Baselines for Image Annotation

Ameesh Makadia (makadia@google.com)
Google Research, New York, NY 10011, USA

Vladimir Pavlovic (vladimir@cs.rutgers.edu)
Rutgers University, Piscataway, NJ 08854, USA

Sanjiv Kumar (sanjivk@google.com)
Google Research, New York, NY 10011, USA

Abstract. Automatically assigning keywords to images is of great interest as it allows one to retrieve, index, organize and understand large collections of image data. Many techniques have been proposed for image annotation in the last decade that give reasonable performance on standard datasets. However, most of these works fail to compare their methods with simple baseline techniques to justify the need for complex models and subsequent training. In this work, we introduce a new and simple baseline technique for image annotation that treats annotation as a retrieval problem. The proposed technique utilizes global low-level image features and a simple combination of basic distance measures to find nearest neighbors of a given image. The keywords are then assigned using a greedy label transfer mechanism. The proposed baseline method outperforms the current state-of-the-art methods on two standard and one large Web dataset. We believe that such a baseline measure will provide a strong platform to compare and better understand future annotation techniques.

1. Introduction

Given an input image, the goal of automatic image annotation is to assign a few relevant text keywords to the image that reflect its visual content. With rapidly increasing collections of image data on and off the Web, robust image search and retrieval is fast becoming a critical requirement. Most current Internet image search engines efficiently exploit text-based search to retrieve relevant images, while ignoring image content. Utilizing image content to assign a richer, more relevant set of keywords would allow one to further exploit the fast indexing and retrieval architecture of these search engines for improved image search. This makes the problem of annotating images with relevant text keywords of immense practical interest.

Image annotation is a difficult task for two main reasons: The first is the well-known *pixel-to-predicate* or *semantic gap* problem, which points to the fact that it is hard to extract semantically meaningful entities using just low level image features, e.g. color and texture. The alternative of doing explicit recognition of thousands of objects or classes reliably is currently an unsolved problem. The second difficulty arises due to the absence of *correspondence* between the keywords and image regions in the training data. For each image, one has access to the keywords assigned to the *entire* image and it is not known which regions of

the image correspond to these keywords. This precludes direct learning of classifiers by assigning each keyword to be a separate class. Only recently, techniques have emerged to circumvent the correspondence problem under a discriminative multiple instance learning (Yang et al., 2006) or generative paradigm (Carneiro et al., 2007).

Image annotation has been a topic of ongoing research for more than a decade leading to several interesting techniques (Duygulu et al., 2002; Blei et al., 2003; Jeon et al., 2003; Wang et al., 2004; Lavrenko et al., 2004; Monay and Gatica-Perez, 2003; Feng et al., 2004; Barnard and Johnson, 2005; Metzler and Manmatha, 2005; Hare et al., 2006; Yang et al., 2006; Carneiro et al., 2007). Most of these techniques define a parametric or non-parametric model to capture the relationship between image features and keywords. Even though some of these techniques have shown impressive results, one thing that is sorely missing in the annotation literature is comparison with very simple 'straw-man' techniques.

The goal of this work is to create a family of baseline measures against which new image annotation methods could be compared to better understand the gains and justify the need for more complex models and training procedures.[1] We introduce several simple techniques characterized by a minimal training requirement that can efficiently serve this purpose. Surprisingly, we also show that these baseline techniques can outperform more complex state-of-the art image annotation methods on several standard datasets, as well as a large Web dataset.

Arguably, one of the simplest annotation schemes is to treat the problem of annotation as that of image-retrieval. For instance, given a test image, one can find its nearest neighbor (defined in some feature space with a pre-specified distance measure) from the training set, and assign all the keywords of the nearest image to the input test image. As we show in Section 4, some simple distance measures defined on even global image features perform similar to or better than several popular image annotation techniques. One obvious modification of this scheme would be to use K-nearest neighbors to assign the keywords instead of relying on just the nearest one. In the multiple neighbors case, as we discuss in Section 3.3, one can easily assign the appropriate keywords to the input image using a simple greedy approach, further enhancing the annotation performance.

The K-nearest neighbor approach can be effectively extended to incorporate multiple distance measures, possibly defined over distinct feature spaces. Recently, combining different distances or kernels has been shown to yield good performance in object recognition tasks (Frome et al., 2007; Varma and Ray, 2007). In this work, we explore two different ways of combining different distances to create the baseline measures. The first one simply computes the average of different distances after scaling each distance appropriately. The second one is based on selecting relevant distances using a sparse logistic regression method, Lasso (Tibshirani, 1996). For this, one needs a training set containing *simi-*

[1] An earlier version of this work has appeared in the European Conference on Computer Vision (ECCV 2008, (Makadia et al., 2008))

lar and *dissimilar* images. A typical training set provided for the annotation task does not contain such information directly. We show that one can train Lasso by creating a labeled set from the annotation training data. Even such a weakly trained Lasso outperforms the state-of-the-art methods in most cases. Surprisingly, however, the averaged distance does better or similar to the noisy Lasso.

The main contributions of our work are that it: (1) introduces a simple method to perform image annotation by treating it as a retrieval problem in order to create a new baseline against which annotation algorithms can be measured, and (2) provides exhaustive experimental comparisons with several state-of-the-art annotation methods on three different datasets. These include two standard sets (Corel and IAPR TC-12) and one web dataset containing about $20K$ images.

2. Prior work

Text-based image annotation continues to be an important practical as well as fundamental problem in the computer vision and information retrieval communities. From the practical perspective, current image search solutions fail to effectively utilize image content for image search. This often leads to search results of limited applicability.

A number of approaches have been proposed in the past to address the annotation task (Datta et al., 2008). Most of them treat the problem as translation from image instances to keywords. They approach it either directly, drawing inspiration from language translation models, or indirectly exploiting inferences made from co-occurrences of textual tags and images. (Mori et al., 1999) was among the first to consider the co-occurrence view of the translation process where the annotation of a query image could be inferred from examples of regions-keyword associations. This concept has been, in different forms, carried through several works such as (Duygulu et al., 2002; Blei et al., 2003; Jeon et al., 2003; Wang et al., 2004; Lavrenko et al., 2004; Monay and Gatica-Perez, 2003; Feng et al., 2004; Barnard and Johnson, 2005; Metzler and Manmatha, 2005; Hare et al., 2006). Particular approaches differ in their use of image representations and association models.

For instance, the Translation Model of (Duygulu et al., 2002) directly approaches the annotation task by estimating the distribution of words used to describe an image region of a particular kind, from a finite set of possible region appearances. This initial translation approach that directly models text-image associations was subsequently extended to models that ascertain them indirectly, through links established in latent topic/aspect/context spaces. One such model, the Correspondence Latent Dirichlet Allocation (CorrLDA) of (Blei et al., 2003) considers associations through a latent topic space in a generatively learned model. CorrLDA can be described as a generative process in which each image is considered as a collection of latent topics each of which generates a

region and possibly a keyword annotation corresponding to that region. Despite its appealing structure, the model's performance tends to lag behind that of other approaches, in part due to the structure of the latent space which may not accurately reflect the complex topic space dependencies. Furthermore, it is unclear whether the generative model itself accurately models the image formation process. Another big problem with these models is the need for simplifying assumptions to do tractable learning and inference. For instance, the generation of an image region (specifically its descriptors) given a topic is assumed to be Gaussian. Even with these simplistic assumptions, exact inference in the overall generative model is intractable and one has to resort to approximations. The variational approximation proposed in (Blei et al., 2003) can be quite sensitive to model initialization due to local minima. Another cause of poor performance may be the form of objective function that is optimized in CorrLDA. Is maximum likelihood a good measure to optimize, or will a more direct discriminative objective give better performance? Finally, CorrLDA uses image segmentation to obtain regions, which can be quite unpredictable and the segmentation quality can affect the annotation results significantly.

Cross Media Relevance Models (CMRM) (Jeon et al., 2003), Continuous Relevance Model (CRM) (Lavrenko et al., 2004), and Multiple Bernoulli Relevance Model (MBRM) (Feng et al., 2004) assume different, nonparametric density representations of the joint word-image space. In particular, MBRM achieves remarkable annotation performance by considering a joint word/image kernel density model estimated from a large set of labeled examples. Its robust performance comes from the image and text representations it employs: a mixture density model of image appearance that relies on regions extracted from a regular grid, thus avoiding potentially noisy arbitrary segmentation, and the ability to naturally incorporate multi-keyword annotations using multiple Bernoulli models. However, for MBRM one of the drawbacks is that estimating the joint probability of an image and its words requires an expectation over all training images. The complexity of the kernel density representations may hinder applicability of the model to large data sets. In addition to the computational challenges, MBRM requires some important parameters to be set manually. For example, a kernel density estimate is used to approximate the density over image features. A Gaussian kernel is typically used and the choice of kernel bandwidth can affect the density estimates significantly. In a practical setting, the selection of this parameter has significant impact on performance. The same is true for the smoothing parameter μ used in estimating Bernoulli probability for each keyword. In our experiments we did extensive cross-validation to select these parameters. Another practical issue with MBRM models is the use of a non-overlapping grid for extracting image regions. Although it reduces the computational complexity in comparison to using overlapping blocks, the overall performance of the model can be quite sensitive to the block size and shifts in the image.

Alternative approaches based on graph-based representation of the word/image queries (Metzler and Manmatha, 2005), probabilistic latent semantic indexing (PLSI) (Monay and Gatica-Perez, 2003) and cross-language LSI (Hare et al., 2006), while proposing appealing venues for linking the occurrences of words and images, have not resulted in significant performance gains.

