Image Categorization Combining Neighborhood Methods and Boosting

Matthew Cooper
FX Palo Alto Laboratory
Palo Alto, CA 94304 USA
cooper@fxpal.com

ABSTRACT

We describe an efficient and scalable system for automatic image categorization. Our approach seeks to marry scalable "model-free" neighborhood-based annotation with accurate boosting-based per-tag modeling. For accelerated neighborhood-based classification, we use a set of spatial data structures as weak classifiers for an arbitrary number of categories. We employ standard edge and color features and an approximation scheme that scales to large training sets. The weak classifier outputs are combined in a tag-dependent fashion via boosting to improve accuracy. The method performs competitively with standard SVM-based per-tag classification with substantially reduced computational requirements. We present multi-label image annotation experiments using data sets of more than two million photos.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; I.2.6 [**Artificial Intelligence**]: Learning—*Concept Learning*

General Terms

algorithms, management

Keywords

image categorization, nearest neighbors, boosting

1. INTRODUCTION

Indexing and retrieval of digital content is a major research focus within multimedia analysis. The central obstacle remains the gap between low-level representations of content and semantic information upon which indices can be constructed. While numerous methods have been proposed to address this difficulty, one line of research aims to incorporate semantic information based on automatic classifiers [15, 24, 28, 21]. Such descriptions can enhance content indexing for retrieval and management applications.

An established approach for constructing such automatic classifiers uses support vector machines (SVMs) to determine the probability than an image or keyframe belongs to a specific category. While this approach has proved successful, it is not easily scaled to either large training sets or large numbers of categories. Although training can often be performed offline, larger training sets also typically incur greater computational costs at test time. As a result, a variety of data sampling approaches have been proposed for classifier training.

At the same time, collections of manually annotated multimedia have proliferated on media sharing sites like Flickr and YouTube. These sites allow unconstrained annotation or tagging for text-based access to content. Throughout, we use the word "label" interchangeably with the words "tag" and "category", and operate under a binary category membership assumption. These public internet resources now provide abundant training data to develop media processing systems, albeit with labels of variable quality. Many standard statistical classifiers are unable to fully exploit these new resources without prohibitive computational complexity.

An emerging point of view is that "model-free" or weak learning techniques can be preferable in large-scale domains [5, 14]. More succinctly, the proposition is that given enough training data, simple models can often do as well or better than more complex models. The use of larger training sets constrains the choice of learning method. In our context, we consider two types of scalability: the number of training examples and the number of categories. To utilize large training sets, the computer vision community has successfully employed approximate non-parametric classification methods for object recognition [32] and scene completion [16]. We also build our approach to categorization on neighborhood-based classification. Spatial indexing of the training data for efficient search is performed once and used for classification *of all labels*. Nearest neighbors (NN) is an attractive learning method because of its minimal limiting assumptions about the statistics of the data. The principal drawback of NN is the computational complexity of naive implementations. We address this difficulty with an approximate nearest neighbor approach that scales with the substantially larger training data sets now available.

Using approximate NN, we efficiently generate a set of classifiers based on subsets of low-level features. We combine

LS-MMRM'09, October 23, 2009, Beijing, China.

these classifiers using boosting [27] to incorporate lightweight per-tag modeling that improves accuracy. The resulting increase in computational complexity is incurred disproportionately in the training phase with negligible cost at test time. With this framework, we leverage publicly available image data and standard features for classification. Our objective in this paper is to evaluate this approach in a generic image annotation context. We consider consumer photographs of highly variable content and annotations of limited quality available from Flickr. We compare directly against a standard SVM baseline for image annotation. Secondly, we study performance and computational tradeoffs with increasing training set size. Our results demonstrate that the approach scales readily to training sets on the order of millions of photos with accuracy that is competitive with per-tag SVM models.

2. RELATED WORK

The research community has proposed various techniques to semantically characterize visual content using automatic annotation. As the scale of available data has grown, research has shifted towards generic learning methods based on a common set of features [25]. One established technique is independent per-label SVM classification. To extend this approach to larger data sets and larger label sets, sampling and bagging strategies are typically employed [26, 30, 39]. Many of these methods demonstrate negligible performance degradations with improved efficiency. Other formal classification approaches also have been proposed and successfully validated in the image analysis and retrieval domain [4, 17, 6, 7], and also for video indexing [29, 38]. While these methods generally train separate models for each label, [33] is notable for extending conventional boosting to multi-label classification while sharing weak classifiers across labels. This work is complementary to ours and provides an avenue for future improvement of our system. In contrast, our approach simplifies training dramatically by unifying indexing for all labels via approximate NN classification.

