Personalized Annotation for Photos with Visual Instance Search

Bao Truong, Thuyen Phan, Vinh-Tiep Nguyen, and Minh-Triet Tran

Faculty of Information Technology
University of Science, VNU - HCM
{pvthuyen, tmbao}@apcs.vn, {nvtiep, tmtriet}@fit.hcmus.edu.vn

1 Introduction

In our lives, there are many emotional and memorable moments that worth keeping and sharing with others. Therefore, services allowing users to upload and share personal photos are always one of many notable products of different companies such as Facebook, Flickr, Instagram, and Google Photos. This shows that sharing photos is one of greatest demand of users on the Internet. These services also allow users to attach some memos to their photos as well as to search their photos more easily using text queries.

Currently, the most common way for user to do so is to tag their photos manually which consumes a lot of time and effort. There are also some proposed methods [3, 5] and smart systems which are able to automatically identify noticeable landmarks or location related to the photos such as Google Photos and Flickr. However, these automated annotation are identical for all users and thus do not reflect one's own memories, feelings or characteristics. For example, these systems would recommend phrases like "Eiffel Tower", "a dog", or "a cat" rather than "where I first met my lover" or the name of your pet. Therefore, it is necessary for a system to automatically tag users' photos with personalized caption corresponding to their personal features.

In this paper, we propose a system that can suggest appropriate annotations for each photo uploaded by users using Visual Instance Search. In our system, users can assign their personalized annotations for some photos as initial examples then, the system will automatically propagate these annotations to other existed photos in their collection based on the visual similarities among the photos. For each uploaded photo, the system bases on the visual similarities between the uploaded photo and already-annotated photos of the corresponding users to identify a list of suitable annotations for the uploaded photo in the descending order of the similarities. Then, users can choose to approve reasonable annotation for the uploaded photo. In addition, if a user upload more than one photos and change the annotations, the system will have more samples to reference from and thus, it will tend to better adapt to the users' interests. As a result, our system is not only able to recommend proper annotations which are unique for each user but also to interactively learn and adapt as users change the annotations.

Since the problem of retrieving similar images in a collection corresponding to a single image has been developed for years, there are many different approaches to the problem. One of them is template matching method, i.e. a technique for finding small parts of an image which match a template image [2, 6, 4]. Another popular technique is to evaluate the similarity of two images by comparing some regions which appear to be critical parts of the images, namely features matching [1, 7, 8]. The authors develop our own Visual Instance Search framework using Bag-of-Words model. In Bag-of-Words model, each image is represented as a histograms of pre-trained visual words (codebook). Since Bag-of-Words allows parts of a query image to appear flexible in the result images, it is a potential approach that is widely used in many Visual Search systems.

Together with the exponential increasing of the number of uploaded images, the system faces lots of difficulty adapting those new images. Since re-training the codebook requires changing Bag-of-Words vector of users' existing images and is also computationally expensive, the authors propose to use a fixed codebook trained with different types of objects (e.g. cars, dogs, cats, buildings...) and use it universally. Because of the varieties of those different images, it is appropriate to compute and represent any new images' Bag-of-Words vectors without changing the codebook. We trained our codebook using ABC dataset and tested our system on XYZ dataset. Our performed experiments show that ...

Our main contributions in this paper are as follows:

First We propose the idea and realize the system that can recommend annotation for photos with visual instance search.

Second Our system allows recommended annotation to be personalized and to vary from user to user.

Third Our system is interactively user adaptive, i.e. the more a user annotates his/her photos via our system, the more accurate the recommended annotations are.

The rest of this paper is organized as follows. In section **??**, we review the background and related words in image retrieval and image classification. The core steps of the BoW model and how we conduct experiments are presented in section **??**. Section IV shows experiment results and evaluations. The conclusion and future works are discussed in section V.

2 Proposed System

2.1 How our system learn and automatically annotate new photos

Figure 1 illustrates the overview of our proposed system to automatically recommend personalized annotations for newly uploaded photos. Firstly, users simply use their smartphones camera to capture scenes or objects in real life such as books, dogs or buildings. These photos are then sent to our server for processing and our server will return the list of visual similar photos. Additionally, each photo attaches a list of annotations and these possible personalized annotations will be re-ranked and sent to the users. The users will have chances to review and approve these personalized annotations before sharing these photos to social networks such as Facebook, Flickr, or Google Plus along with the approved personalized tags.