More recent research efforts have focused on important extensions of the translation paradigm that exploits additional structure in both visual and textual domains. For instance, (Jin et al., 2004) achieved image annotation by utilizing a coherent language model, and not relying on independence between keywords. Multi-level annotations in (Gao and Fan, 2006) aim not only to identify specific objects in an image, but also incorporate concept ontologies to group similar items and also label a theme for the image. For example, an image of an office might be annotated not only with the items in the image such as a computer monitor and mousepad, but also as office and indoors. The added complexity of such approaches has, unfortunately, restricted their applicability to somewhat limited settings with small-size dictionaries. Despite improved association models, computational complexity of resulting annotation rules has often prevented their applicability to large, real-world datasets.

To address this problem, (Li and Wang, 2003; Li and Wang, 2006) developed a real-time implementation which uses multiresolution 2D Hidden Markov Models to model concepts determined by a training set. This method uses no segmentation to define objects within images, and instead relies on a region based multiresolution approach implemented in the ALIPR image search engine (`http://alipr.com`). While this method may infer higher level semantic concepts based on global features, identification of more specific categories and objects remains a challenge. In an alternative approach, (Carneiro and Vasconcelos, 2005a; Carneiro and Vasconcelos, 2005b; Carneiro et al., 2007) proposed Supervised Multiclass Labeling (SML) technique that aims to learn class-conditional densities using the training data where each keyword is considered a class. The basic assumption behind this method is that in a collection of images annotated with a specific keyword, the background is uniformly distributed while the keyword-related features follow a specific distribution. Hence, when individual image densities in the collection are combined, the keyword-specific density gets reinforced while the background densities get diminished by the process of averaging. Each image in the collection is modeled as a mixture of Gaussians. The image densities are further combined hierarchically to yield class-conditional densities, which are also assumed to be mixture of Gaussians. Even though SML is computationally efficient and based on sound concepts of multiple instance learning, its performance is susceptible to lack of enough training data associated with each keyword. In such a case, it becomes very hard to separate background density from the concept density. This problem is further aggravated if a concept has wide variability in its appearance and each image contains only a subset of the modes. The performance of SML also

depends on several parameters such as the size of blocks in the overlapping grid, the number of Gaussians in different mixtures and levels in the hierarchy.

Even though interesting results have been reported by many techniques, one thing that is common to all the annotation methods mentioned in this section is the lack of comparison with any simple baseline measure. In the absence of such a comparison, it is hard to understand the gains and justify the need for a complex model and training process as required by most of the current annotation methods. Our work addresses this issue by suggesting a family of baseline measures, some of which surprisingly do better than the current state-of-the-art in image annotation on several large real-world datasets.

3. Baseline Methods

We propose a family of baseline methods for image annotation that are built on the hypothesis that images similar in appearance are likely to share keywords. To this end we present image annotation as a process of transferring keywords from nearest neighbors. The neighborhood structure is constructed using image features, resulting in a rudimentary baseline model. The model intricately depends on the notion of distances between features, and we address the necessary steps for constructing this model in the following subsections.

3.1. Features and Distances

Color and texture are recognized as two of the most important low-level visual cues for image representation. The most common color descriptors are based on coarse histograms of pixel color values. These color features are frequently utilized within image matching and indexing schemes, primarily due to their effectiveness and simplicity of computation. Texture is another low-level visual feature that is a key component of image representation. Image texture is most frequently captured with Wavelet features, and in particular Gabor and Haar wavelets have been shown to be quite effective in creating sparse yet discriminative image features. To limit the influence and biases of individual features, and to maximize the amount of information extracted, we choose to employ a number of simple and easy to compute color and texture features.

3.1.1. *Color*

We generate features from images in three different color spaces: RGB, HSV, and LAB. While RGB is the default color space for image capturing and display, both HSV and LAB isolate important appearance characteristics not captured by RGB. For example, the HSV (Hue, Saturation, and Value) colorspace encodes the amount of light illuminating a color in the Value channel, and the Luminance channel of LAB is intended to reflect the human perception of brightness. We compute the RGB feature as a normalized 3D histogram of RGB pixel

values, with 16 bins in each channel. Similarly, the HSV and LAB features are 16-bin-per-channel histograms in their respective colorspaces. We evaluated three distance measures commonly used for histograms and distributions (KL-divergence, L_1-distance, and L_2-distance) on the human-labeled training data from the Corel5K dataset. L_1 performed the best for RGB and HSV, while KL-divergence was found suitable for LAB distances. Throughout the remainder of the paper, RGB and HSV distances imply the L_1 (Manhattan) measure, and the LAB distance implies KL-divergence.

3.1.2. *Texture*

We represent the texture of an image with Gabor and Haar Wavelets. Each image is filtered with Gabor wavelets at three scales and four orientations, resulting in twelve response images (i.e. a single response image is the result of the original image filtered with a Gabor wavelet at a particular scale and orientation). Each of these response images are divided into rectangular blocks (non-overlapping). The mean filter response magnitudes from each block over all twelve response images are concatenated into a feature vector (throughout the text this feature is referred to as 'Gabor'). In a similar process our second feature captures the quantized Gabor phase. In each of the twelve response images, the Gabor response phase angle is averaged over non-overlapping 16×16 blocks. These mean phase angles in $[0, 2\pi)$ are quantized to eight values (which requires only 3 bits of storage). The quantized values over all blocks and over all response images are concatenated into a feature vector (referred to throughout the text as 'GaborQ'). We use L_1 distance for the Gabor and GaborQ features.

The Haar filter is a very simple 2×2 edge filter. Haar Wavelet responses are generated by block-convolution of an image with Haar filters at three different orientations (horizontal, diagonal, and vertical). Responses at different scales were obtained by performing the convolution with a suitably subsampled image. After rescaling an image to size 64x64 pixels, a Haar feature is generated by concatenating the Haar response magnitudes (this feature is referred to as just 'Haar'). As with the Gabor features, we also consider a quantized version, where the sign of the Haar responses are quantized to three values (either 0, 1, or -1 if the response is zero, positive, or negative, respectively). Throughout the text this quantized feature is referred to as 'HaarQ.' We use L_1 distance for the Haar and HaarQ features, as with the Gabor features.

3.2. Combining basic distances

As explained in the previous section, each image is represented with seven features (3 color histograms, and 4 texture features). Although one can compute *basic distances*, i.e. distances between corresponding features in different images, we wish to define a composite distance measure between images that incorporates all seven features. As the goal of this work was to develop simple baseline methods, we focused on linear combinations of basic distances to yield the

composite distance measure. The simplest linear combination would allow each basic distance to contribute equally to the total combined distance. We introduce such a method below as *Joint Equal Contribution* (JEC). A natural extension of this will be to combine basic distances non-uniformly, giving preference to those features which are more relevant for capturing image similarity. We present a second method that learns the weights for combining basic distances using a sparse logistic regression technique, Lasso (see (Tibshirani, 1996)). Although more aggressive methods based on max-margin approaches, recently used in training object classifiers (see (Frome et al., 2007; Varma and Ray, 2007)), can be adapted for this task, for simplicity we do not explore these options here. Also, as we show later on, learning weights with more complex methods using the training data available for the annotation task does not necessarily yield better solutions.

3.2.1. *Joint Equal Contribution (JEC)*
If labeled training data is unavailable, or if the labels are extremely noisy, the simplest possible way to combine distances from different features is to allow each individual basic distance to contribute equally to the total combined cost or distance. Let I_i be the i-th image, from which we have extracted N features $F_i = f_i^1, \ldots, f_i^N$ (in our case $N = 7$). Suppose we can compute the basic distance, $d_{(i,j)}^k$, between corresponding features f_i^k and f_j^k in two images I_i and I_j. We would like to combine the N individual basic distances $d_{(i,j)}^k, k = 1, \ldots, N$ to provide a comprehensive distance between image I_i and I_j. In JEC, where each basic distance is scaled to fall between 0 and 1, each scaled basic distance contributes *equally*. The scaling terms can be determined empirically from the training data. For example, the scaling term for the LAB feature (with KL-divergence as the distance measure) can be taken as the maximum LAB feature distance between all pairs of images in the training set. If we denote $\tilde{d}_{(i,j)}^k$ as the distance that has been appropriately scaled, we can define the comprehensive image distance between images I_i and I_j as $\frac{1}{N}\sum_{k=1}^{N} \tilde{d}_{(i,j)}^k$. We call this distance the Joint Equal Contribution or simply JEC.

3.2.2. *L_1-Penalized Logistic Regression (Lasso)*
Another approach to combining feature distances would be to identify those features that are more relevant for capturing image similarity. This is the well-known problem of feature selection. Since we are using different color (and texture) features that are not completely independent, it is important to determine which of these color (or texture) features are redundant. Logistic regression with L_1 penalty, also known as Lasso (Tibshirani, 1996), provides a simple way to answer this question.

