Automatic Image Tagging Using Community-Driven Online Image Databases

Marius Renn, Joost van Beusekom,
Daniel Keysers, and Thomas M. Breuel

IUPR Group, Technical University of Kaiserslautern, Germany
m_renn@informatik.uni-kl.de, joost@iupr.dfki.de,
keysers@iupr.com, tmb@informatik.uni-kl.de
http://www.iupr.org

Abstract. Automatic image tagging is becoming increasingly important to organize large amounts of image data. To identify concepts in images, these tagging systems rely on large sets of annotated image training sets. In this work we analyze image sets taken from online community-driven image databases, such as Flickr, for use in concept identification. Real-world performance is measured using our flexible tagging system, *Tagr*.

1 Introduction

With the rise of the internet and the rapid growth of storage, a number of large image databases have emerged on the web. Fueled by the popularity of low-cost digital image capturing devices and the web 2.0 trend of dynamic community driven websites, many of these databases consist of photos submitted by community members. As our observations using the Flickr API[1] show, these websites may see growth rates of over one million photo submissions per day (see Figure 1). Likewise, offline personal photo collections now often contain thousands of photos, that can be stored and viewed on high resolution computer screens at nearly zero cost.

Due to this rapid growth of image content both on- and offline, it has become increasingly difficult to organize these massive amounts of visual data. To overcome this difficulty, many photo sharing websites and modern offline photo organizing software applications allow the user to add textual annotations to the images. These annotations usually consist of a list of keywords or *tags*, that describe some aspect of the image *content*, and allow organizing, searching and filtering images, using algorithms based on these keywords. Unfortunately, a great deal of images both on- and offline exist, that have no textual representation whatsoever. Asking humans to manually label such images is not only costly and time-consuming, but also poses privacy and security issues. Furthermore, the almost exponential growth of photos on community websites would require an ever growing team of labelers. Thus it is desirable to add missing tags to images automatically.

[1] http://www.flickr.com/services/api/

M. Detyniecki, U. Leiner, and A. Nürnberger (Eds.): AMR 2008, LNCS 5811, pp. 112–126, 2010.

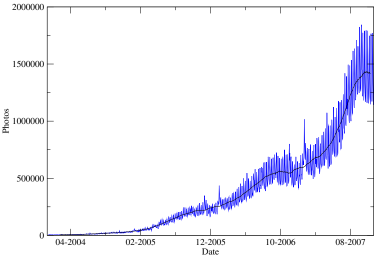

Fig. 1. The number of photos uploaded to Flickr measured on a daily basis from January 2004 until November 2007, using the Flickr API

Although computers are still a long way from identifying and textually describing image concepts in the way humans do, it is possible to train computers on large previously annotated image databases, in order to learn the associations between visual image data and their textual descriptions. Now that large online image databases are available free of charge, these provide an interesting alternative to professionally labeled commercial sets. Furthermore, many of these community-driven image collections are annotated with tags, submitted by the users.

In this work we make use of large online image databases, such as Flickr[2], and compare their performance to a professionally annotated image set. Our analysis will show where potential problems lie, and what can be done to overcome them. To evaluate performance we use our flexible automatic image tagging system, called *Tagr*.

In Section 2 we will give an overview of related literature. Section 3 describes the tagging system on which the measurements are based. The image sets are described in Section 4. Section 5 discusses the results. The paper is concluded by Section 6.

2 Related Work

In this work, our focus lies on the compared performance between commercial image sets and online community-driven databases as models for image tagging. To our knowledge, there has been no prior work on such a performance analysis for automatic image tagging. However, the development of image tagging itself has become a subject of interest in recent years, and numerous approaches to accomplish this task have emerged. We shall give a brief outline here. One popular method of automatic annotation is the association of keywords with image regions. These image regions can be extracted using common image segmentation techniques, as in [7], where a grid-based segmentation method to identify image regions is used, or using clustering methods as in [1], [2] and [5]. In [6] Li and Wang present a real-time automatic image annotation system called ALIPR,

[2] http://www.flickr.com

that uses advanced statistical modeling and optimization techniques to train computers various semantic concepts using example pictures. Their system is evaluated on manually selected images obtained from Flickr. Other systems, like the one used here, base annotations on the entire image scene, using global image features. In [12] nonparametric density estimation of global image features is used as a model for keyword probabilities. The Corel and Getty datasets are used for performance evaluation. As in our system, the authors in [11] describe an image tagging system, that uses content-based image retrieval at its core. Users interact with the system to produce correct high-level metadata. In this work we will use our own scene-based tagging system, called *Tagr*, to investigate the use of various annotated image sets as models for keyword probabilities. The high flexibility, speed, and access to the source code led to the choice of *Tagr* for these experiments.