Figure 2 shows how our system learns to annotate photo from samples provided by users. Firstly, users manually choose suitable tags for some photos and these photos along with the tags are then sent to the server. Subsequently, our server process will identify and recommend the users to also apply these changes to visual similar photos in their albums. The users will have the rights to approve before these changes take effect in the database. From this point of time, our system will automatically annotate new photos based on these new configurations.

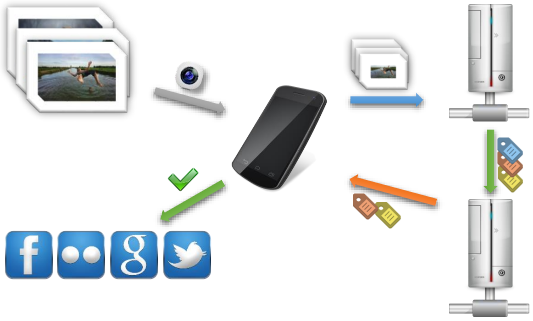

Fig. 1. Overview of our proposed system to automatically recommend personalized tags.

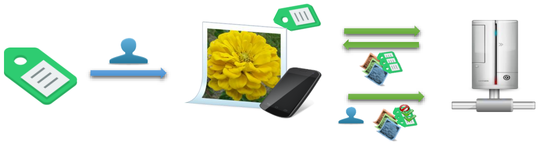

Fig. 2. Overview on how our system learn to annotate photo from samples provided by users.

2.2 Visual Instance Search Method

Feature Extraction To detect and extract features from images, there are many methods that have been proposed (Harris-Affine, Hessian-Affine detectors

[?], Maximally stable extremal region (MSER) detector [?], Edge-based region detector [?], Intensity extrema-based region detector [?] ...). The authors choose to use Hessian-Affine detector, for detecting and extracting features from images. By using Hessian-Affine detector, which is also used in other baseline methods, our experiment's result can easily be compared with other ones.

As we tested on Oxford Building 5K Dataset [?], there are typically 3,300 features for each image and a total about 16 millions of features for the whole dataset. Then, we compute the SIFT descriptor [?] of all the features and these descriptors is used for matching images in the next step.

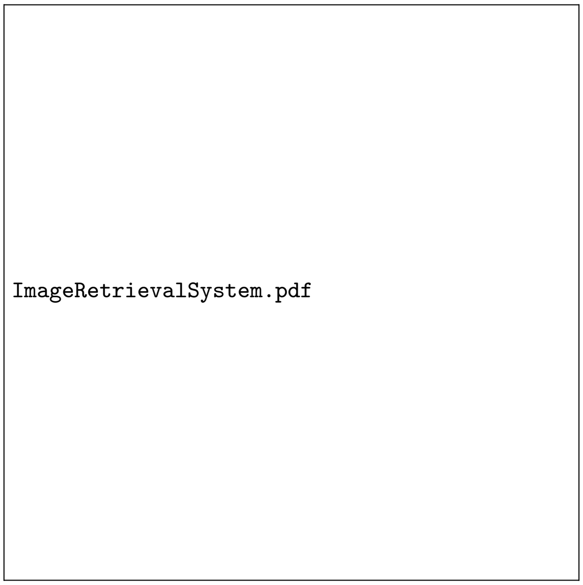

Fig. 3. How an Image Retrieval System works

Dictionary Building Treating each descriptor as an individual visual words in the dictionary results in a worthless waste of resources and time. In order to overcome this obstacle, the authors therefore build the dictionary by considering some similar descriptors as one. In other words, all descriptor vectors are divided into k clusters, each representing a visual word. There are many algorithms that are proposed to solve this kind of problem. However, the authors use the approximate k-means (AKM). AKM is proposed by Philbin et al. [?]. Comparing to the original k-means, AKM can reduce the majority amount of time taken by exact nearest neighbors computation but only gives slightly different result. Also, in [?], Philbin et al. shows that using 1M dictionary size would have the best performance on the Oxford Building 5K Dataset [?].