To apply logistic regression for feature selection, one needs to cast the image annotation scenario into something that can be used for Lasso training. To this end, we define a new set X, and each data point $x_l \in X$ is a pair of images

(I_i, I_j). The training set is given by $X = \{x_l = (I_i, I_j) | I_i, I_j \in S, i \neq j\}$, where S is the input set of all training images. Let $y_l \in \{+1, -1\}$ be the label attached to each training point x_l. If a pair (I_i, I_j) contains 'similar' images, then x_l is assigned the label $y_l = 1$, otherwise $y_l = -1$. In Lasso, the optimal weights $(\hat{\boldsymbol{\omega}})$ are obtained by minimizing the following penalized, negative log-likelihood:

$$\hat{\boldsymbol{\omega}} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \sum_{l=1}^{L} \log\left(1 + \exp\left(-\boldsymbol{\omega}^T \boldsymbol{d}_{\boldsymbol{x_l}} y_l\right)\right) + \lambda \left|\boldsymbol{\omega}\right|_1 \tag{1}$$

Here L is the number of image pairs used for training, $|\cdot|_1$ is the L_1 norm, $\boldsymbol{d}_{\boldsymbol{x_l}}$ is the vector containing the individual basic distances for the image pair x_l, and λ is a positive weighting parameter tuned via cross-validation. Given the training data $\{(x_l, y_l)\}$ one can easily solve (1) by converting this into a constrained optimization problem as described in (Tibshirani, 1996). Note that a linear combination of basic distances using the weights computed in (1) will provide a measure of image similarity, so the result is negated to yield the corresponding distance.

The main challenge in applying this simple learning scheme to image annotation lies in creating a training set containing pairs of similar and dissimilar images. Clearly, the typical image annotation datasets do not have this information since each image contains just a few text keywords, and there is no notion of similarity (or dissimilarity) between images. In this setting, we consider any pair of images that share enough keywords to be a positive training example, and any pair with no keywords in common to be a negative example. Clearly, the quality of such a training set will depend on the number of keywords required to match before an image pair can be called 'similar.' A higher threshold will ensure a cleaner training set but reduce the number of positive pairs. On the contrary, a lower threshold will generate enough positive pairs for training at the cost of the quality of these pairs. In this work, we obtained training samples from the designated training set of the Corel5K benchmark (see Section 4). Images that had at least four keywords in common were treated as positive samples for training. Figure 1 shows ten images pairs that had at least 4 keywords in common, and Figure 2 displays ten pairs that had zero keywords in common. Note that a larger overlap in keywords does not always translate into better image similarity, implying that the training set is inherently noisy.

Combining basic distances using JEC or Lasso gives us a simple way to compute distances between images. Using such composite distances, one can find the K nearest neighbors of an image from the test set in the training set. But how should one assign keywords to the test image given its nearest neighbors? In the next section, we present our algorithm for transferring keywords from an image's nearest neighbors.

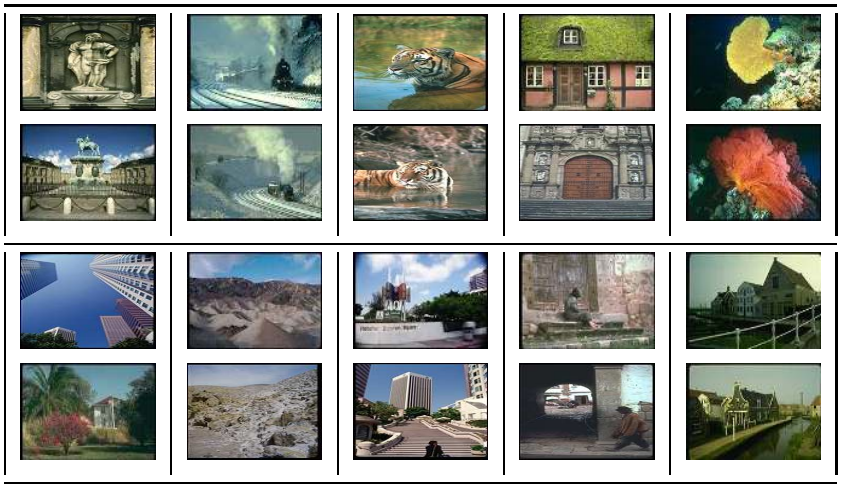


Figure 1. Ten pairs of images from the Corel5K training set that were used as positive training examples for Lasso. In each pair the two images shared at least 4 keywords.

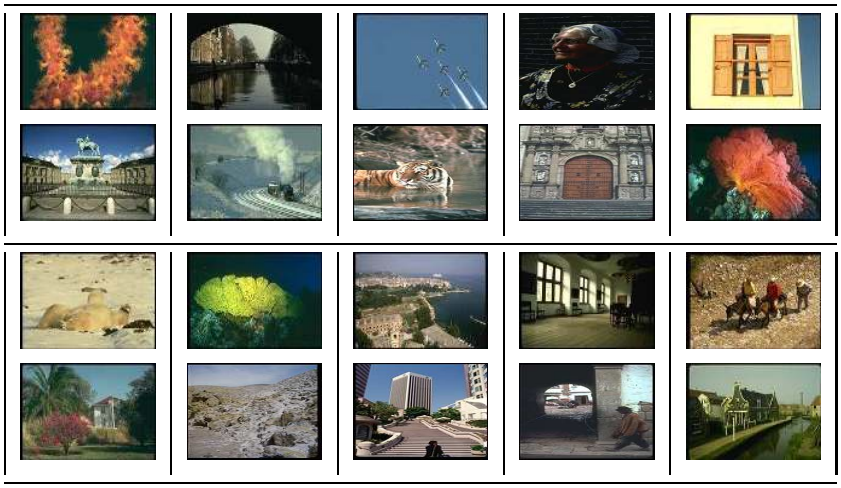


Figure 2. Ten pairs of images from the Corel5K training set that were used as negative training examples for Lasso. In each pair the two images had no keywords in common.

3.3. Label transfer

We propose a simple method to transfer n keywords to an input image $\tilde{I}$ from the input's K nearest neighbors in the training set. Let $I_i, i \in 1, \ldots, K$ be the K nearest neighbors of $\tilde{I}$ in the training set, ordered according to increasing distance (i.e. I_1 is the most similar image). Let $|I_i|$ denote the number of keywords associated with I_i. We then annotate the input image using the following greedy label transfer algorithm.

1. Rank each keyword of I_1 according to its frequency in the training set (where the frequency is just the number of training images in which the keyword appears).

2. Transfer the n highest ranked keywords of I_1 to the input image $\tilde{I}$. If $|I_1| < n$, we still need to transfer more keywords, so proceed to step 3, otherwise terminate.

3. Rank each keyword of neighbors I_2 through I_K based on the product of two factors: 1) their co-occurrence in the training set with the keywords transferred in step 2, and 2) their local frequency (how often they appear as keywords of images I_2 through I_K). Co-occurrence is defined as the number of training images in which a keyword appears alongside keywords transferred in step 2, normalized by the sum of co-occurrences of all candidate keywords. Local frequency is the number of neighborhood images in which the keyword appears, normalized by the sum of local frequencies of all candidate keywords. Based on this keyword ranking, select the best $n - |I_1|$ keywords to transfer to the input image $\tilde{I}$.

Essentially, this label transfer scheme first selects keywords from the nearest neighbor. If more keywords are needed, they are selected from neighbors 2 through N, based on co-occurrence and frequency.

This transfer algorithm differs from other obvious choices. One can imagine simpler algorithms where keywords are selected simultaneously from the entire neighborhood (i.e., all the neighbors are treated equally), or where the neighbors are weighted according to their distance from the test image. However, an initial evaluation showed that these simple approaches underperform in comparison to our two-stage transfer algorithm (see Section 4).

In summary, our baseline annotation methods are composed of two steps. First, a composite image distance (computed with JEC or Lasso as discussed in Section 3.2) is used to identify nearest neighbors. Next, the desired number of keywords are transferred from the nearest neighbors as described above. Is there any hope to achieve reasonable results for image annotation using such simplistic methods? To answer this question, we evaluate our baseline methods on three different datasets as described in the following section.

4. Experiments and Discussion

Our experiments examined the behavior and compared the performance of the proposed baselines for image annotation on three different image datasets.

– The Corel5K set (Duygulu et al., 2002), (illustrated in Figure 3) has become the de facto evaluation benchmark in the image annotation community

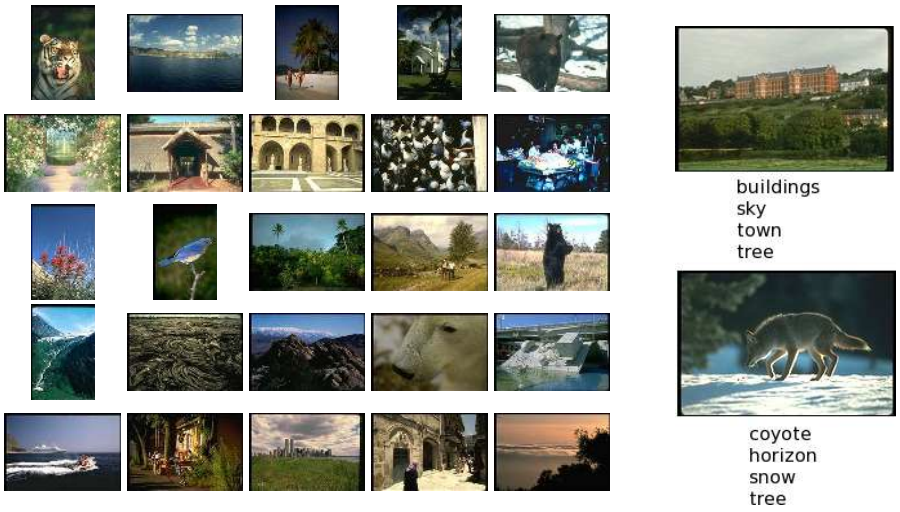


Figure 3. Sample data from the Corel5K benchmark. On the left are 25 randomly selected images from the dataset. On the right are two sample images and their associated annotations.