3 System Overview

We now give a brief overview of *Tagr*, the tagging system used for our experiments. To meet our requirements of tagging a broad class of images while offering flexibility in the methods used, *Tagr* combines techniques from three, not strictly separate, domains of pattern recognition and machine learning:

- Hard-Coded Rules: *Tagr* uses simple rule-based image analysis to generate higher-level metadata, such as image dimensions, dominant colors, and image format.
- Machine Learning: To assign images to a certain category (such as `graphic` or `photo`), *Tagr* uses well-known classifiers on extracted image features.
- Content-Based Image Retrieval: In content-based image retrieval (CBIR), images in a database are searched by their content, and not by their textual representation (i.e. filename). *Tagr* uses CBIR to find the k closest matches of some query image in an image database (k-nearest-neighbor search), and requires some definition of a distance metric between the images or image features. *Tagr* requires the images in the database to have been previously supplied with tags, that each describe the image content. This way, *Tagr* is able to extract a set of tags for a given query image, by analyzing the most frequent tags among the nearest neighbors of the database.

At the heart of the classification and tagging process lies a *query tree*, in which an image query is handed to the root node and passed down to the children for further processing. Nodes may apply filters to the image, or process it in any other way. The leaf nodes are responsible for creating an initial textual annotation of the image. Most often we make use of the `FireNode`, which sends the image query to the *Flexible Image Retrieval* (short: *FIRE*) system for nearest neighbor comparison. *FIRE* is an image retrieval system originally developed by Thomas Deselaers of the RWTH Aachen, and now maintained by him and Daniel Kaisers. Interested readers can find more information about it in [4]. For a given query image, *FIRE* returns the k nearest neighbors of this image over a database of model images, given a set of features and distance metrics for comparison. These images are returned as a scored list. In the query tree, the FIRE node simply sends the query image to a specified running FIRE server, and passes the result list back up the tree. This image list is then converted to a list of tags by the `ImagesToTagsNode`, which maps each image file to its textual description. The resulting tags obtain the scores of the images they resulted from. Usually this list is passed into a `PackResultsNode`, which combines equally named result strings by summing their scores.

3.1 Feature Selection

FIRE offers a variety of feature extraction methods and distance measures to be used in the nearest neighbor search. In the most basic configuration, we employ color histograms and Tamura texture [10] features, and the Jensen-Shannon divergence as a similarity measure. However, the flexibility of FIRE and the query tree approach allow us to test many other configurations of features and comparison methods. Although color and texture histograms produce convincing results, their downside is the loss of all spatial information. Therefore, a number of spatial features were tested, including various configurations of spatiograms [3]. The best results however were obtained using *weighted* histograms: These make use of the observation, that important classifying aspects of a photo usually lie in distinct regions on the image plane. For instance, the subject of a photo usually lies approximately in the center of the image. Color information close to the photo's boundaries, on the other hand, often shows other concepts, such as the ground or sky. To capture these distinct areas, we use a set of weighted histograms, one for each region in the image. In [9] a similar approach, called *fuzzy regions*, is used, where the image is subdivided into 5 regions. In our case, we use 3 weighted histograms to represent the top, center and bottom regions of the image.

4 Image and Tag Sets

Large pre-annotated image sets not only provide the model for the k-nearest neighbor search, but are also of use as test sets for evaluation. The given tag data of an image is compared to the tags returned by the system, which allows us to measure performance in terms of precision and recall. Table 1 gives an overview of the image sets used, along with their size (in number of images), how they were aggregated, and the most frequent tags.

