashing for fast cross-media retrieval.. In *SIGIR*. 395–404.

[52] Zheng-Jun Zha, Linjun Yang, Tao Mei, Meng Wang, and Zengfu Wang. 2009. Visual Query Suggestion. In *ACM Multimedia*. 15–24.

[53] Zheng-Jun Zha, Linjun Yang, Tao Mei, Meng Wang, Zengfu Wang, Tat-Seng Chua, and Xian-Sheng Hua. 2010. Visual query suggestion: Towards capturing user intent in internet image search. *TOMCCAP* 6, 3 (2010).

[54] Fang Zhao, Y. Huang, L. Wang, and Tieniu Tan. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*. 1556–1564.