(Lavrenko et al., 2004; Metzler and Manmatha, 2005; Yavlinsky et al., 2005; Feng et al., 2004; Carneiro et al., 2007). The set contains 5,000 images collected from the larger Corel CD set, split into 4,500 training and 500 test examples. The set is annotated from a dictionary of 374 keywords, with each image having been annotated by an average of 3.5 keywords. Out of the 374 keywords, only 260 appear in both the train and test sets.

– IAPR TC-12 is a collection of 20,000 images of natural scenes that include different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life[2]. Unlike other similar databases, images in IAPR TC-12 are accompanied by free-flowing text captions in three languages (English, Spanish and German). While this set is typically used for cross-language retrieval, we have concentrated on the English captions and extracted keywords (nouns) using the TreeTagger part-of-speech tagger[3]. From this initial corpus, grayscale images were discarded as well as those keywords that appeared too infrequently (along with their associated images). The final set contains 19,805 images and 291 keywords, with each image having an average of 4.7 keywords. 17,825 images were used for training and 1,980 images for testing, keeping the test-train ratio similar to that of Corel5K. Example images and captions from IAPR are depicted in Figure 4.

[2] `http://eureka.vu.edu.au/~grubinger/IAPR/TC12_Benchmark.html`
[3] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`

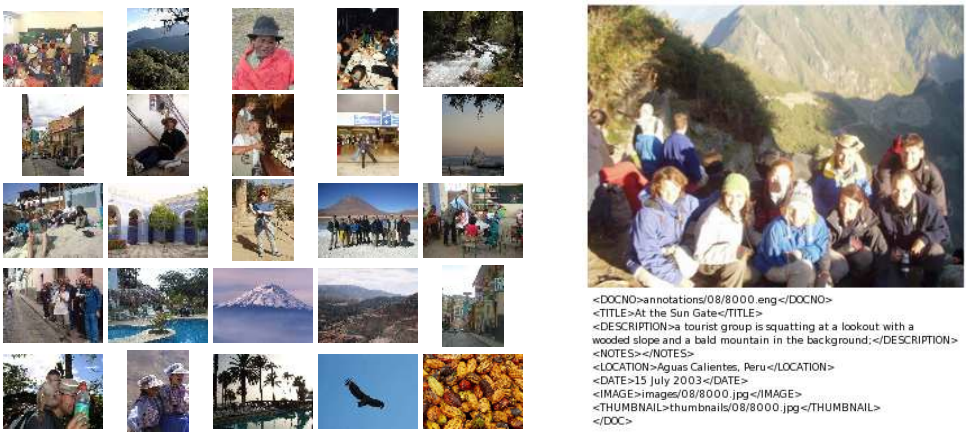


Figure 4. Sample IAPR data. On the left are 25 randomly selected images from the dataset. On the right is a single image and the associated text. Noun extraction from the caption provides keywords for annotation.

– The ESP set consists of images collected from the ESP collaborative image labeling game[4] (von Ahn and Dabbish, 2004)). In the ESP game, two players assign labels to the same image without communicating. Only common labels are accepted. As an image is shown to more teams, a list of taboo words is accumulated, increasing the difficulty for future players and resulting in a challenging dataset for annotation. The set we obtained[5] contains 67,796 images. After discarding images associated with infrequent keywords, we were left with 21,844 images annotated by a dictionary of 269 keywords. On average each image is annotated with 4.6 keywords, and the image set is split into 19,659 training and 2,185 test images. Examples are shown in Figure 5.

For both the IAPR TC-12 and ESP datasets, we have made available the dictionaries and test-train partitions used in our evaluations[6]. For all three datasets, we evaluated the performance of a number of baseline methods. For comparisons on Corel5K, we summarized published results of several approaches, including the most popular topic model (i.e. CorrLDA (Blei and Jordan, 2003)), as well as MBRM (Feng et al., 2004) and SML (Carneiro et al., 2007), which have shown state-of-the-art performance on the Corel5K set. On the two new datasets used in this study (IAPR TC-12 and ESP), we compared the performance of our baseline methods against MBRM (Feng et al., 2004)[7].

When comparing the performance of different image annotation methods, we focused on three different types of baseline measures: 1) performance of basic

[4] `http://www.espgame.org`
[5] `http://hunch.net/~jl/`
[6] `http://www.cis.upenn.edu/~makadia/annotation/`
[7] No implementation of SML (Carneiro et al., 2007) was publicly available.

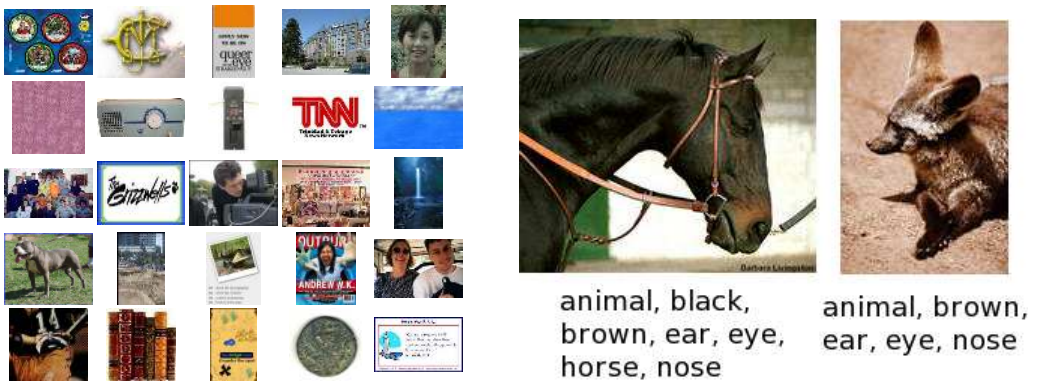


Figure 5. Sample ESP data. On the left are 25 randomly selected images from the dataset, while on the right are two images and their associated keywords. These images are quite different in appearance and content, but share many of the same keywords.

Table I. Weights for the seven different features learned with Lasso (see equation 1). Note that the RGB weight is very close to zero, which indicates that the RGB basic distance contributes very little to the composite image distance. While these weights were learned using the Corel5K training images as described in Section 3.2.2, they were applied to all three datasets: Corel5K, IAPR, and ESP.

	RGB	HSV	LAB	Haar	HaarQ	Gabor	GaborQ
Weights ($\hat{\omega}$)	0.03	-0.39	-0.61	-0.19	-0.42	-0.65	-0.17

distance measures, 2) performance of the trained weighted distance model using Lasso, and 3) performance of the Joint Equal Contribution (JEC) model, where all basic distances contribute equally to the global distance measure. For Lasso, the weights learned for combining basic distances (as described in Section 3.2.2) are shown in Table I. Other than evaluating their comparative performance, to understand the effects of individual basic distances, we also examined the impact of leaving out one basic distance measure at a time in the JEC model. Furthermore, experiments were conducted to understand the contribution of color and texture features separately.

The performance of each model was evaluated using five measures, following the methodology used in (Carneiro et al., 2007; Feng et al., 2004). We report average precision and recall rates obtained by different models, as well as the number of total keywords recalled. Precision and recall are defined in the standard way: the annotation precision for a keyword is defined as the number of images assigned the keyword correctly divided by the total number of images predicted to have the keyword. The annotation recall is defined as the number of images assigned the keyword correctly, divided by the number of images assigned the keyword in the ground-truth annotation. The results shown in the following

					
Predicted keywords	sky, jet, plane, smoke, formation	grass, rocks, sand, valley, canyon	sun, water, sea, waves, birds	water, tree, grass, deer, white-tailed	bear, snow, wood, deer, white-tailed
Human annotation	sky, jet, plane, smoke	rocks,sand, valley, canyon	sun, water, clouds, birds	tree, forest, deer, white-tailed	tree, snow, wood, fox

Figure 6. Predicted keywords using JEC versus the human annotations for a few example images in the Corel5K dataset (using all 260 keywords).

sections are the average precision and recall over all keywords. The number of individual keywords with positive recall is also reported.

Similar to other approaches, we assign exactly 5 keywords to each image using label transfer. In addition to annotation, we report two retrieval performance measures: retrieval precision averaged over all keywords and retrieval precision averaged over the recalled keywords (Carneiro et al., 2007). The images predicted to have a particular keyword are ranked according to the 'strength' of this assignment, where the strength is determined by the frequency in which the keyword appears in the 5 nearest neighbors (ties are broken by random selection). The retrieval precision is then defined as the fraction of the ten highest ranked images which contain the keyword in the true annotation.

4.1. Corel

The label transfer method defined in the previous section explains how we can assign keywords to any input image. Using the JEC scheme with our proposed label transfer algorithm, we assign five keywords to each test image in Corel5K. Figure 6 compares the five predicted keywords against the ground-truth (i.e. human-assigned) keywords for a number of sample images. Since the human-annotations often contain less than five keywords, in some cases JEC predicts keywords that are not in the ground-truth set but correctly describe the image content nonetheless. For example, the first image in the figure is predicted to have the keyword *formation*. Arguably this is a correct description of the planes in the image even though it is not one of the human-assigned keywords.

The quantitative results from experiments on the Corel5K set are summarized in Table II. The top portion of the table displays published results from several top-performing methods that approach the annotation problem from different perspectives, using different image representations: CRM (Lavrenko et al., 2004), InfNet (Metzler and Manmatha, 2005), NPDE (Yavlinsky et al., 2005), MBRM (Feng et al., 2004) and SML (Carneiro et al., 2007). The middle part of the table shows results from using only the basic distances computed

Table II. Results for all 260 keywords in Corel5K for different annotation algorithms. The second and third column show the mean precision and mean recall, respectively, over all keywords. Note the published results for CRM, InfNet, and NPDE did not include the retrieval results. The average distance (JEC) performs the best in all categories.