The use of NN classification for scalable image search is an established approach. Zhang and Zhou [40] used exact nearest neighbors successfully for multi-label classification of several data sets. Hays and Efros [16] used public image databases and NN in combination with other computer vision techniques for scene completion. They used a distance measure for scene matching with exact NN to find candidate images for additional processing. Torralba *et al.* [32] used large data sets culled from the internet for object and scene recognition. Their work employed an approximate NN method based on principal components analysis and a distance measure selected for object recognition. Muja and Lowe have recently proposed randomized methods for accelerating nearest neighbors search with higher dimensional features in vision applications [23]. Li *et al.* [18] extend the annotation by search framework proposed in [37] using public photo collections. They focus their experiments on high quality photographs with annotations and comments. They use content-based image search to identify related photos in the training set and mine the associated text to annotate the test image. [19, 20] construct a tag relevance measure from visual and user information for annotation using Flickr photos. Instead, we consider an alternate approach to image similarity using different features and distance measures. We use an alternate approximation strategy without resorting to offline clustering, image downsampling, or feature projection while maintaining scalability to large labeled data sets. We also employ a different voting method that incorporates distances in the feature space.

Boosting has also been used in object recognition [33], image annotation [39], and image retrieval [31] applications. It has also been used to learn distance embeddings to approximate complex or costly distance measures for nearest neighbors search [3]. We don't use boosting to share information among tag models, although this is a potential extension of our work here. We also are not using boosting to either learn a distance measure or select a subset of low-level features. Instead, we use it to combine neighborhood-based weak annotations in a tag-dependent and efficient manner for categorization.

Our approach aims to combine the accuracy of tag dependent modeling [10] and the scalability of search-based methods [35]. Approximate NN indexing generates a set of weak annotations for each label according to its distribution in the neighborhood of a test photo. Boosting provides per-tag modeling to complement neighborhood-based classification while preserving scalability. The weak annotations are combined using RankBoost [13] to produce the final annotations. The proposed framework enables the use of large training sets to enhance performance without unduly increasing computational costs. Our experiments demonstrate competitive performance with per-tag modeling baselines, scalability to large training sets, and consistent improvements in accuracy with the use of additional training data.

3. SCALABLE CATEGORIZATION

Our system has four components. We first extract standard edge and color features to represent each photo. We next index the training features using a set of spatial trees that enable efficient approximate nearest neighbor search. The training images in the local neighborhood of a test image are then located by searching each tree. A corresponding set of weak annotations are calculated via distance-weighted vote. Finally, the weak annotations are combined using boosting to produce a final annotation score for each tag. The system design emphasizes scalability to large training sets and efficiency at test time.

3.1 Feature extraction

We employ a modified version of the edge orientation histogram feature developed for object recognition applications [9]. We process image blocks in a uniform non-overlapping 5×5 spatial grid. In each block, after obtaining the Canny filter responses, the edge orientations are quantized into 16 uniform bins to form a histogram. The magnitudes of the edge response at each pixel are accumulated in the corresponding bin. An extra bin is used for any edge magnitudes below a threshold, or equivalently, to count the smooth points. This bin dominates the histogram for blocks without significant detectable edges. The resulting feature dimensionality is $25 \cdot 17 = 425$. We also compute the first and second color moments in the LUV color space per block with dimensionality $25 \cdot 6 = 150$. All features are normalized in a per-dimension min-max style. For dimension i, f_i^m and f_i^M are the minimum and maximum of that dimension, each feature value f_i is normalized to $\hat{f}_i = \frac{f_i - f_i^m}{f_i^M - f_i^m}$ where $\hat{f}_i$ is used for further processing.

3.2 Classification Scheme

For categorization, we utilize a standard non-parametric classifier: nearest neighbors. To limit computational complexity, we use established approximation techniques. The nearest neighbor problem is to find k points in a data set $X \subset \mathbb{R}^D$ containing N points that are closest to a query point $q \in \mathbb{R}^D$. Approximation algorithms return the points within $(1+\epsilon)$ distance of the true nearest neighbors. Several effective methods exist for this problem when the dimension is moderate (i.e. on the order of ten), such as kd-trees and metric trees. In [2], Arya and Mount devised a variant of the standard kd-tree that can be constructed in $O(DN \log N)$ time with a space complexity of size $O(DN)$. We have used their data structure and software implementation [22] for approximate search throughout. We also apply an early termination criterion which limits the number of points evaluated and can mitigate the impact of outliers on average performance. Let the parameter m denote the maximum number of points visited before terminating the current search. This parameter is set to $m = 10000$.

