Learning to Match Images in Large-Scale Collections

Song Cao and Noah Snavely

Cornell University
Ithaca, NY, 14853

Abstract. Many computer vision applications require computing structure and feature correspondence across a large, unorganized image collection. This is a computationally expensive process, because the graph of matching image pairs is unknown in advance, and so methods for quickly and accurately predicting which of the $O(n^2)$ pairs of images match are critical. Image comparison methods such as bag-of-words models or global features are often used to predict similar pairs, but can be very noisy. In this paper, we propose a new image matching method that uses discriminative learning techniques—applied to training data gathered automatically during the image matching process—to gradually compute a better similarity measure for predicting whether two images in a given collection overlap. By using such a learned similarity measure, our algorithm can select image pairs that are more likely to match for performing further feature matching and geometric verification, improving the overall efficiency of the matching process. Our approach processes a set of images in an iterative manner, alternately performing pairwise feature matching and learning an improved similarity measure. Our experiments show that our learned measures can significantly improve match prediction over the standard *tf-idf*-weighted similarity and more recent unsupervised techniques even with small amounts of training data, and can improve the overall speed of the image matching process by more than a factor of two.

1 Introduction

A key problem in recent Web-scale vision systems is to take a large, unstructured image collection (e.g., a large set of Internet photos) and discover its visual connectivity structure, i.e., determine which images overlap which other images, in the form of an *image graph*, and find feature correspondence between matching images. Finding this structure often involves testing many pairs of images, by matching SIFT features and performing geometric verification. For example, 3D reconstruction methods for large-scale Internet photos—such as all photos of Rome—require finding feature correspondence by matching many pairs of images [1, 2], and other applications, such as summarizing photo collections [3] and unsupervised discovery of objects [4] require similar connectivity information. The computational cost for such feature matching and geometric verification can be quite high, especially if more than a small fraction of the total $O(n^2)$ possible image pairs in a set of n images are matched. However, many large image collections exhibit sparse visual connectivity—only a fraction of possible image pairs overlap. The question is then: how can we compute a good approximation of the image connectivity graph, as efficiently as possible? We present a method that *learns* a good measure for comparing images during such an image matching process, improving this measure as it discovers the structure of the image graph.

To avoid exhaustive feature matching on all $O(n^2)$ image pairs, recent work has used fast, whole-image similarity measures, such as bag-of-words (BoW) [5, 1, 2, 4] or GIST features [6], to predict a smaller set of candidate image pairs on which to perform

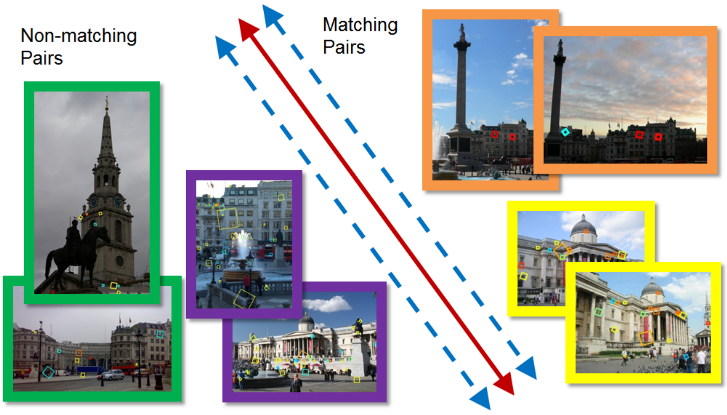

Fig. 1. Training an SVM classifier with positive and negative image pairs. Although each image pair (images with borders of the same color) shown above contain images with many common visual words (shown as boxes with same colors each pair), some pairs are true matches (top right), while others are false matches (bottom left). Accordingly, some visual words are more discriminative (or confusing) than others. Our goal is to learn a weighting of different visual words to better predict matching and non-matching image pairs. This weighting is shown here as a max-margin separating hyperplane. The images shown above are from the **Trafalgar** dataset.