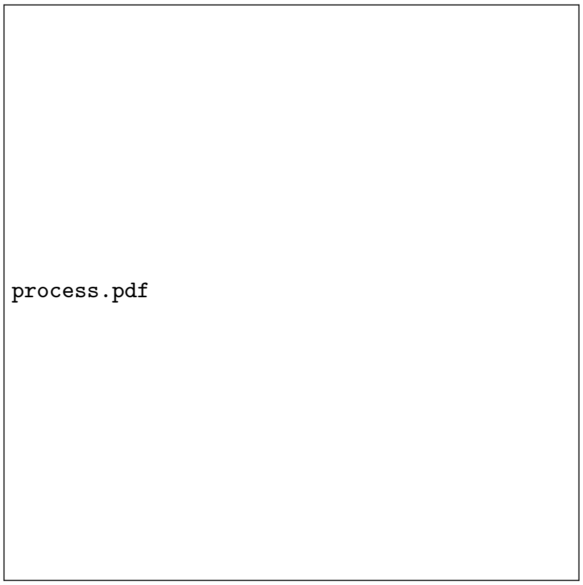

Fig. 4. Proposed framework

Quantization Subsequently, each 128-dimension SIFT descriptor needs to be mapped into the dictionary. Commonly, each descriptor is assigned into the nearest word in the dictionary. Thus, when two descriptors are assigned to diferent words, they are considered as totally different. In practice, this hard assignment leads to errors due to variability in descriptor (e.g. image noise, varying scene illumination, instability in the feature detection process ...) [**?**]. In order to handling this problem, the authors use soft assignment instead of hard assignment. In particular, each 128-dimension SIFT descriptor is reduced to a k-dimension vector of their k nearest visual words in the dictionary. Each of these k nearest cluster is assigned with weights calculated from the formula proposed by Sivic et al. [**?**], $weight = \exp(-\frac{d^2}{2\delta^2})$, where d is the distance from the cluster center to descriptor point. Then, by adding all these weights to their corresponding bins, we will have the BoW representation of an image.

In this work, k and δ^2 are chosen to be 3 and 6250, respectively.

tf-idf Weighting Scheme As mentioned in section **??**, tf-idf is a popular weighting scheme that is used by almost any BoW model. In this section, the authors will show how this scheme is applied to our system.

For a term t_i in a particular document d_j, its term frequency $tf_{i,j}$ is defined as follow:

$$tf_{i,j} = \frac{n_{i,j}}{\sum\limits_k n_{k,j}} \tag{1}$$

Where $n_{i,j}$ is the number of occurrences of the considered term t_i in the document d_j. The denominator is the sum of the number of occurrences of all the terms in document d_j.

The inverse document frequency idf_i of a term t_i is computed by the following formula:

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2}$$

Where, $|D|$ is the total number of documents in the corpus, $|\{j : t_i \in d_j\}|$ is the number of documents where the term t_i appears, i.e. $n_{i,j} \neq 0$

The tf-idf weight of a term t_i in a document d_j is then calculated as the product of tf and idf:

$$tfidf_{i,j} = tfi, j \times idf_i \tag{3}$$

The tf-idf weight is then used to compute the similarity score between an image d_i and a query q:

$$s_{d_i,q} = \boldsymbol{tfidf_i} \cdot \boldsymbol{tfidf_q} = \sum_{j=1}^{|T|} tfidf_{i,j} \times tfidf_{q,j} \tag{4}$$

Finally, by sorting the list of images corresponding to their similarity score with a query, we achieve the raw ranked list of this query which is then used for the Spatial Rerank step.

References

1. Belongie, S., Carson, C., Greenspan, H., Malik, J.: Color- and texture-based image segmentation using em and its application to content-based image retrieval. In: Computer Vision, 1998. Sixth International Conference on. pp. 675–682 (Jan 1998)
2. Brunelli, R.: Template Matching Techniques in Computer Vision: Theory and Practice. Wiley (2009)
3. Chen, M., Zheng, A., Weinberger, K.Q.: Fast image tagging. In: Dasgupta, S., Mcallester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning (ICML-13). vol. 28, pp. 1274–1282. JMLR Workshop and Conference Proceedings (May 2013), http://jmlr.org/proceedings/papers/v28/chen13e.pdf
4. Gharavi-Alkhansari, M.: A fast globally optimal algorithm for template matching using low-resolution pruning. Image Processing, IEEE Transactions on 10(4), 526–533 (Apr 2001)
5. Lan, T., Mori, G.: A max-margin riffled independence model for image tag ranking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2013)
6. Rosenfeld, A., Vanderburg, G.J.: Coarse-fine template matching. Systems, Man and Cybernetics, IEEE Transactions on 7(2), 104–107 (Feb 1977)
7. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40(2), 99–121 (2000)

8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. vol. 1, pp. I–511–I–518 vol.1 (2001)