Kd-trees do not scale well with increasing feature dimensionality. There are many choices by which these limitations can be surmounted. Often, clustering is used to accelerate nearest neighbors search in multimedia applications [11, 35]. [32] used image downsampling and principal components analysis to reduce the dimensionality prior to nearest neighbor search. Instead, we process disjoint subsets of our features independently for search and annotation, and combine the results using boosting. Assume the features have dimensionality D. We select T, B such that $D = T \cdot B$, and divide the features in to T subsets of size B. In this way, each kd-tree indexes a B-dimensional feature subset, and we compute T corresponding kd-trees. This approach has been successfully used for image orientation classification [36].

Each tree is used as a weak classifier that produces a corresponding annotation result for each tag. Given a query image q, the k nearest neighbors from each of the T trees are identified. Denote these approximate nearest neighbors for tree t by $\mathcal{N}_t(q) = \{x_i : i = 1, \cdots, k\}$. Assume the elements of $\mathcal{N}_t(q)$ are sorted in strictly increasing order of distance from q such that x_1 is the closest to q. Denote the distance from the i^{th} closest of the k images by $d(x_i, q)$. Throughout, we use the Euclidean distance measure. Each training sample x_i is associated with a vector $y(i) \in \{\pm 1\}^L$ where $y(i, l) = 1$ indicates that training sample x_i was tagged with the l^{th} label in the tag vocabulary. A unnormalized confidence value is produced for label l and tree t:

$$\hat{y}_t(q, l) = \sum_{x_i \in \mathcal{N}_t(q)} \left(\frac{d(x_k, q) - d(x_i, q)}{d(x_k, q) - d(x_1, q)} \right) \cdot \delta(y(i, l) - 1) \quad . \quad (1)$$

To combine the rankings using boosting in the following section, we normalize these confidence values such that $\hat{y}_t(q, l) \in [0, 1]$.

3.3 Per-label boosting

The process above generates T weak classifier outputs for each tag. To augment our model-free classification with per-tag modeling, we use boosting [27] to aggregate the rankings of each weak classifier. For this, we reserve a portion of training data that is not used for constructing the spatial trees above. This reserved data is annotated using the spatial trees, and the results and corresponding training labels are used for learning the final boosted classifier. We employ RankBoost [13], which is a boosting algorithm specifically developed for ranking problems. RankBoost iteratively optimizes a reweighted exponential ranking loss to produce a final boosted classifier with the form

$$\hat{Y}(q, l) = \sum_t \alpha_t^{(l)} \hat{y}_t(q, l) \quad , \quad (2)$$

where $\hat{y}_t(q, l)$ are weak classifier outputs for test photo q and tag l, and the $\alpha_t^{(l)}$ are combination weights.

Because the labels are binary and induce a natural bipartite ordering of the training data, RankBoost can be implemented very efficiently. This case is addressed by the RankBoost.B algorithm, as detailed in [13]. In contrast to typical applications of RankBoost, we don't call a weak learner at each iteration. Instead, we select one among the unselected search trees, and determine the appropriate combination weight $\alpha_t^{(l)}$. Denote the sets of positive and negative training examples by X_1 and X_0 respectively. The weighted ranking loss for a specific weak classifier $\hat{y}_t$ is

$$r_t(l) = \sum_{x_0 \in X_0, x_1 \in X_1} W(x_0, x_1)(\hat{y}_t(x_1, l) - \hat{y}_t(x_0, l)) \; . \quad (3)$$

$W(x_0, x_1)$ are weights representing the importance of the pair (x_0, x_1) within the training set. At each iteration, we calculate

$$\alpha_t^{(l)} = \frac{1}{2} \log \left(\frac{1 + r_t(l)}{1 - r_t(l)} \right) \quad . \quad (4)$$

This somewhat resembles the use of boosting in [39] to combine SVM classifiers in multiple per-tag boosted classifiers. In contrast, we employ a loss function designed for ranking problems to combine weak rankings independently per tag. We have yet to explore further accelerating our model training by sharing weak classifiers across models. In summary, RankBoost provides an efficient and principled means of combining the outputs from each search tree into a final classifier. The boosted classifiers use tag specific weightings to aggregate the weak neighborhood-based classifiers.

