Supplementary Material: Learning to Match Images in Large-Scale Collections

Song Cao and Noah Snavely

Cornell University

This supplementary material accompanies the paper "Learning to Match Images in Large-Scale Collections".

1 Datasets

Figure 1 shows a few example images from the five new datasets we collected and used in addition to Oxford5K and Paris datasets in our experiments.

2 Modified Regularization

Table 1 shows the complete performance comparison between our modified SVM regularization (described in the paper) and standard SVM regularization for all datasets we used for evaluation, as a function of training data size; this accompanies Table 1 in the paper. Recall that for this task, we evaluate performance by computing a mAP score for each dataset, obtained by creating, for each of a set of query images for that dataset, a ranked list of all of the other images in that dataset, and comparing that ranking to the ground truth set of matches to that query image.

As described in the paper, the SVMs trained with modified regularization outperform both the *tf-idf* similarity and "co-ocset" similarity, even with as few as 20 training examples (on average); these SVMs also perform much better than the SVMs trained with normal regularization, given the same small amounts of training data.

3 Classification Accuracy

We can also evaluate our learned models as classifiers (e.g., if we wanted to make a forced prediction of all edges in the image graph at some point during the matching process). For each dataset we measured the classification accuracies on a held-out set of test image pairs, with equal numbers of matching and non-matching pairs. With our fully trained models ($\sim$100K examples), we observed an average accuracy of 90.4($\pm$2.9)%.

4 Running Time

To accompany Figure 5 in our paper, Figure 2 shows the number of successful matches found as a function of total CPU running time for all 7 datasets, for our image matching system and for the baseline based on raw *tf-idf* similarities. As described in the paper, we observe that to obtain the same number of true matches, the CPU time required by the *tf-idf* similarity-based approach is often more than a factor of two times higher than our approach (except for the first couple of rounds of learning and matching, where the overhead of our method is comparatively high).

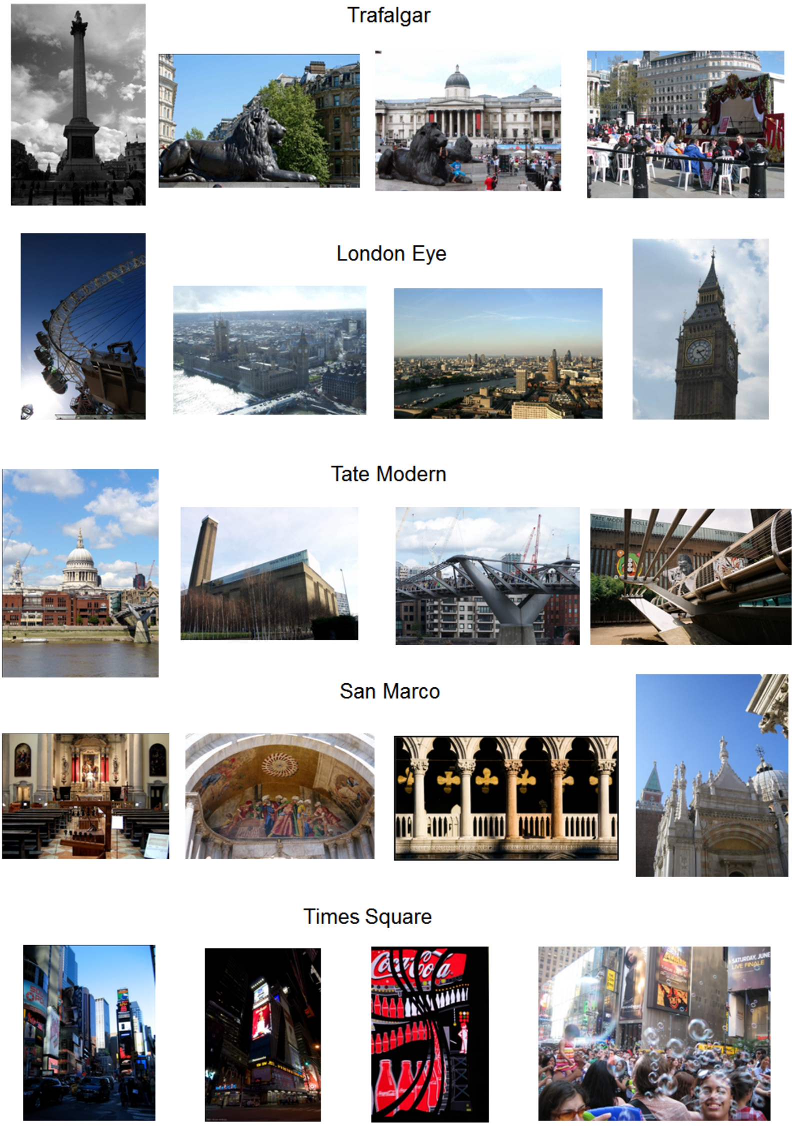

Fig. 1. Some example images from the 5 new datasets we used for evaluation. Note that in these 5 datasets, there also exist many irrelevant images due to the noisiness inherent in Internet image collections.

Table 1. Improved mAP on all 7 datasets using modified regularization with limited training examples. This table shows the performance of our learning method with the standard SVM regularization, with our modified regularization, and the baselines, *tf-idf* similarity and "co-ocset" similarity, in the task of ranking as described in Section 4.1 in the paper. "# Pairs" shows the number of training image pairs used for training corresponding models. The modified regularization performs better than either other method when small amounts of training data are available, though the performance gap between the two regularization approaches shrinks, as expected, when more training data is used.