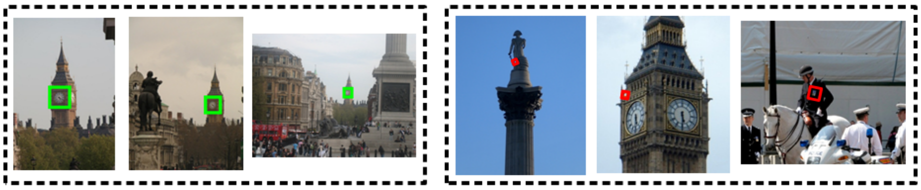

Fig. 2. Two example visual words with different discriminative power. The three images on the left contain a common visual word (in green), which is highly weighted by our learned model. In contrast, the three images on the right also share a common visual word (in red), but do not match; this word is given low weight.

detailed matching. BoW methods in particular, often used in image retrieval [7], have had increasing success for this image matching problem. However, BoW similarities are quite noisy, due to quantization error and imperfect feature detections. As a result, when used to predict image pairs for matching, many cycles are wasted matching features between non-overlapping images, making the matching process unnecessarily time-consuming.

In this paper, we explore a new, iterative approach that learns to predict which pairs of images in an input dataset match, and which do not, using discriminative learning of BoW models. Our method adapts over time in the process of discovering the structure of the image graph; as it attempts to match pairs of input images, the results are used as training data to learn a model specific for that dataset. Motivating our approach is the observation that some visual words are inherently more reliable for measuring image similarity than others, and that these good features seem to be specific to a given dataset (e.g., images of Times Square). For example, some visual words might be more stable across viewpoint and illumination, or less sensitive to quantization errors, than others (Figure 2). This suggests that if each visual word is correctly weighted, then our ability to predict whether two images match can improve. While there are many unsupervised ways to define such weights—e.g., *tf-idf* weighting [5], burstiness [8], co-occurrence measures [9]—we explore the use of supervised learning to find a good set of weights, given example pairs of matching and non-matching image pairs from an image set. Unlike prior heuristic approaches, our method is free to leverage whatever structure is present in the data to learn to separate matching image pairs from non-matching pairs.

Given a collection of images (represented as BoW histograms) of a place, our method starts with an unsupervised similarity measure (e.g., *tf-idf*) and automatically generates training data by first finding a small number of image pairs with high similarity, then applying relatively expensive feature matching and verification steps on these pairs. This results in both positive image pairs (successful matches) and negative pairs (unsuccessful matches). We then use discriminative learning (e.g., SVMs), to learn a new similarity measure on features derived from these example image pairs, by posing this as a linear classification problem. Unlike many classification problems, these features are formed from image *pairs*, rather than individual images, as illustrated in Figure 1. This process iterates, alternating between proposing more images to match, and learning a better similarity measure. We show that, even with very small amounts of training data, our learned models consistently outperform recent unsupervised techniques. Moreover, the overhead of learning is quite low; the linear SVMs we use are extremely efficient to compute, even when using a vocabulary of 1M visual words.

Our contributions are two-fold. First, we propose a fast, simple method for using discriminative learning to classify image pairs in large-scale matching problems, showing significant improvement over state-of-the-art unsupervised methods; we also show that a modified form of regularization, as well as drawing negative training examples from unrelated datasets, can improve our learned models. Second, we propose a new iterative image matching method, based on this learning approach, that can reduce the amount of time needed to find matches in large image sets by a factor of more than two on average.

2 Related Work

Bag-of-words models and image retrieval. In BoW models, features such as SIFT [10] are extracted from an input image, then vector-quantized according to a vocabulary of visual words learned from a large set of features (ideally from a related dataset). An image is then represented as a histogram over visual words. Often *tf-idf* weighting is applied to these histograms [5], inspired by techniques from text retrieval. The similarity of an image pair can then be computed as, say, the dot product or histogram intersection of their weighted histograms. BoW models are often used in image retrieval, but are also common in object classification problems [11], where they have been shown to work well combined with discriminative methods. Our problem differs from traditional classification problems in that we seek to classify *pairs* of images of some scene as matching or non-matching, rather than classifying images into categories. This fits our goal of discovering the structure of a large input collection; such collections are often better described as a graph of pairwise connections, rather than a set of discrete categories. While our problem is related to image retrieval, it differs in that the database and query images are one and the same, and we want to discover the structure of the database from scratch—we aren't matching to a database known in advance. However, we build on methods of computing weights for visual words proposed in the retrieval literature. Many such methods are, like *tf-idf* weighting, unsupervised; Jegou et al. downweight confusing features by modeling burstiness in BoW models [8], while Chum et al. downweight highly correlated sets of visual words ("co-ocsets") [9]; sparse methods have also been applied to identifying informative features [12]. Although such unsupervised weighting schemes improve retrieval performance, we find that supervised learning can exploit structure in the data for our image matching problem much more effectively (Figure 4). Other methods use a form of supervision, but in a more limited way. For instance, Mikulik et al. create a very fine visual vocabulary and compute a probabilistic model of correlations between similar words [13]; others use image geo-tags [14, 15] to select important features. Probably most related to our work is that of Turcot and Lowe, who also gauge feature importance by performing image matching [16]. However, their approach requires matching every image to k other database images, then modifying each database vector individually. In contrast, our discriminative learning approach can generalize much more efficiently, learning a useful metric before touching much of the database, which is key to our goal of quickly predicting matching images in large collections. Supervised learning has also been applied to learn better features through non-linear projection of feature descriptors [17]. We instead learn linear classifiers in the high-dimensional BoW feature space.