4. EXPERIMENTAL RESULTS

In this section we review a series of image categorization experiments. First, we examine several choices for designing weak neighborhood-based classifiers. We compare edge and color features and several different feature subsets for spatial indexing and search. Second, we perform two experiments to benchmark our approach relative to per-tag SVMs. In the first, we compare our results to the Mediamill benchmark [29], a high performance per-tag SVM baseline. Next, we compare against per-tag SVMs using a larger test set of Flickr photos. Lastly, we perform scaling tests to examine performance and computational complexity tradeoffs as the number of training photos varies two orders of magnitudes to more than two million photos. Our data was collected from Flickr by searching for images possessing at least one of a set of 900 popular tags uploaded over a several month period. However, any single Flickr user's photos are restricted to a single partition (training, fusion, or test) to limit the impact of near-duplicates in all experiments. Although Flickr tags are subjectively applied and thus pose problems as a source for ground truth, we use them here in experiments with Flickr data. In the experiments with the Mediamill data,

Table 1: Comparing weak classifiers and feature modalities using 42 tags: RS and RP denote random subspaces and random projection, respectively. MAP is averaged per tree/classifier (Avg.) and evaluated after combining classifiers using mean fusion (MF).

Feature	Classifier	MAP	
		Avg.	MF
EHST	SVM (9)	0.078153	0.08550
CLRM	SVM (9)	0.086878	0.084102
EHST	Def. Trees (17)	0.059243	0.109524
CLRM	Def. Trees (17)	0.056484	0.100534
EHST	Spa. Trees (25)	0.060595	0.116385
CLRM	Spa. Trees (25)	0.059607	0.101832
EHST	RS Trees (18)	0.072201	0.111445
CLRM	RS Trees (18)	0.069308	0.081987
EHST	RP Trees (18)	0.083880	0.111445
CLRM	RP Trees (18)	0.086616	0.081987
32×32	PCA Trees (3)	0.063581	0.075528

the provided manual ground truth is used for evaluation. The experiments show that our approach successfully combines large training sets for model-free spatial search with boosting for tag dependent modeling. Both elements of the approach enhance performance while achieving scalability with both the number of tags and training set size.

Before detailing our results we describe our codebase for testing. Throughout we employ a hybrid Python/C++ codebase that is designed for large scale testing in a distributed computing environment. Where possible, computations are performed in independent threads on a per-tree or per-tag basis. The code is written to minimize per-thread memory consumption rather than optimize per-photo complexity. Per-photo processing time could be reduced by using more memory per-thread or otherwise reducing file system accesses. Whenever multiple processing cores are used in parallel, we multiply the resulting computation times by the number of processors to obtain a normalized estimate of timing on a single processor. As a result, these experiments allow us to understand the basic scalability of our system with both the number of tags and training set size without limiting hardware assumptions. Most of the experiments were conducted on a Linux machine with eight 2.66GHz Intel Xeon processors either alone or in combination with other Linux machines with slower processors.

4.1 Comparing weak classifiers

First, we consider several choices for constructing the spatial trees and corresponding weak classifiers using Equation 1. We use a pilot data set containing photos with at least one of the 50 most popular tags. The training set consisted of 28,157 photos, while the test set was 28,158 photos. We consider a set of 42 tags that result after Porter stemming and ranking the tags by thresholding the ratio of the average precision of our method against a random result. This allows us to measure our annotation performance using tags requiring less disambiguation [20]. The tags appear in Table 5. Throughout, we report results for annotation using per-tag average precision (AP) and the mean average precision (MAP) over the set of tags. Table 1 shows MAP results for several choices for weak classifiers. The "Avg." column shows the mean of the per-tree MAPs. The "MF" column shows the MAP after first combining the results of each weak classifier using mean fusion. The edge histogram and color moment features are denoted EHST and CLRM, respectively. Our default trees combine each feature dimension across all spatial blocks so that $B = 25$ for 5×5 spatial blocks and $T = D/B$, using the definitions of Section 3.2. The spatial trees group features within each spatial block together in a corresponding tree so that $T = 25$ and $B = D/25$. We include two other variations for indexing: random projection(RP) [12] and random subspaces (RS) [30, 39]. In both cases, $B = 25$ and $T = 18$. The last row shows the use of principal components analysis (PCA) on downsampled color images as in [32]. For this case, we retain the first $B = 19$ principal component projection coefficients and build a tree for each color channel ($T = 3$).

We also include results for a per-tag SVM baseline. We use reduced training sets for parameter tuning and train the SVMs using asymmetric bagging [30]. Denote the positive and negative training examples for a given tag used for classifier construction by $\mathcal{T}^+$ and $\mathcal{T}^-$, respectively. Then

$$\begin{aligned} |\mathcal{T}^+| &= \min(990, |\{\text{all positive samples}\}|) \\ |\mathcal{T}^-| &= \min(1800, 9 \times |\mathcal{T}^+|) \quad . \end{aligned}$$

The choices for these training set sizes were not systematically optimized but have produced good performance in other experiments [1]. Given the reduced training set $\mathcal{T} = \mathcal{T}^+ \cup \mathcal{T}^-$, we perform a basic parameter optimization via grid search using LibSVM [8], and train nine SVMs using the learned parameters. For each we use different training sets by resampling the training data using the proportions of positive and negative examples above.