4.1 COREL Set

To compare results of imagery from community-driven websites to those of commercial image sets, we randomly selected $26,803$ textually annotated photos from the Corel database, as a commercial representative. Each photo is tagged by 4 keywords on average, and there are a total of $4,900$ unique tags in the database. The main advantage of the Corel set is the fact, that the photos are professionally annotated, and thus exhibit consistency and objectivity.

Table 1. The annotated image sets used for our experiments

	COREL	Flickr	FotoCommunity	LabelMe
Images	26,803	52,478	20,834	32,025
Aggregation	commercially available	Flickr API	crawled website	download available
Top Tags	sky, water, people, trees, building	wedding, 2007, beach, nature, sky	motives, nature, people	car, head, tree, window, building

4.2 Flickr Set

Two prominent examples of online photo communities are Google's Picasa and Yahoo's Flickr, that provide millions of publicly available tagged photos. In this work, we chose to use Flickr's database, as an API for a variety of host languages is available. Using this API for Python, we implemented a number of tools, that allowed us to access the photo and tag data from the Flickr database. This was especially useful to accomplish the following tasks:

- Extraction: Given a number n, and a set of input tags T, download up to n photos and tags, that are annotated with the tags T', where $T \subseteq T'$.
- Tagging: As each image file from Flickr has a unique name, tag data can be added to previously downloaded photo sets.
- Sampling: Download random tags to sample keyword frequency.

Using these tools, first a set of typical tags was extracted from the Flickr database. These tags were analyzed to obtain a set of common topics, some of which are shown in Table 2. Using these topics as tag search words, a set of $52,478$ photos along with their full set of user annotations were downloaded from Flickr. The advantages of extracting photos from a large online database are the flexibility in quantity and scope. For instance, in order to compare results on the Corel set to a similar set, the Flickr API was utilized to download an image database with approximately the same size and same tags as the Corel set. This allowed us to test classifiers, that were trained on the Corel set, on a similarly labeled set of different photos.

Table 2. An excerpt from the categories (top row) used to download Flickr image sets. These categories were evaluated by looking at common tags and subjects depicted in the photos.

animal	event	food	nature	people	sports
cat	concert	fruit	beach	face	golf
dog	party	vegetable	forest	person	hockey
farm	wedding	fastfood	sky	portrait	tennis

4.3 FotoCommunity Set

A different approach to annotated image aggregation was taken, by crawling the German online community site `fotocommunity.de`. This website's intended contributors are amateur and professional photographers, who can get help and tips from other photography enthusiasts. Photos tend to be more professional and artistic than on Flickr. Although the photos are not tagged, they have been categorized (by the site's maintainers) well enough, that the category hierarchies themselves can be used as textual annotations. The category tree, downloaded from the website by our web-crawler, has over 950 nodes, with a total of 740 leaf nodes (most detailed categories). A set of 20, 834 photos distributed among all categories was downloaded for the tagging model.

4.4 LabelMe Set

Finally, we gathered image collections from websites, that also focus on textually annotating image data. The *LabelMe* project[3] from the Massachusetts Institute of Technology aims to collect contributions from many people to build a large high quality database for research on object recognition. Instead of merely supplementing given images with keywords, users trace the boundaries of objects in images, and add labels to these regions. Each time an object is labeled, the data is continuously saved and made immediately available to interested researchers[4]. A total of 32, 025 labeled images were downloaded. The most common object tags with frequencies of at least 100 occurrences were extracted and assigned to the images they occurred in. The segmentation information itself was ignored. This resulted in a list of 116 distinct tags. Images that did not contain objects with these tags were filtered out. It should be noted, that most images in this set display an inner-city street scene, so that the use of this set is rather limited.

4.5 Tag Selection

As we intend to label a broad domain of images, we needed to make sure that the image sets showed enough concepts to cover a wide range of topics. In this respect, the Flickr database allows for a greater flexibility, than the pre-annotated Corel set. However, the Flickr tags also proved to be more problematic than those of the professionally labeled set: Many of the tags are subjective (such as `wow`, `myfav`, or `top10`), over-detailed (name of depicted person), or do not describe the contents of the image (name of the author, group or collection). Furthermore, the level of abstraction is far from consistent, with some photos being tagged very detailed, and others very generalized. This is not only apparent in the number of tags used, but also in the words chosen (i.e. `llama` vs. `animal`, or `manhattan` vs. `city`). Frequent subjective tags were therefore filtered, and common over-specific tags generalized. However due to the large number of unique

[3] http://labelme.csail.mit.edu/

[4] http://labelme.csail.mit.edu/guidelines.html

tags $(33,967)$, such mechanisms could only weaken the problems to some extent, and not eliminate them. Unfortunately, simply filtering the tag set to very frequent tags only, often removed important concepts from images altogether.