Method	Precision	Recall	# words with rec>0	Retrieval precision	Retrieval for words with recall >0
CRM (Lavrenko et al., 2004)	0.16	0.19	107	-	-
InfNet (Metzler and Manmatha, 2005)	0.17	0.24	112	-	-
NPDE (Yavlinsky et al., 2005)	0.18	0.21	114	-	-
MBRM (Feng et al., 2004)	0.24	0.25	122	0.30	0.35
SML (Carneiro et al., 2007)	0.23	0.29	137	0.31	0.49
RGB	0.20	0.23	110	0.24	0.49
HSV	0.18	0.21	110	0.23	0.45
LAB	0.20	0.25	118	0.25	0.47
Haar	0.06	0.08	53	0.12	0.33
HaarQ	0.11	0.13	87	0.16	0.35
Gabor	0.08	0.10	72	0.11	0.31
GaborQ	0.05	0.06	52	0.07	0.26
Lasso	0.24	0.29	127	0.30	0.51
JEC	**0.27**	**0.32**	**139**	**0.33**	**0.52**

over individual features (RGB through GaborQ). Finally, the last two rows list results from the baseline methods that rely on combinations of basic distances from multiple features.

Individual basic distances show a wide spread in performance scores, ranging from high-scoring LAB and RGB color measures to the potentially less effective quantized Gabor phase (GaborQ). It is interesting to note that some of the individual basic distances perform on par or better than several more complex published methods. For instance, the LAB color feature alone outperforms CRM (Lavrenko et al., 2004), InfNet (Metzler and Manmatha, 2005), and NPDE (Yavlinsky et al., 2005). More surprising, however, is the fact that the measures which arise from combinations of individual distances (Lasso and JEC) perform better than most other published methods. In particular, JEC, which emphasizes

Table III. Results for 168 keywords in Corel5K. The evaluation is the same as performed in Table II except that it is on a subset of 168 keywords to match the dictionary used for CorrLDA in (Blei and Jordan, 2003).

Method	Precision	Recall	# words with rec>0	Retrieval precision	Retrieval for words with recall >0
CorrLDA	0.06	0.09	59	0.27	0.37
RGB	0.27	0.31	95	0.27	0.44
HSV	0.21	0.27	90	0.24	0.40
LAB	0.25	0.32	99	0.28	0.43
Haar	0.09	0.12	51	0.13	0.31
HaarQ	0.15	0.18	81	0.19	0.34
Gabor	0.10	0.14	60	0.11	0.29
GaborQ	0.08	0.11	46	0.08	0.27
Lasso	0.27	0.36	101	0.30	0.46
JEC	**0.32**	**0.40**	**113**	**0.35**	**0.48**

equal contribution of all the feature distances, shows domination in all five performance measures. One reason for its strong performance may be due to the use of a wide spectrum of different features. Such features contribute along different "orthogonal" dimensions to the final distance measure, enhancing the annotation performance.

It should be noted that most top-performing methods in the literature rely on instance-based representations (such as MBRM, CRM, InfNet, and NPDE) which are closely related to our baseline approach. While generative parametric models such as CorrLDA (Blei et al., 2003) have significant modeling appeal due to the interpretability of the learned models, they fail to match the nonparametric representations on this difficult task. Table III confirms that the gap between the two paradigms remains large. Note that the evaluation in Table III is over a dictionary of 168 keywords rather than the larger set of 260 keywords. This is because the CorrLDA (Blei et al., 2003) used this smaller set of 168 keywords.

Another interesting result is revealed by comparing the JEC baseline with Lasso. One may expect that the selection ability of Lasso should result in increased levels of performance compared to the equal contributions in JEC. However, this is not the case in part because of the different requirements posed

Table IV. All-but-one testing of JEC scheme. In each row, a different feature was left out of JEC. It is clear from these results that all seven features make some positive contribution to the combined distances. The last row shows the JEC results for the full set of features for reference.

Feature left out	Precision	Recall	# words with rec>0	Retrieval precision	Retrieval for words with recall >0
RGB	0.27	0.31	134	0.32	0.53
HSV	0.27	0.31	137	0.32	0.52
LAB	0.27	0.32	134	0.33	0.53
Haar	0.26	0.31	133	0.32	0.54
HaarQ	0.26	0.30	130	0.31	0.53
Gabor	0.25	0.29	128	0.30	0.53
GaborQ	0.26	0.31	134	0.33	0.53
None	0.27	0.32	139	0.33	0.52

by underlying techniques of the two models. Lasso relies on the existence of the sets of positive (similar) and negative (dissimilar) pairs of images, while JEC requires no learning. Since the Lasso training set was created artificially from the annotation training set, the effect of noisy labels undoubtedly reflects on the model's performance. Thus, in the presence of label noise, it is not clear if using more aggressive objective functions (e.g., those based on maximizing the margin) will do any better than simple Lasso.

We further contrast the role of individual features and examine their contribution to the combined baseline models in experiments summarized in Tables IV and V. The performance of individual features discussed above may tempt one to leave out the low-performing features, such as the texture-based Haar and Gabor descriptors. However, Table IV suggests that this is not necessarily the right thing to do. Correlated features, such as HSV and LAB may contribute little jointly and could potentially be left out. While the texture-based descriptors individually lead to inferior annotation performance, Table IV offers evidence that they complement the color features. A similar conclusion may be reached when considering the joint performance of all color and all texture features, depicted in Table V: either of the two groups alone results in performance inferior to the JEC combined model.

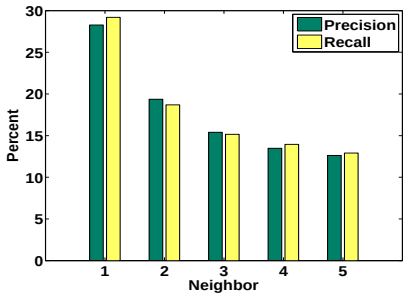
As mentioned earlier, the greedy label transfer algorithm utilized in JEC is not immediately obvious. One straightforward alternative is to transfer all keywords simultaneously from the entire neighborhood while optionally weighting

Table V. Results for all 260 keywords in Corel5K. The first row of results, 'Texture', evaluates the JEC distance-combination scheme using only the four texture basic distances (Gabor, GaborQ, Haar, and HaarQ). Similarly, the second row, 'Color', evaluates only the three color basic distances (RGB, HSV, and LAB). The third row shows the full JEC results combining all the texture and color distances.

Feature Class	Precision	Recall	# words with rec> 0	Retrieval precision	Retrieval for words with recall > 0
Texture	0.16	0.19	101	0.24	0.45
Color	0.23	0.26	120	0.27	0.51
Texture + Color	0.27	0.32	139	0.33	0.52

Table VI. Evaluation of alternative label transfer schemes on Corel5K. On the left, we assess two simple methods. *All neighbors equal* simultaneously selects keywords from all 5 nearest neighbors. Keywords are ranked by their frequency in the neighborhood. *All neighbors weighted* applies an additional weighting relative to the distance of the neighbor from the test image. On the right, we evaluate the individual neighbors in isolation (i.e. all keywords transferred from a single neighbor).

	P%	R%	N^+	rP%	rP^+%
All neighbors equal	23	24	113	39	56
All neighbors weighted	25	31	135	32	50
Proposed method (Section 3.3)	27	32	139	33	52



the neighbors according to their distance from the test image. Additionally, by evaluating the labels transferred from a single neighbor, we can estimate the average "quality" of neighbors in isolation. These results are summarized in Table VI. The simple alternative of selecting all keywords simultaneously from the entire neighborhood (with and without weighting the neighbors) underperforms our proposed label transfer algorithm. In the case of weighted neighbors, the weight assigned to a neighbor's keywords is inversely related to the distance ($\exp^{-dist}$). Regarding individual neighbors, the difference in performance between the first two neighbors is greater than the difference between the second and fifth neighbor. This observation led us to treat the first neighbor specially.

4.1.1. *Discussion*

Looking back at the results above, the relative performance of techniques such as CorrLDA, MBRM, and SML against the proposed JEC baseline on multiple

datasets indicate possible limitations of some of the established techniques. CorrLDA's poor performance on the Corel5K set indicate the suggested generative model for image formation and annotation may not be suited for this task. The need for image segmentation (e.g., using N-cuts (Shi and Malik, 2000)), simplistic distribution assumptions (e.g., image features are distributed as Gaussian given a topic) and inexact inference using variational methods can all lead to poor performance of CorrLDA. As a generative model, MBRM's performance is better than that of CorrLDA but it comes with a strong computational disadvantage. At the run-time, one needs all the training images to estimate kernel densities. Moreover, the kernel density estimates can differ substantially for different choices of block size and small shifts in the data due to lack of overlap in blocks. As discussed earlier, SML's limitations lie in the fact that it tries to learn densities for each keyword from weakly labeled data. When only few observations are available for a keyword, there is a higher chance that the learned densities will not distinguish between foreground and background. For instance, in Corel5K there are 13 keywords which appear in fewer than 5 training images. The lack of training data becomes even a more acute problem if image appearance corresponding to certain keywords varies significantly (e.g. see the different sky appearances in Figure 12). Moreover, various parameters such as the size of blocks in the grid, number of components in mixture models and number of hierarchies also affect the results.