Distance metric learning. Our problem can be considered as treating images as high-dimensional feature vectors, and learning a distance metric between images [18–20]. We forumulate this as learning a classifier over pairs of images, predicting a binary variable (matching/non-matching) for each pair. Although online similarity learning over images has been considered before [21, 22], these formulate the learning problem using triplets of training images; in our problem setting, however, matching or non-matching image *pairs* are more readily available as training data, motivating our formulation. While our automatic training data generation procedure is related to that of [4], we use it in an iterative manner to achieve a different objective than learning topic models.

3 Our Approach

Given a set of images $\mathcal{I}$ of a location, our goal is to efficiently compute an image graph on $\mathcal{I}$ with edges linking overlapping images, by performing detailed SIFT matching and geometric verification on some set of image pairs (edges). Through this matching process, we can determine whether or not the pair overlaps (and which features correspond), by thresholding on the number of geometrically consistent matches. For large collections, we wish to check a small subset of the $O(n^2)$ possible edges, yielding an approximate graph; hence, we want to intelligently select a subset of edges to match, so as to quickly compute as complete a graph as possible. Our approach seeks an efficient way to predict whether or not a given image pair will match, by learning over time how to classify pairs of images as matching or non-matching. In this section, we formulate this problem as a one of discriminative learning, and propose an iterative approach that alternates between detailed images matching and learning a discriminative model using the matching results.

3.1 Discriminative Learning of a Classifier for Image Pairs

Consider two images represented as *tf-idf* weighted, sparse, normalized BoW histograms a and b, each with dimension n (with n equal to, say, 1 million). A typical similarity measure $sim(a, b)$ is the cosine similarity, i.e., the dot product $sim(a, b) = a^T b$. A more general way to define a similarity function is $sim(a, b) = a^T M b$, where M is a symmetric $n \times n$ matrix. When M is the identity matrix, this definition reduces to the *tf-idf*-weighted similarity (since a and b are "pre-weighted" with their *tf-idf* weights).[1] At the other extreme, one could learn a full matrix M; however, this would be expensive

[1] We found that such preweighting works better than raw histograms for our learning method.

given the high dimensionality of the histograms. In our case, we restrict our method to learning a diagonal matrix W, which results in a *weighted* dot product of the histograms: $sim(a, b) = a^T Wb = \sum_i w_i a_i b_i$, where the w_i's are the diagonal entries of W. Note that we do not enforce that the w_i's are non-negative, hence $sim(a, b)$ not a true metric; nonetheless, we can still use the output as a decision value for prediction. While forcing M to be diagonal is somewhat limiting, our results suggest that this method still works well in the high-dimensional space of BoW histograms.

Our goal, then, is to learn a weighting w_i on different dimensions of the visual vocabulary specific to a given dataset; for this, we use the tools of discriminative learning. For a pair of images (a, b), we define a feature vector $x^{a,b}$ as the vector of pair-wise products of corresponding dimensions of a and b: $x_i^{a,b} = a_i b_i$. Given these features, $sim(a, b)$ is simply the dot product of the weight vector w with the feature vector $x^{a,b}$.[2] Given this representation, there is a natural formulation of the learning problem as that of learning a hyperplane—or equivalently a set of weights w_i—that separate positive (matching) pairs with negative (non-matching) pairs of images. For this problem, we can automatically generate training data by checking if two images match using detailed SIFT matching and geometric verification: pairs (a, b) that pass become positive training examples $x^{a,b}$ with label $y = 1$; pairs (c, d) that do not match become negative training examples $x^{c,d}$ with label $y = -1$. Figure 1 illustrates this formulation.