Furthermore, the photo sets downloaded from community websites naturally reflect subjects that are popular among the community. These tend to be more personal, than the photo sets found on research websites or in commercial sets. For instance, the most popular Flickr tag (at the time of writing) is *wedding*, which is probably due to the fact that many photos are taken during a wedding ceremony. However, it is very unlikely that such an emphasis on wedding photos applies to other image collections, such as those found on a news site or a blog. For this reason, the concepts used as search terms for image aggregation did not reflect the top tags of the website they were downloaded from. Instead, a subset of the most popular tags was manually selected to reflect a broader range of topics. Furthermore, the actual distribution of photos over these categories was neglected, and a more uniform distribution chosen instead. While these measures helped create a more objective categorization, it should be noted that the photo contents themselves still tend to be much more personalized than those found on general websites. For instance, even though we did not explicitly download wedding photos in some of the photo sets, this keyword was still among the most frequent in all downloaded sets from Flickr.

The fotocommunity set, on the other hand, has a relatively small number of distinct categories, and thus better avoids the problem of over-detailed tags. Also, as the category tree seems to be maintained by a few people only (presumably the site's administrators), the descriptions are clear, objective and consistent. However, unlike the other sets, here every image is described by a single concept only. Photos that depict more than one concept are thus only partially described by their tags.

5 Experiments and Results

In the most basic test, we classify and tag each photo using a FIRE server configured to return the 5 nearest neighbor images using color- and texture-histogram features, and the Jensen-Shannon Divergence. These image lists are converted to a list of tags, and the top 4 tags make up the result. The test results are displayed in Figure 2 (left) for the Corel and Flickr image sets, using leave-one-out classification on both sets. Performance is given by the following measures:

- *Mean Precision*: The mean tag precision, i.e. $\frac{|\{\text{correct tags}\} \cap \{\text{returned tags}\}|}{|\{\text{returned tags}\}|}$ over all images.
- *Mean Recall*: The mean tag recall, i.e. $\frac{|\{\text{correct tags}\} \cap \{\text{returned tags}\}|}{|\{\text{correct tags}\}|}$ over all images.
- > 0 *Precision*: The portion of results, that had a precision greater than zero (i.e. at least one correct tag).

As the results show, in terms of precision and recall, the Flickr database is very competitive to the Corel set. However, the non-zero precision is much lower in the

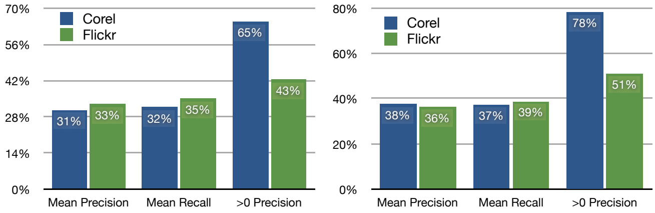

Fig. 2. The tagging results, using color and Tamura texture histograms (left), and weighted histograms (right)

Flickr set, where the majority of images were labeled exclusively with incorrect tags.

Results improve when employing weighted histograms for both the color and texture data. Here, a query tree is used, that merges the result lists of 3 separate FIRE servers - one for each region into one result list. As Figure 2 (right) shows, performance does increase overall, but the non-zero precision gap between the two photo sets remains.