In contrast to methods discussed above, our simple baseline approach of JEC does not need any segmentation or blocking of images. Furthermore, even if a few images are available per concept, the nearest neighbor based approach works fine since no density learning is involved. Finally, since no generative assumptions are imposed, no modeling or optimization approximations are needed. Computationally, however, run-time annotation does require visiting all training images to determine nearest neighbors. However, the run-time complexity can be significantly reduced by using any of the fast approximate nearest neighbor methods such as Locality Sensitive Hashing (LSH (Gionis et al., 1999)).

4.2. IAPR TC-12

The Corel set of (Duygulu et al., 2002) has served as the de facto standard for evaluating many annotation methods. Nevertheless, this set if often criticized for its bias due to insufficiently varying appearance and contrived annotations. We therefore measure the performance of our baseline models, as well as that of individual basic distances, on a more challenging IAPR set which contains 20,000 images (of which we used 19,805) having varying appearance and where the keywords are extracted from free-flowing text captions. Table VII summarizes the quantitative performance of numerous annotation methods on this set. Figure 7 shows some examples of annotated images using the JEC baseline. The annotated images show how inconsistent and challenging the IAPR dataset can be. For example, the JEC baseline correctly predicted the keyword *sky* in both images

Table VII. Results for 291 keywords in the IAPR set. As with the results shown for Corel5K, JEC and Lasso-weighted distances outperform MBRM significantly.

Method	Precision	Recall	# words with recall > 0	Retrieval precision	Retrieval for words with recall > 0
MBRM	0.21	0.14	186	0.23	0.36
RGB	0.20	0.13	189	0.23	0.35
HSV	0.18	0.12	190	0.21	0.31
LAB	0.22	0.14	194	0.25	0.37
Haar	0.17	0.08	161	0.12	0.22
HaarQ	0.16	0.10	173	0.16	0.27
Gabor	0.14	0.09	169	0.15	0.27
GaborQ	0.08	0.06	137	0.08	0.16
Lasso	**0.26**	**0.16**	**199**	**0.27**	0.39
JEC	0.25	**0.16**	196	**0.27**	**0.41**

					
Predicted keywords	clothes, jean, man, shop, square	edge, front, glacier, life, tourist	court, player, sky, stadium, tennis	brick, grave, mummy, stone, wall	desert, grass, mountain, sky, slope
Human annotation	clothes, jean, man, pavement, shop, square	glacier, jacket, life, rock, sky, water, woman	court, player, sky, stadium, man, tennis	brick, grave, mummy, wall	desert, grey mountain, round, stone

Figure 7. Predicted keywords using JEC versus the human annotations for a sampling of images in the IAPR dataset

where the sky appears: the tennis court (image 3) and the desert (image 5). However, since the "ground truth" annotations are extracted from text captions, *sky* does not appear as a keyword for the desert image.

Trends similar to those observed on the Corel set carry over to the IAPR setting as well. The baseline methods show performance superior to that of the MBRM. While color features contribute consistently more than the texture descriptors, we observe improved individual performance of Gabor and Haar

					
Predicted keywords	bikini, girl grass, hair, woman	bear, black, brown, nose, white	band, light, man, music, play	man, old, picture, red, wall	cloud, grass green, hill, red
Human annotation	bed, girl woman	animal, bear black, brown, head, nose	band, light, man, music, red, wheel	black, man, old, red, sit	cloud, gray, green, mountain, picture, rock sky, stone

Figure 8. Predicted keywords using JEC versus the human annotations for a sampling of images in the ESP dataset.

measures. This can be due to the presence of a larger number of images exhibiting textured patterns in IAPR compared to the Corel set. It is also interesting to note that the selection of relevant features using Lasso exhibits performance on par with JEC in two out of the five measures. This is a potential indicator that the criterion for determining the similar pairs in the Lasso training set is more reflective of the true image similarities in IAPR than in Corel.

4.3. ESP

As explained earlier, the ESP game set has arisen from an experiment in collaborative human computing—annotation of images in this case (von Ahn and Dabbish, 2004). The set contains a wide variety of images and annotations, of which we used a small part (21,844 images) for our evaluations. An advantage of this set, compared to Corel and IAPR, lies in the fact that its human annotation reflects a collective semantic agreement among annotators, leading to annotations with less individual bias. Table VIII depicts results of MBRM and our baseline methods on this set. Figure 8 shows some annotation examples using the JEC baseline. Although the predicted keywords using the JEC baseline do not overlap perfectly with the human annotation, in many cases the "incorrect" predicted keywords correctly describe the image. For example, in the fourth image showing a man sitting on couch in front of a wall full of framed pictures, the JEC-assigned keywords arguably describe the image as (or more) accurately than those generated through the ESP game.

In comparison with the results on the Corel5K and IAPR sets, texture features play a much more critical role in the annotation process for the ESP set. For instance, both Haar and Gabor induced-distances fall not far behind the color features. As we speculated with the IAPR dataset, the large ESP image set contains a larger variety of image content and thus there is probably a larger ratio of images that is more suitably represented by their texture.

Table VIII. Results for 268 keywords in ESP. Compared to MBRM, both JEC and Lasso perform much better.

Method	Precision	Recall	# words with rec> 0	Retrieval precision	Retrieval for words with recall > 0
MBRM	0.21	0.17	218	0.14	0.17
RGB	0.21	0.17	221	0.14	0.17
HSV	0.18	0.15	217	0.11	0.14
LAB	0.20	0.17	221	0.14	0.17
Haar	0.21	0.14	210	0.12	0.15
HaarQ	0.19	0.14	210	0.12	0.15
Gabor	0.16	0.12	199	0.10	0.13
GaborQ	0.14	0.11	205	0.10	0.12
Lasso	0.22	0.18	225	0.15	0.18
JEC	**0.23**	**0.19**	**227**	**0.16**	**0.19**

In the last few sections, we provided a thorough evaluation of our baseline methods for annotation. Our label transfer algorithm, a key component of the proposed baselines, utilizes an image's neighborhood structure to select the appropriate keywords for transfer. In all our experiments so far, the label transfer algorithm used five nearest neighbors. In the following section, we study the sensitivity of our label transfer algorithm by varying the number of neighbors used for keyword transfer.

4.4. Varying the neighborhood size for label transfer

How does the change in number of nearest neighbors influence the annotation results obtained from our greedy label transfer algorithm? Clearly, the number of nearest neighbors changes the number of available keywords that can be utilized for annotation. In Figures 9 and 10, we see the effect of varying the number of nearest neighbors between 1 and 5 for all three image datasets using the JEC and Lasso baseline methods for annotation. One would expect the recall to increase as the neighborhood size increases, since the likelihood of seeing the true keywords in the candidate set increases as more images are included. One may also expect a reverse effect with precision. The figures show that the recall does indeed jump from using just one neighbor to using two neighbors. From two to five neighbors, we observe a much smaller improvement in recall, especially in the IAPR and

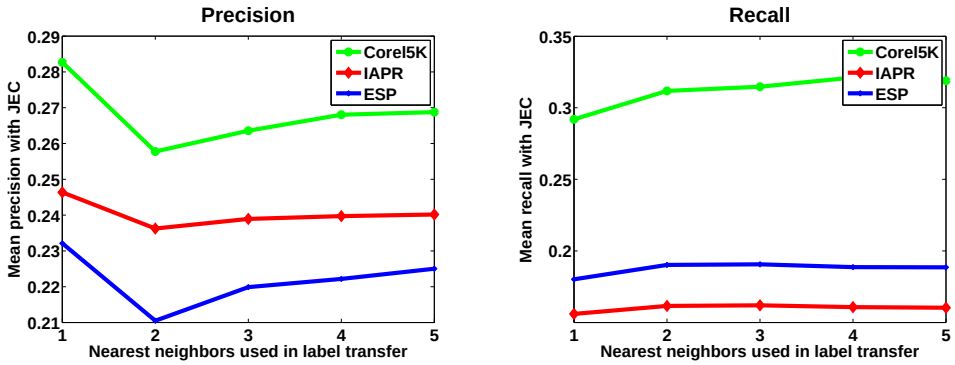


Figure 9. Precision and recall for JEC on all three datasets as the number of nearest neighbors used for label transfer is varied between 1 and 5.

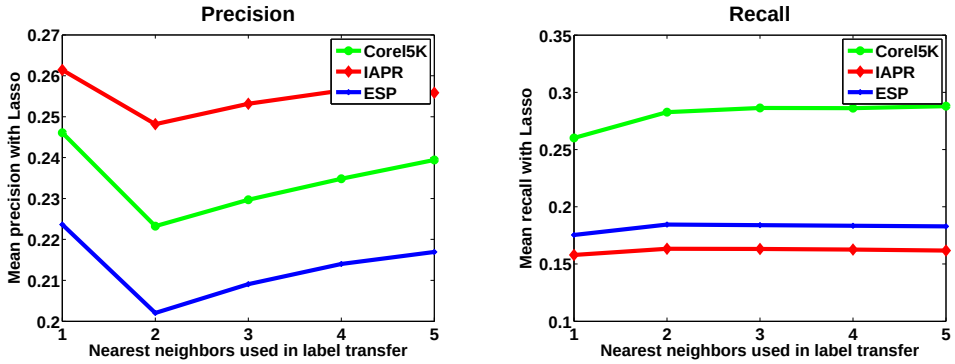


Figure 10. Precision and recall for Lasso on all three datasets as the number of nearest neighbors used for label transfer is varied between 1 and 5.