We use L_2-regularized L_2-loss SVMs for learning, which in our problem optimize:

$$\min_w \frac{1}{2} w^T w + C \sum_{(a,b) \in S} (\max(0, 1 - y^{a,b} w^T x^{a,b}))^2, \qquad (1)$$

where S is the set of training pairs (a, b). The output weight vector w defines a separating hyperplane, but we also interpret it as a similarity measure (a weighted dot product).

While we find that standard linear SVMs work well given sufficient training data, in our setting we start out with no training data, as it is only generated once we start matching images. Given small amounts of training data, standard SVMs can severely overfit the data, performing worse than *tf-idf* weighting. We propose two extensions to address this problem. First, if negative examples from other image collections are available, we find that these can boost the performance when combined with current training data (though positive examples don't seem to help). Second, we utilize a *modified regularization* for SVMs that uses the *tf-idf* weights as a prior. In particular, our modified approach regularizes the weight vector w to be close to a vector of all ones, w_0 (representing *tf-idf* weighting). To regularize, we substitute w in the regularization term in (1) with $w - w_0$, and solve this modified optimization problem. This smoothly transitions between the original *tf-idf* weights and our learned weights, and softly enforces positiveness of the weights, which helps in preventing overfitting and showing significant improvement over both approaches given limited amounts of training data (Section 4).

Compared to the feature selection method of Turcot and Lowe [16], we do not rely on explicit correspondence found by SIFT, and instead allow the SVMs to choose the weights as they see fit. Interestingly, although our training data is defined by the output of feature matching, in some cases feature matching fails to identify truly matching image pairs, that our learned model can correctly predict (Figure 3). Figure 4 demonstrates the predictive power of our method, by comparing histograms of similarities for matching

[2] Other features defined on an image pair could also be used; e.g., defining the features as the element-wise *min* of the two vectors results in a weighted histogram intersection similarity, and creating a feature vector from all n^2 products of word pairs results in learning a full matrix M.

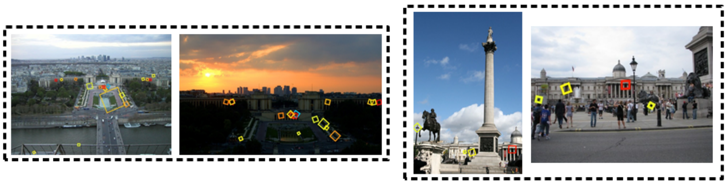

Fig. 3. Correct predictions of non-matching pairs. Due to challenging differences in contrast, illumination and viewpoints, these two image pairs both failed the SIFT matching and verification process, despite exhibiting visual overlap (as well as common visual words, which are marked with boxes of the same color). In contrast, our model is able to correctly highly rank these images, as they happen to have very discriminative visual words (in red). Note that the common visual words may not always imply exact correspondence (e.g., because of repeating patterns).

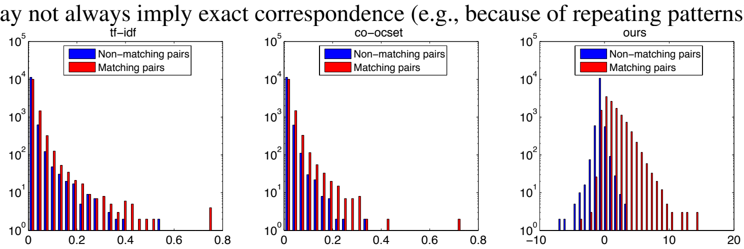

Fig. 4. Histograms of matching vs. non-matching testing pairs. From left to right: histograms of *tf-idf* similarity, co-ocset similarity [9] and our output values (∼100K training pairs) respectively, for matching and non-matching test pairs. Note the log scale on the y-axis. The test pairs consist of randomly chosen unseen image pairs from the **TateModern** dataset.

and non-matching pairs generated by our approach and two unsupervised methods (*tf-idf* and the co-ocset method [9]) on the TateModern dataset (Section 4). Our method can significantly improve separability of matching and non-matching pairs.