In a slightly more advanced experiment, we adjust the FIRE servers to return 10 nearest neighbors each, and raise the number of tags in the result to 15. The reason for this change, is that such a configuration allows us to analyze the result lists in more detail: By sorting the top tags by their confidence, we can measure the performance of the tagging system for any number of returned tags n ($n \leq 15$), simply by extracting the top n tags only. Also, instead of leave-one-out classification we use disjunct model and test sets for evaluation. Using these parameters and weighted histogram features, we gathered the non-zero precision rates for $5,360$ test images from the Corel dataset. In order to easily match result tags to the true tags in performance measurement, both the test and model sets must use a similar tag dictionary. In this case, we chose a disjunct set of $21,444$ annotated images from the Corel database to act as our model. The results are given in Figure 3. The graph shows, that even when returning just one tag, we obtain a coverage of over 50%. When returning 15 tags, the non-zero precision rate improves to 93%.

The same experiment was repeated for the Flickr set. Here, we downloaded two image sets from the Flickr database, using the same list of search terms (a portion of which is shown in Table 2), but making sure the image sets themselves were disjunct. For every keyword k, a number of images were downloaded, that were tagged with the tag set T_k, and $k \in T_k$. Although this method does not necessarily aggregate images, where k is the main concept, it does guarantee that at least one keyword in each image description occurs in both sets. A total of $26,238$ Flickr images were used in the model set, and $5,247$ images for testing. We tested on a filtered version of the Flickr tags, and on the original tags directly. To filter the Flickr tags, the top tags, with frequencies > 100 were manually reviewed, and those tags removed, that were overly subjective or abstract. As the results in Figures 4 and 5 show, manual tag preprocessing is essential to

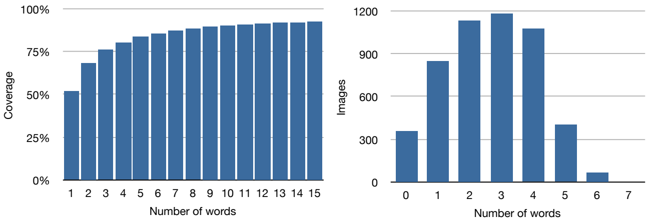

Fig. 3. Tagging performance on a subset of the Corel database (5360 images). The graph on the left shows the percentage of images that are tagged with at least one correct tag, when the top n words are returned. The graph on the right is a histogram of the number of correct words for each image.

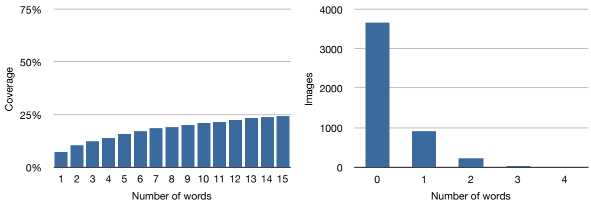

Fig. 4. Using an unfiltered Flickr tag set directly for tag evaluation shows very poor performance

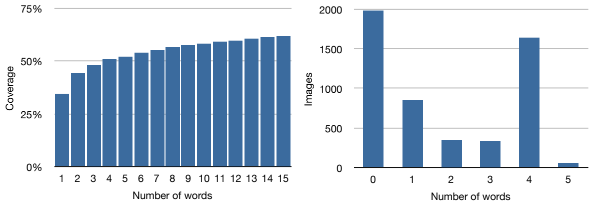

Fig. 5. Tagging performance on the filtered Flickr image set (5247 images). Here, we reach a maximum non-zero precision rate of 62%, and a 34% coverage when returning just one tag.

acceptable performance. However, even after the tags have been filtered, we fall short of reaching the same accuracy as in the Corel test. There are several possible explanations for the poor performance of the Flickr set. One obvious problem is the high diversity of Flickr tags, that very often describe non-relevant image properties (despite having been somewhat pre-filtered). The question is whether a larger model is able to overcome the difficulty of this diversity.

Thus, to analyze whether the employed model is too small for accurate tagging, we set up FIRE servers with various model sizes, ranging from around 100 images up to around 26, 000 images for the Corel set, and 50, 000 images for the Flickr set. Approximately 500 disjunct images from both sets each were tested against their counterparts. The models for each size were obtained by taking the full image sets and down-sampling them to the desired sizes. Due to time constraints, we only used color and Tamura histogram features for the evaluation. The performance of each test for the two image sets is shown below in Figure 6. While the recall and precision of the Corel test are asymptotically bounded by approximately 38%, the percentage of images with a non-zero precision shows a less smooth curve with a local maximum at a model size of roughly 13,400 images. These results suggest, that a larger model size (than 26,000 images) would not lead to significantly higher accuracy. The Flickr results, on the other hand, show that even model sizes larger than 25,000 images can still lead to significantly higher scores. Thus it may make sense to employ even larger model databases for the Flickr images than used in our tests.