ESP datasets. This can be attributed to our fixed assignment of 5 keywords. In many cases, the first nearest neighbor has 5 or more keywords, which means the additional neighbors will have no effect on the transfer algorithm. In general, the stability of the recall values over neighborhood sizes is due to the fact that many of the 5 keywords are usually transferred from the first neighbor.

Figures 9 and 10 allow us to see how precision and recall change as the number of nearest neighbors is varied. We would like to go further and thoroughly evaluate the relationship between annotation precision and recall. One simple way to change recall is to vary the number of keywords assigned during the label transfer stage of our baseline annotation methods. Intuitively, assigning more keywords to an image increases the chance that the true keywords will be selected, thus increasing recall. However, by controlling recall with the number of keywords assigned to an image, we would expect an inverse relationship between recall and precision. Until now, we have assigned exactly 5 keywords to each input image in order to stay consistent with the other state-of-the-art methods proposed in the literature. However, in the next section we vary the number

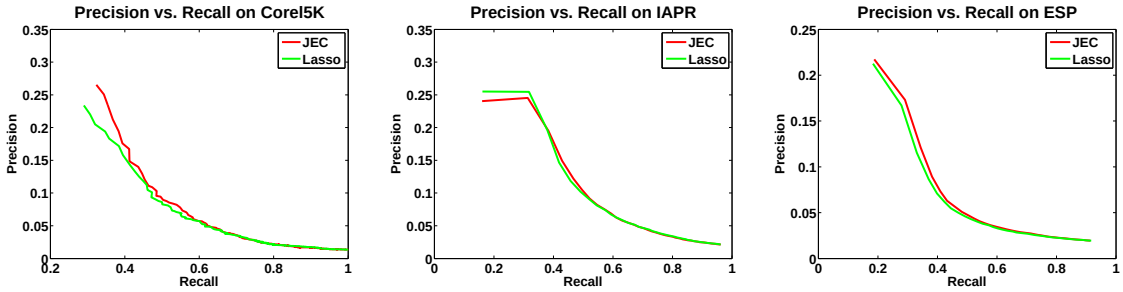


Figure 11. Precision-versus-recall plots generated by varying the number of keywords assigned to an image. On the left is the evaluation on Corel5K, in the middle is the evaluation for the IAPR dataset, and on the right is the evaluation for the ESP set.

of keywords assigned to an image to understand the full relationship between precision and recall for our baseline methods.

4.5. Tradeoff between precision and recall

One of the key challenges in the image annotation task is knowing just how many keywords are necessary to describe the content of an image. For instance, in the ESP dataset, the images have on average 4.6 keywords, but some images have as many as 15 keywords in their ground-truth annotation. Similarly, in the IAPR dataset the average is 4.7 keywords but some images have as many as 23 keywords in their ground-truth annotation. Obviously, for these datasets assigning only 5 keywords during the label transfer stage artificially limits the number keywords that can be recalled correctly for many of the images. Although increasing the number of keywords assigned to an image can help increase the recall (e.g. in the extreme case, if we assigned all 291 keywords to each image in the IAPR test set, we could ensure 100% recall for all keywords), it will lead to a drop-off in the precision. We study this classic tradeoff between precision and recall for all three datasets and show the results in Figure 11. The precision-versus-recall plots are generated by increasing the number of keywords assigned to an image from 5 up to the total number of keywords in each dataset. In order to assign more than 5 keywords to an image using our baseline methods, we ensure that the number of nearest neighbors used during the label transfer stage is the minimum required to see enough unique keywords. The results are shown for both the JEC and Lasso baseline methods. As expected, we see an inverse relationship between precision and recall for both JEC and Lasso on all three datasets. The convexity in the precision-recall relationship can be explained by the fact that we are using the nearest neighbors for label transfer. As expected, the quality of the neighborhood decreases quickly as its size increases, which is necessary for assigning many keywords.

Although the experimental evaluation of our baseline methods in this and previous sections has focused mostly on annotation precision and recall, another

method of evaluation is to study the performance on applications that can easily integrate automatic annotation techniques. The most important of such applications is probably text-based image retrieval, and in the following section we explore some results of our annotation baselines applied to the task of image retrieval.

4.6. Retrieval

The task of automatic image annotation is of great interest because it can play a crucial role in building an effective engine for image retrieval. Assigning descriptive keywords to images allows users to search for images using only text-based queries. Evaluating the performance of an image retrieval engine is different than that of an annotation engine because in retrieval we are only interested in the quality of the first few images associated with a given keyword. Following (Carneiro et al., 2007), we have reported the average retrieval precision over all keywords, as well as just the recalled keywords, for the first 10 retrieved images (see the fifth and sixth columns of Tables II, III, IV, V, VII, and VIII). In addition to these results, here we show some visual examples of the first few images retrieved for a number of different keywords using the JEC scheme. Figures 12, 13, and 14 show the first seven retrieved images for several keywords in the Corel5K, IAPR, and ESP datasets, respectively.

Even for particularly challenging keywords (e.g. *cyclist* and *skull* in IAPR, *diagram* and *tie* in ESP), many of the top retrieved images are correct. Also, many keywords have multiple meanings, commonly referred to as "word sense". In some such cases we see that the retrieved images span numerous meanings of the word (for example, the keyword *ring* in ESP).

4.7. Conclusion

It is widely acknowledged that image annotation is an open and very difficult problem in computer vision. Solving this problem at the human level may, perhaps, require that the problem of scene understanding be solved first. However, identifying objects, events, and activities in a scene is still a topic of intense research with limited success. In the absence of such information, most of the image annotation methods have suggested modeling the joint distribution of keywords and images to learn the association of keywords and low-level image features such as color and texture. Most of these state-of-the-art techniques require elaborate modeling and training efforts. The goal of our work was not to develop a new annotation method but create a family of very simple and intuitive baseline methods for image annotation, which together create a useful annotation evaluation platform. Comparing existing annotation techniques with the proposed baseline methods helps us better understand the utility of the elaborate modeling and training steps employed by the existing techniques.

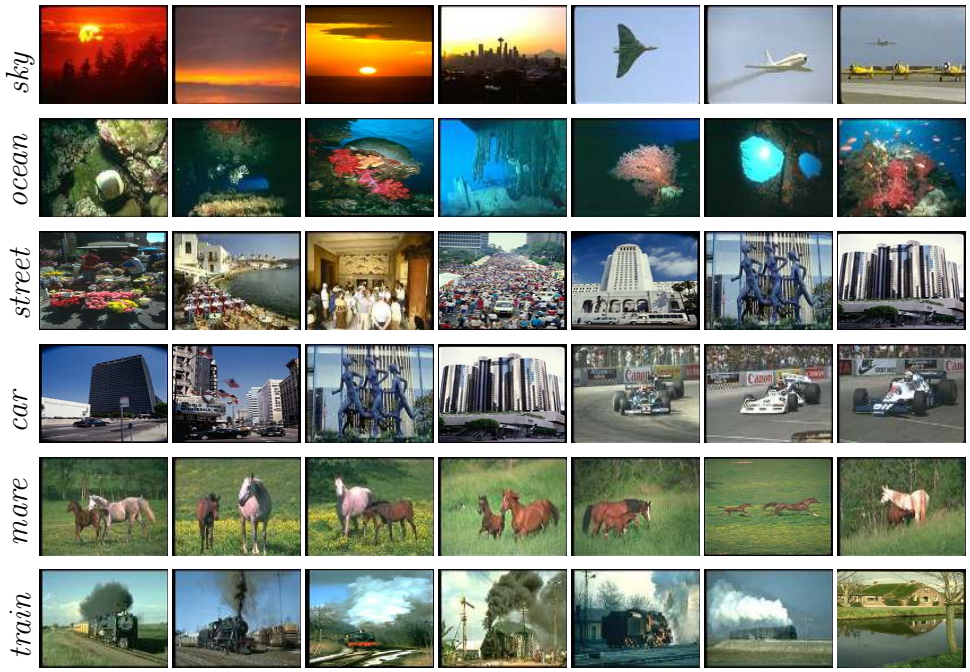


Figure 12. Retrieval results using JEC on Corel5K. Each row shows the first 7 images retrieved for a particular keyword which is shown in the leftmost column.

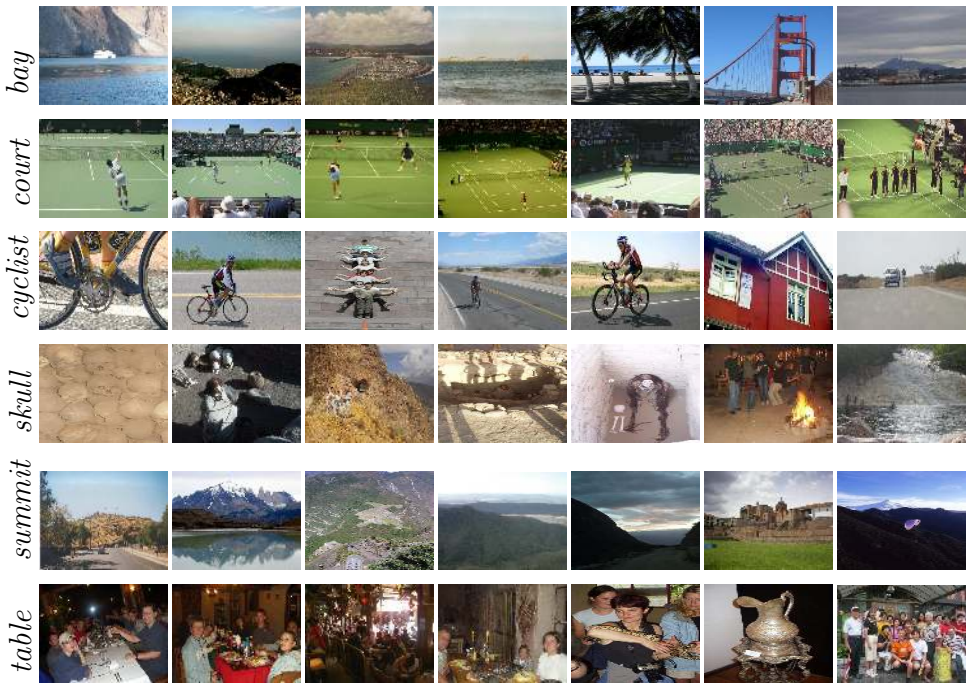


Figure 13. Retrieval results on the IAPR dataset for a number of challenging keywords. Each row shows the first 7 images retrieved for a particular keyword. The images have been scaled independently to have the same aspect ratio for display purposes.