3.2 Iterative Learning and Matching

In practice, given a new set of images, there is initially no training data to learn from. However, given even a relatively small amount of training data, our algorithm can still boost performance in predicting matching and non-matching image pairs. Thus, we can bootstrap by matching a small subset of pairs, then learning a better similarity measure from the outcome of matching. We start by using the vanilla *tf-idf* weighted image similarities to rank, for each image, all other images. Then our method performs SIFT matching and verification on a small number of highly-ranked image pairs, and trains a linear SVM using the resulting training data. We use the resulting classifier weights to recompute a similarity measure, to re-rank the candidate lists for all images. The system then resumes the image matching process using the new rankings, and repeats.

Given a learned similarity measure, there are many ways to decide the order in which to attempt to match image pairs. We considered two simple strategies: one is to match all image pairs with similarity values above some threshold; the other is to go down re-ranked candidate lists of each images “layer by layer”, matching each image to its most similar candidate in turn. These two strategies have different impacts on the overall system behaviour. In general, the threshold-based strategy generates a higher percentage of true matching pairs out of all pairs tested, while the layer-based strategy “grows” the image graph more uniformly. In our experiments, we adopt the layer-based strategy, as it is less biased towards parts of the image set that are initially ranked as very similar.

4 Experiments

To evaluate our approach, we collected 5 image datasets from Flickr, each corresponding to a popular landmark and consisting of several thousand images, as summarized in Table 1. The sets were chosen so that each contains a diversity of views, rather than a single dominant view that would be relatively easy to learn an appearance model for. In addition, each dataset contains images that are not pertinent to the scene itself, such as close-ups of people and photos of water. We created a vocabulary of 1M visual words [23] on SIFT features from a separate set of images of Rome, used for all 5 datasets. We also tested our method on two standard image retrieval datasets, Oxford5K and Paris [7, 24]; for each we learned specific vocabularies from the database images. We used LIBLINEAR and SVM-LIGHT[3] to learn our SVMs.

4.1 Performance of Discriminative Learning

First, there are a few key questions that we'd like to answer: How much training data do we need to see an improvement, and how quickly does performance improve with more training data? How much do our two proposed extensions help given limited data? In the limit, given large amounts of training data, how good of a similarity function can we learn for a given location? To answer these questions, we devised an experiment testing how well our approach can separate matching and non-matching pairs in each dataset, given different amounts and types of training data. A perfect similarity measure will, for any given image in the dataset, rank all of the matching images in the rest of the dataset above all of the non-matching ones. To measure this, we selected 50 images for each dataset as "queries" and created ground truth by performing SIFT matching and geometric consistency check between these images and all of the other images in that set (for Oxford5K and Paris, however, the standard query images and ground truth are used). We compare our performance with two unsupervised baseline methods: raw *tf-idf* [5] and co-occurrence set (co-ocset) [9] similarities. We measure the quality of the ranking of the rest of the dataset for each query by the average precision (AP), and performance of each model is measured by its mean AP (mAP) over our test set (higher is better).

We trained SVMs with 200, 1,000, and 2,000 randomly sampled image pairs (with no test query images involved in the training), using equal numbers of positive and negative pairs, and determining the regularization parameter C through cross-validation. To gauge how well our method can perform in the limit, we also trained models with a much larger training set (around 100K training pairs) for each dataset. We also test the effect of our proposed two extensions: modified regularization and adding negative training pairs from an unrelated image set; for the latter, we used the same set of about 1M negative examples from several other datasets.[4]

The results are shown in Table 1. For our 5 Flickr datasets, our models trained with 200 examples (with standard regularization) are slightly worse on average than *tf-idf* similarity, probably due to overfitting to insufficient training data, but both extensions of our approach prove effective in dealing with this issue, each exceeding our baselines on average. Unsupervised co-ocset similarity also shows improvement over *tf-idf* similarity, but our models consistently outperform both; even our unmodified method trained with 1,000 examples outperforms both baselines, and with 2,000 examples the mAP improves even further. The performance of models trained with $\sim$100K examples jumps by a significant margin, illustrating the large potential improvement of our discriminative

[3] http://www.csie.ntu.edu.tw/~cjlin/liblinear/ , http://svmlight.joachims.org/

[4] We only test adding such negative data for our 5 Flickr sets, as they share a common vocabulary.

Table 1. mAP performance of models trained with various training data sizes. The baselines are the mAP scores for rankings using *tf-idf* similarity and co-ocset similarity [9]. Columns marked with **200**, **1000**, **2000**, and $\sim$**100K** show the performance of models trained with corresponding number of examples. N**+neg** corresponds to models trained with the same set of N examples combined with large amounts of negative examples from other datasets; N**+mr** denotes models trained with the same set of N examples using our modified regularization.