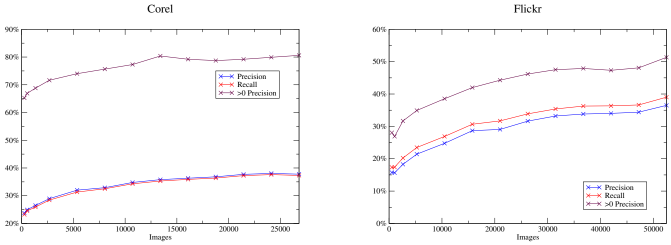

Fig. 6. Approximately 500 images were tested from the Corel and Flickr image set, using FIRE servers with increasing model sizes

Finally, we analyze how the number of images that are returned from the FIRE server affects performance on both sets. This should not be confused with the number of *tags* that make up the result. Recall that each result image from the FIRE server is mapped to a set of tags, where each tag is given the score of the image. Identical tags are merged into one by summing their scores, and the top n tags are extracted to form the result. The more images the FIRE server returns, the more tag sets contribute to the final result list. However, image results that are so distant from the query image, that their score is close to zero, may not be of any relevance to the result at all. By configuring our FIRE servers to return the top k images for various $k \leq 100$, we obtain the results shown in Figure 7. Note that the initial drop of precision is due to the fact that a very low k results in only a small number of returned tags (usually ≤ 4). As we chose to return the top 10 tags in each test, this requires at least $k = 3$ returned images from the FIRE servers. The Corel results show, that roughly $k = 30$ returned images gives an optimal result. The Flickr results on the other hand show a continuing drop in precision, and suggest that a much smaller k leads to overall better results. The explanation for this phenomenon is most likely that popular tags outweigh the correct tags for large k. Figure 8 supports this assumption, showing the percentage of results that contain the popular tag `wedding` for each k. As the result shows, for large k the popular keyword occurs more and more frequently, to the point where over 20% of the images are marked with `wedding`.

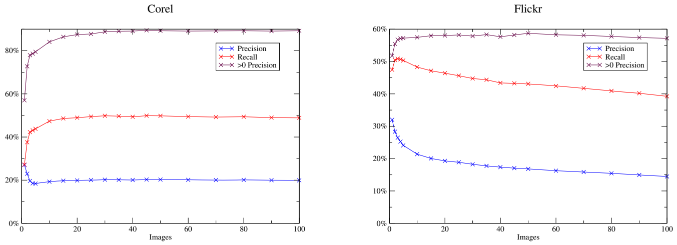

Fig. 7. Again, roughly 500 images were tested from the Corel and Flickr image set, using FIRE servers returning an increasing amount of images

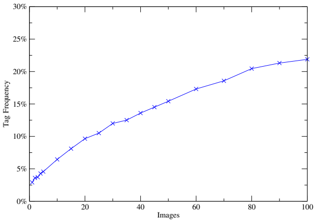

Fig. 8. The frequency of images that were tagged with `wedding` increases with the number of images returned by the FIRE server

Of course, this is far from the actual proportion of our images that depict a wedding ($\approx 1\%$).

To overcome the problem of over-frequent tags, another large Flickr set was downloaded, and repeatedly subsampled, until the tag frequency variance was sufficiently small. However, despite solving the problem of popular tags, overall performance increased only minimally.