Dataset	# Pairs	Original Reg.	Modified Reg.	tf-idf	co-ocset
Trafalgar	20	0.388	0.598	0.558	0.563
	40	0.485	0.616		
	200	0.620	0.653		
	1,000	0.689	0.698		
	2,000	0.719	0.725		
LondonEye	20	0.412	0.631	0.621	0.629
	40	0.536	0.646		
	200	0.586	0.657		
	1,000	0.650	0.677		
	2,000	0.673	0.687		
TateModern	20	0.564	0.776	0.712	0.716
	40	0.699	0.788		
	200	0.771	0.813		
	1,000	0.828	0.836		
	2,000	0.839	0.846		
SanMarco	20	0.197	0.609	0.577	0.601
	40	0.306	0.611		
	200	0.518	0.618		
	1,000	0.606	0.636		
	2,000	0.637	0.658		
TimesSquare	20	0.247	0.497	0.491	0.492
	40	0.305	0.497		
	200	0.410	0.503		
	1,000	0.474	0.511		
	2,000	0.498	0.518		
Oxford5K	20	0.128	0.598	0.592	0.608
	40	0.151	0.608		
	200	0.303	0.615		
	1,000	0.354	0.626		
	2,000	0.397	0.629		
Paris	20	0.228	0.641	0.635	0.636
	40	0.292	0.643		
	200	0.505	0.652		
	1,000	0.620	0.668		
	2,000	0.632	0.676		
Average	20	0.309	0.621	0.598	0.606
	40	0.396	0.630		
	200	0.530	0.644		
	1,000	0.603	0.664		
	2,000	0.628	0.677		

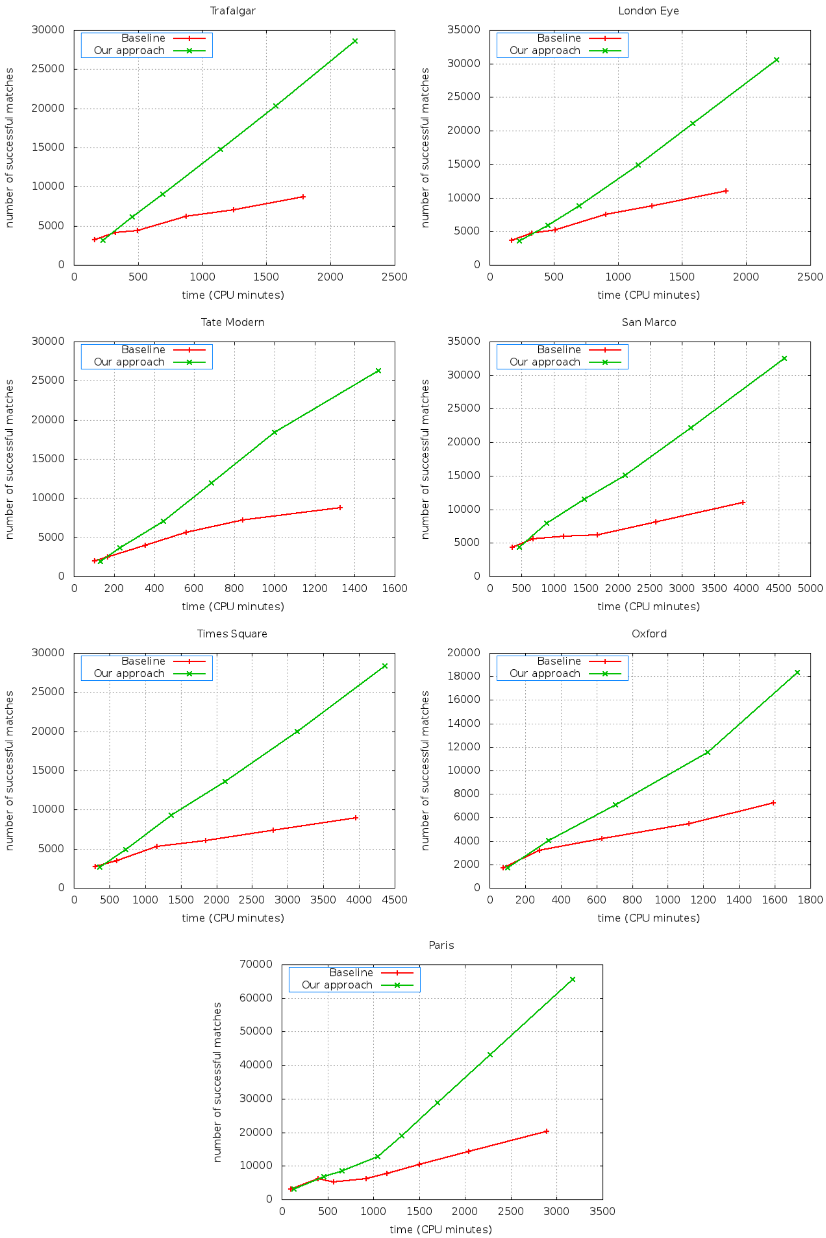

Fig. 2. Number of successful matches as a function of CPU time. This figure plots the numbers of true matches found as a function of CPU time spent, for both our approach and the baseline, for each dataset. After each round (shown by the data points above), the time for our approach is measured as the sum of image matching time, training time and re-ranking time, while that of baseline only includes image matching time. As described in the paper, our approach shows significant improvement in terms of efficiency in discovering true matching image pairs.