Dataset	#img	tf-idf	co-ocset	200	200+neg	200+mr	1000	1000+neg	1000+mr	2000	2000+neg	2000+mr	$\sim$100K
Trafalgar	6981	0.558	0.563	0.620	0.629	0.653	0.689	0.703	0.698	0.719	0.733	0.725	0.794
LondonEye	7047	0.621	0.629	0.586	0.632	0.657	0.650	0.676	0.677	0.673	0.694	0.687	0.783
TateModern	4813	0.712	0.716	0.771	0.793	0.813	0.828	0.835	0.836	0.839	0.851	0.846	0.884
SanMarco	7792	0.577	0.601	0.518	0.535	0.618	0.606	0.633	0.636	0.637	0.658	0.658	0.766
TimesSquare	6426	0.491	0.492	0.410	0.446	0.503	0.474	0.535	0.511	0.498	0.563	0.518	0.617
Average		0.592	0.600	0.581	0.607	0.649	0.650	0.676	0.672	0.673	0.700	0.687	0.769
Oxford5K [7]	5062	0.592	0.608	0.303	-	0.615	0.354	-	0.626	0.397	-	0.629	0.655
Paris [24]	6412	0.635	0.636	0.505	-	0.652	0.620	-	0.668	0.632	-	0.676	0.695
Average		0.613	0.622	0.404	-	0.633	0.487	-	0.647	0.514	-	0.652	0.675

learning approach over time. Note that 100K examples is still a small fraction of the total number of possible pairs in each set (e.g., the **Trafalgar** dataset, with 6,981 images, has over 24M image pairs). Comparing our two extensions, we find that the improvement by modified regularization is more significant when there is very little training data (e.g. 200 examples), while adding unrelated negative examples gives a larger improvement when more data is available (e.g. 2,000 examples). Because Oxford5K and Paris each encompass several disparate landmarks, they require more training data, and hence modified regularization is essential for these two datasets. With modified regularization, models trained with only 200 examples outperform the baselines. We also tested with much lower amounts of training data; we found that with as few as 20 training examples, our method can consistently outperform both baselines in all datasets.

The mAP score above is also used in image retrieval, though we emphasize that we address a different problem in that we seek to discover the connectivity of an entire image set. Our method focuses on learning similarity measures, and as such is orthogonal to other popular methods for improving image retrieval, such as query expansion [25], Hamming embedding [26], or using better feature detectors than the DoG detector. Hence, while our baseline is not as good as that achieved in [7] (e.g. 0.618 for Oxford5K), our method could be combined with others to achieve even better performance.

4.2 System Evaluation

While the experiment above illustrates that our learning framework can yield better similarity measures, how well does our iterative matching system work in practice? The training pairs we get while matching will be different from the random ones selected above. Hence, we also evaluate the performance of our iterative matching system as a whole by running it on the datasets described above. As a reminder, our algorithm matches images in rounds, initially using *tf-idf* similarity to rank candidate pairs for matching, but learning a better model over time. Learning initially takes place once a certain number N of image pairs have been processed. We observe that the margin of performance improvement decreases as the number of training instances (rounds) increases, so at each round we match more image pairs than last round by a factor of β before training. This increases overall efficiency, as learning and re-ranking take time. In our experiments, we use $\beta = 1.5$ and $N = 2000$. We compare to a baseline system that does not rerank image pairs, and simply processes each image's most similar candidates

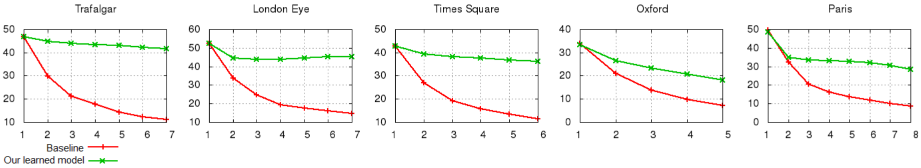

Fig. 5. Per-round *match success rates* for five datasets. The x-axis is the round number, and the y-axis is the percentage of image pairs tested so far found to be true matches. Since we use *tf-idf* similarity in the first round, the corresponding percentages are the same for that round.

in the order computed by *tf-idf* similarity. This mimics current similarity-based large scale image matching methods [1, 4]. We terminate when ≥40 candidates are processed for each image. For this experiment, *efficiency* is the key metric—how quickly can we find matches, and what percent of the image pairs we try turn out to be true matches (meaning we didn't waste effort matching them)? Hence, we evaluate performance after each round of matching using the percentage of image pairs tested so far that were found to be true matches. A higher *match success rate* indicates better efficiency.