At this point, it makes sense to ask whether automatic performance evaluation is suitable for the Flickr image set at all. Simple keyword matching of the result to the true data may simply be too strict for such a large number of distinct tags. Therefore, we manually evaluated the results returned by Tagr. Two configurations of the system were used: One, trained on the Corel set, and another on the Flickr set. A test set of 1000 random Flickr images were sent to each system, and the tag lists returned by the tagging system were manually reviewed. Those tags that clearly described some concept of the image were marked as correct. Figure 9 shows the performance of the Flickr test set on the Corel and Flickr based tagging systems. The results show, that while the Corel system showed performance consistent to the one measured using automatic evaluation, the results for the Flickr model improved greatly. This suggests that Flickr based tagging does indeed find fitting descriptive words for many images. However these descriptions do not match the tags of the ground truth, submitted by the community users. In fact, during this brief evaluation, a number of critical issues were observed:

- The high diversity and ambiguity of Flickr tags became quickly apparent. For instance, images of people are often tagged with `friends`, rather than `people`. Filtering these subjective words out, removes the concept of *people* from the description altogether. On the other hand, replacing occurrences of `friends` with `people` leads to incorrect descriptions for the many animal photos labeled with this tag.
- Occasionally, users annotate whole image sets with the same set of tags. However, usually these tags only apply to a subset of the collection.
- Some of the Flickr photos are greatly distorted or stylized, making an accurate tagging extremely difficult.

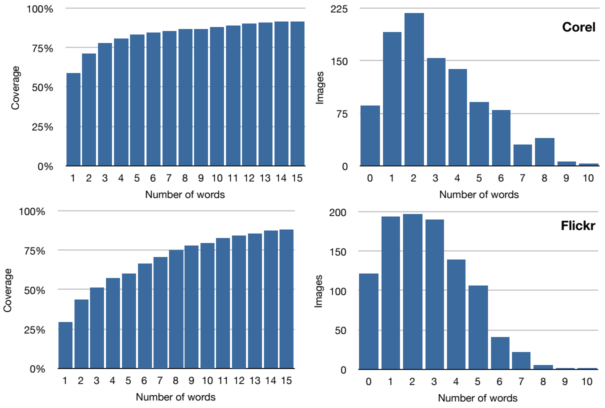

Fig. 9. Roughly 1000 images from the Flickr set were manually evaluated against two models. Tagging performance increased greatly on the Flickr model (bottom), when compared to the results of automatic performance evaluation. Still, the Corel set (top) shows the overall better results.

Using the Corel image set as a model for tagging still provided the overall better results. However, it should be noted, that the Corel tags tended to be much more general than the Flickr tags. Many concepts were missing in the Corel set altogether. High scores were still obtained due to the many general tags, such as `people`, `ground` and `wall`, that occured in 99.3% of the results returned by the Corel based system, and which applied to the majority of images. On the other hand, users on Flickr rarely tag an image with `wall`, even if a wall is depicted, and instead tend to focus on the specifics of the image subject. In this respect, the Flickr set provides an interesting alternative, if more detailed or specialized tags are important. However, the poor performance, most notable when returning less than 4 tags, must be kept in mind.

Overall, these results show some of the challenges involved with using photo sets from community driven web sites. In order to evaluate whether photo sets from other websites show similar results, the (weighted) histogram methods were additionally tested on the remaining sets. Figure 10 shows the results obtained

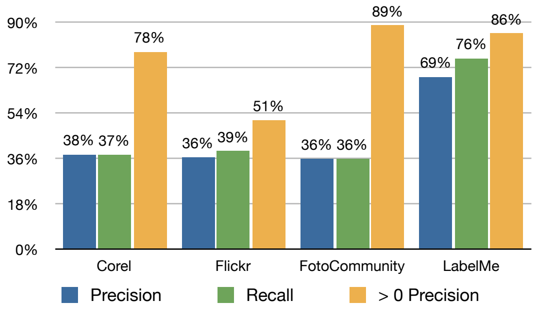

Fig. 10. The precision, recall and non-zero precision for all of the used photo sets

for all of the photo sets, using leave-one-out classification, and 4 result tags. Recall that the FotoCommunity set is not actually tagged. Instead, we used the hierarchical categorization as tags, so that each image is actually tagged with keywords of various levels of abstraction. For instance, as the highest level can only be either `people`, `nature`, or `scenes`, every annotation list contains exactly one of these instances. Note, that this is not necessarily the case for the result list returned by Tagr. These very frequent tags explain the high non-zero precision rate for the FotoCommunity test. Furthermore, as the keywords describe concepts on a much more general level, than for our earlier photo sets, we chose to use simple histogram features instead of weighted histograms for the FotoCommunity set.