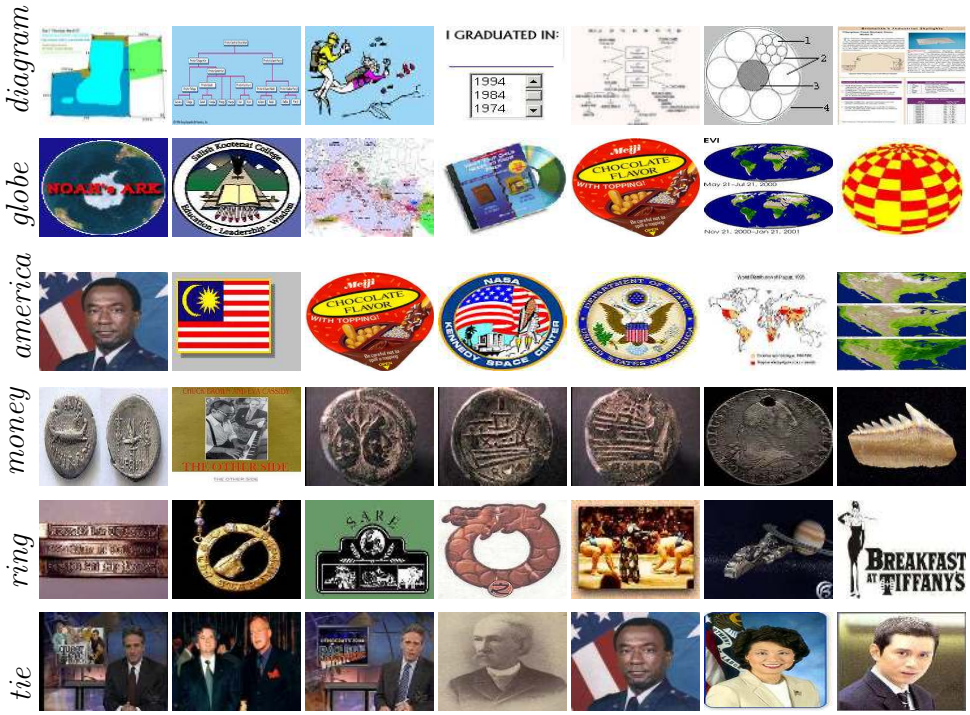


Figure 14. Retrieval results on the ESP dataset for a number of challenging keywords. Each row shows the first 7 images retrieved for a particular keyword. The images have been scaled independently to have the same aspect ratio for display purposes.

Our proposed baseline methods combine basic distance measures over very simple global color and texture features. K-Nearest Neighbors computed using these combined distances form the basis of our simple greedy label transfer algorithm. Our thorough experimental evaluation reveals that nearest neighbors, even when using the individual basic distances, can outperform a number of existing annotation methods. Furthermore, a simple combination of the basic distances (JEC), or a combination trained on noisy labeled data (Lasso), outperforms the best state-of-the-art methods on three different datasets. These somewhat surprising results make a case for revisiting the state-of-the-art methods and carefully analyzing their different modeling and training steps to understand why they fail to achieve performance at the level of these simplistic baseline methods.

Given the general performance level of current annotation methods as well as our proposed baselines, it is clear there is much room for improvement in the state-of-the-art. Our hope is that the existence of such baseline methods as proposed in this work will spur the development of more powerful annotation techniques in the future by providing an effective evaluation platform.

Acknowledgments: Our thanks to Ni Wang for the Lasso training code and Henry Rowley for helpful discussions on feature extraction.

References

Barnard, K. and M. Johnson: 2005, 'Word sense disambiguation with pictures'. *AI* **167**(2005), 13–30.

Blei, D., A. Ng, and M. Jordan: 2003, 'Latent Dirichlet allocation'. *Journal of Machine Learning Research* **3**, 993–1022.

Blei, D. M. and M. I. Jordan: 2003, 'Modeling annotated data'. In: *Proc. ACM SIGIR*. pp. 127–134.

Carneiro, G., A. B. Chan, P. J. Moreno, and N. Vasconcelos: 2007, 'Supervised Learning of Semantic Classes for Image Annotation and Retrieval'. *IEEE TPAMI* **29**(3).

Carneiro, G. and N. Vasconcelos: 2005a, 'A database centric view of semantic image annotation and retrieval'. In: *SIGIR*. pp. 559 – 566.

Carneiro, G. and N. Vasconcelos: 2005b, 'Formulating Semantic Image Annotation as a Supervised Learning Problem'. In: *IEEE CVPR*. pp. 559–566.

Datta, R., D. Joshi, J. Li, and J. Z. Wang: 2008, 'Image Retrieval: Ideas, Influences, and Trends of the New Age'. *ACM Computing Surveys*.

Duygulu, P., K. Barnard, J. F. G. de Freitas, and D. A. Forsyth: 2002, 'Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary'. In: *ECCV*. pp. 97–112.

Feng, S. L., R. Manmatha, and V. Lavrenko: 2004, 'Multiple Bernoulli Relevance Models for Image and Video Annotation'. In: *IEEE Conf. Computer Vision and Pattern Recognition*.

Frome, A., Y. Singer, F. Sha, and J. Malik.: 2007, 'Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification'. In: *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil*.

Gao, Y. and J. Fan: 2006, 'Incorporating concept ontology to enable probabilistic concept reasoning for multi-level image annotation'. In: *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*. pp. 79–88.

Gionis, A., P. Indyk, and R. Motwani: 1999, 'Similarity Search in High Dimensions via Hashing'. In: *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*. San Francisco, CA, USA, pp. 518–529.

Hare, J. S., P. H. Lewisa, P. G. B. Enserb, and C. J. Sandomb: 2006, 'Mind the Gap: Another look at the problem of the semantic gap in image retrieval'. *Multimedia Content, Analysis, Management and Retrieval*.

Jeon, J., V. Lavrenko, and R. Manmatha: 2003, 'Automatic image annotation and retrieval using cross-media relevance models'. In: *Proc. ACM SIGIR Conf. Research and Development in Informaion Retrieval*. New York, NY, USA, pp. 119–126.

Jin, R., Chai, J.Y., and L. Si: 2004, 'Effective automatic image annotation via a coherent language model and active learning'. In: *Proc. ACM Multimedia Conference*. pp. 892–899.

Lavrenko, V., R. Manmatha, and J. Jeon: 2004, 'A model for learning the semantics of pictures'. In: *Advances in Neural Information Processing Systems 16*.

Li, J. and J. Wang: 2003, 'Automatic linguistic indexing of pictures by a statistical modeling approach'. *IEEE TPAMI* **25**.

Li, J. and J. Z. Wang: 2006, 'Real-time Computerized Annotation of Pictures'. *Proc. ACM Multimedia* pp. 911–920.

Makadia, A., V. Pavlovic, and S. Kumar: 2008, 'A New Baseline for Image Annotation'. In: *ECCV*.

Metzler, D. and R. Manmatha: 2005, 'An inference network approach to image retrieval'. In: *Image and Video Retrieval*. pp. 42–50.

Monay, F. and D. Gatica-Perez: 2003, 'On image auto-annotation with latent space models'. In: *Proc. ACM Int'l Conf. Multimedia*. pp. 275–278.

Mori, Y., H. Takahashi, and R. Oka: 1999, 'Image-to-word transformation based on dividing and vector quantizing images with words'. In: *Proceedings of the First International Workshop on Multimedia Intel ligent Storage and Retrieval Management (MISRM).*

Shi, J. and J. Malik: 2000, 'Normalized Cuts and Image Segmentation'. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905.

Tibshirani, R.: 1996, 'Regression shrinkage and selection via the Lasso'. *J. Royal Statistical Soc., B* **58**(1), 267–288.

Varma, M. and D. Ray: 2007, 'Learning The Discriminative Power-Invariance Trade-Off'. In: *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil.*

von Ahn, L. and L. Dabbish: 2004, 'Labeling images with a computer game'. In: *ACM CHI.*

Wang, L., L. Liu, and L. Khan: 2004, 'Automatic image annotation and retrieval using subspace clustering algorithm'. In: *ACM Int'l Workshop Multimedia Databases.*

Yang, C., M. Dong, and J. Hua: 2006, 'Region-based Image Annotation using Asymmetrical Support Vector Machine-based Multiple-Instance Learning'. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition.*

Yavlinsky, A., E. Schofield, and S. Ruger: 2005, 'Automated Image Annotation Using Global Features and Robust Nonparametric Density Estimation'. In: *CIVR.*