Match success rate over time for five datasets are shown in Figure 5; the other datasets show a similar trend. Aside from the initial round (where we use *tf-idf* similarity), our system significantly improves the match success rate. For instance, for the Trafalgar dataset, after seven rounds of matching, our method has a success rate of over 40%, while the baseline method has a success rate of just over 10%. We also found the mAP metric used in Section 4.1 also improves gradually over time.

We found that the overhead of training and re-ranking between rounds is much less than the time spent on image matching. For the Oxford5K dataset, our measured CPU time for matching was 2,621 minutes, while training and re-ranking took 17 and 118 minutes respectively (0.66% and 4.49% of image matching time). To obtain 7,000 matching image pairs, the *tf-idf* similarity-based image matching method checked over 90K image pairs (≥1525 CPU minutes) while our approach checked fewer than 31K (<707 CPU minutes including training and re-ranking overhead), more than a factor of two improvement in efficiency. We also observed similar speedups with other datasets.

5 Conclusions and Discussion

In conclusion, we have shown that even with small amounts of training data, our learned SVM models can predict matching and non-matching image pairs significantly better than *tf-idf* and co-ocset methods for large-scale image matching. Our image matching algorithm iteratively learns a better model of image similarity using accumulated image matching results, in turn improving the efficiency of the matching process.

We find that in our datasets, there are often a small number of reliable, discriminative, highly weighted features. We tried using training data from other datasets to predict them, but found this didn't work well; these "good" features seem specific to each dataset. One limitation of our approach is that discriminative learning is biased towards visual words that appear frequently, which could lead to good classification for canonical images in a dataset, but worse results for rarer ones. This relates to a trade-off between generality and specificity. The more specific the dataset, the easier to learn a good similarity measure. On the other hand, recent work has proposed learning per-image classifiers or similarity functions [27, 28]. It would be interesting to explore what level of granularity of similarity measure (global, local, or something in between) works best.

Acknowledgements. This work was supported in part by the NSF (grants IIS-1149393 and IIS-0964027) and Intel Corporation. We also thank Flickr users for use of their photos.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a day. In: ICCV. (2009)
2. Frahm, J., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B., Lazebnik, S., et al.: Building Rome on a cloudless day. In: ECCV. (2010)
3. Simon, I., Snavely, N., Seitz, S.: Scene summarization for online image collections. In: ICCV. (2007)
4. Philbin, J., Sivic, J., Zisserman, A.: Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. IJCV (2010)
5. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. (2003)
6. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: ECCV. (2008)
7. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR. (2007)
8. Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR. (2009)
9. Chum, O., Matas, J.: Unsupervised discovery of co-occurrence in sparse high dimensional data. In: CVPR. (2010)
10. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
11. Zhang, J., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. IJCV (2007)
12. Naikal, N., Yang, A., Sastry, S.: Informative feature selection for object recognition via sparse PCA. In: ICCV. (2011)
13. Mikulik, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: ECCV. (2010)
14. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR. (2007)
15. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: ECCV. (2010)
16. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: Workshop on Emergent Issues in Large Amounts of Visual Data, ICCV. (2009)
17. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor learning for efficient retrieval. In: ECCV. (2010)
18. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. NIPS (2003)
19. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: NIPS. (2003)
20. Frome, A., Malik, J.: Learning distance functions for exemplar-based object recognition. In: ICCV. (2007)
21. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: An online algorithm for large scale image similarity learning. In: NIPS. (2009)
22. Bai, B., Weston, J., Grangier, D., Collobert, R., Sadamasa, K., Qi, Y., Chapelle, O., Weinberger, K.: Supervised semantic indexing. In: CIKM. (2009)
23. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR. (2006)
24. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR. (2008)
25. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV. (2007)
26. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV **87** (2010) 316–336
27. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV. (2007)
28. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-SVMs for object detection and beyond. In: ICCV. (2011)