The LabelMe set surprises with its very high accuracy. This is most likely due to the small amount of concepts, and therefore keywords (116) in the set. Nevertheless, as these keywords in the LabelMe set are actually descriptions of certain objects within the images, our scene based tagging system performs remarkably well. Overall these results only give a brief glimpse of the performance of other online image sets. However, they do show which image set criteria may lead to improved tagging performance. Like the Corel database, the FotoCommunity set achieves high non-zero precision by including high-level concepts in its list of keywords. The LabelMe set shows that for small image domains, online databases, used as annotation models, can produce very satisfying tagging results.

6 Conclusion

We have shown that it is possible to utilize the internet and its ever-growing community-driven image databases to obtain large annotated image sets, that may be used for automatic image tagging of a broad image domain. Our analysis shows that the direct use of these image sets, without any further filtering or other processing, does not provide satisfying results. Critical issues, such as subjective or non-relevant keyword descriptions, greatly diminish the overall quality of the concept descriptions, and in turn lead to poor results of the tagging system. However, if appropriate measures, such as keyword filtering or uniform keyword sampling, are employed to overcome these drawbacks, the resulting tagging system may in fact be more suitable to an annotation task, than a commercial set. The reason for this is that the immense size of online databases, such as Flickr, allow a much more flexible aggregation of concepts, that can be tuned to the intended domain. In our case, we were able to aggregate a much higher quantity of concepts from online databases, than from the Corel set, which was previously annotated with a fixed set of tags from a fixed domain. Measuring performance on such a vast set of image tags is challenging, and in most cases will require tedious manual evaluation of the tagging results. Even if the tagging system was trained on a carefully selected annotated image set, random test images are most likely tagged with a different set of keywords than those found in the model set. Leave-one-out classification or cross-validation methods may help here, but have the downside of operating exclusively on the preselected imagery, which may differ greatly from the one found in an installed environment. Furthermore, the keyword preprocessing itself may introduce new problems to the tag set. For instance, although filtering unwanted words from the tag lists may be beneficial to some images, in others it may remove important concepts from the description. Ambiguities in such subjective descriptions make a simple replacement of these tags by more appropriate ones difficult.

Although our analysis highlights many important aspects of image tagging using community image databases, this work is far from complete. Many important questions have yet to be answered. For instance, how do the results change when focussing on only a small domain ($2 - 5$ different keywords)? How well do community-driven image sets work on classifying regions of images? How useful are they for classification methods other than nearest-neighbor search? Can performance be increased by incorporating ontologies into the tagging process as proposed in [8]? While these questions open new areas of research, the most important next steps will be the continuing analysis of current results. Most importantly, more user studies on standard data sets must be performed, to evaluate real-world performance of various configurations of the tagging system.

References

1. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research 3, 1107–1135 (2003)
2. Barnard, K., Forsyth, D.: Learning the semantics of words and pictures. In: Eighth International Conference on Computer Vision, vol. 2, p. 408 (2001)
3. Birchfeld, S.T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, June 2005, vol. 2, pp. 1158–1163 (2005)
4. Deselaers, T., Keysers, D., Ney, H.: Fire - flexible image retrieval engine. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 688–698. Springer, Heidelberg (2005)
5. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proceedings of the 26th Intl. ACM SIGIR Conference, vol. 3, pp. 119–126 (2003)

6. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. In: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 911–920 (2006)
7. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: First International Workshop on Multimedia Intelligent Storage and Retrieval Management (1999)
8. Srikanth, M., Varner, J., Bowden, M., Moldovan, D.: Exploiting ontologies for automatic image annotation. In: Proceedings of the 28th annual international ACM SIGIR Conference, pp. 552–558 (2005)
9. Stricker, M., Dimai, A.: Color indexing with weak spatial constraints. In: Proceedings of SPIE, March 1996, vol. 2670, pp. 29–40 (1996)
10. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. IEEE Trans. Systems, Man, and Cybernetics 8, 460–472 (1978)
11. Wenyin, L., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M., Field, B.: Semiautomatic image annotation 01, 326–333 (2001)
12. Yavlinsky, A., Schofield, E., Ruger, S.: Automated image annotation using global features and robust nonparametric density estimation. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 507–517. Springer, Heidelberg (2005)