The SVM and projection methods perform best in terms of average MAP (Avg. column) over the set of weak classifiers. This is likely due to their use of the complete feature data in each tree or classifier. However, the feature subset methods perform better in terms of mean fusion MAP (MF column). This demonstrates that the individual trees capture diverse information regarding the tags, and that bagging effectively integrates this information. The individual SVMs all use the same features and many of the same positive exemplars. As a result, bagging offers more limited performance gains with the SVMs. RS and RP do not significantly outperform the disjoint default and spatial feature subsets in combination. As a result, we use the disjoint subsets for simplicity in the remaining experiments.

4.2 Incorporating boosting

In the second experiment, we use RankBoost to combine our weak classifiers. We use the trees constructed in the first experiment (from 28,157 photos). The boosted classifiers are learned using corresponding weak annotations for the test data from the first experiment (28,158 photos). We use an additional set of 85,997 photos for testing. The time complexity for constructing the spatial trees appears in the second column from the left of Table 2. These times are cumulative for all trees, and are *independent of the number of tags*. The test times for weak annotation are averaged over the number of tags and number of photos to indicate the time complexity of weak annotation for a single tag-photo pair. To be clear, these average times are for weak annotation of the final test set of 85,997 photos. We have not

Table 2: Time complexity and MAP results for boosted classifiers: The weak annotation trees are constructed using a set of 28,157 photos. The boosted classifiers are trained using a set of 28,158 photos. Training times are averaged over 42 tags and indicate the average time for learning a boosted classifier for a single tag. Testing times in both cases are averaged over 85,997 test photos and 42 tags. All times are in seconds.

Trees (#)	Weak Annotation		Boosted Annotation		MAP
	Training	Avg. Testing	Avg. Training	Avg. Testing	
Def. EHST (17)	54.558901	0.0045757	15.840919	0.000126	0.052416
Spa. EHST (25)	54.714945	0.008531	21.433983	0.000171	0.060359
Def. CLRM (6)	18.516154	0.0018630	5.265293	0.000051	0.035054
Spa. CLRM (25)	20.086161	0.008190	17.624727	0.000153	0.040994
Def. EHST + Def. CLRM (23)	73.07505	0.0064387	16.347376	0.000136	0.053896
Def. EHST + Spa. CLRM (42)	74.645061	0.0127657	9.284721	0.000143	0.059515
Spa. EHST + Def. CLRM (31)	73.231099	0.010394	22.612014	0.000192	0.0617867
Spa. EHST + Spa. CLRM (50)	74.801106	0.016721	21.556980	0.000209	0.0676825
ALL (73)	147.876156	0.0231597	16.531516	0.000284	0.070676
SVM					
			Avg. Training	Avg. Testing	MAP
EHST			1364.280289	0.0237984	0.044572
CLRM			181.19786	0.005647	0.048438
EHST+CLRM			1545.478149	0.0475968	0.054360

included the times for weakly annotating the fusion set of 28,158 photos here as this processing can be done offline; that cost is readily estimated from the results here. Generally speaking, the cost is under 0.5ms per tree or weak annotation, and under 20ms per photo for the set of all weak annotations for a single tag.

We also consider combinations of the edge (EHST) and color moment (CLRM) features for categorization. Generally, the edge features produce improved performance with additional cost due to their higher dimensionality relative to the color features. Table 2 shows the training and testing complexity for boosted annotation. To clarify, when combining edge and color features, a single boosted classifier is learned per tag from all available trees. A hierarchical alternative is to learn a separate boosted classifier for each feature modality, and then combine them in a second boosting step. We have left this option for future work. Likewise, for the results using "ALL" trees, a single boosted classifier is learned per tag that combines the 73 possible weak classifiers. Classifier training is around 20 seconds per tag, and testing is on the order of 0.1ms per photo and tag. In all, testing per tag-photo pair requires 5ms-25ms. We include the results of baseline per-tag SVM classifiers constructed as in Section 4.1. The combination of neighborhood-based weak annotation and boosting outperforms the bagged SVMs which require significantly more training time, and about twice as much testing time. Note that the training time for the weak classifiers (tree construction) is a tag-independent total, while the SVM (and boosted classifier) training times are averaged per tag.

4.3 Mediamill challenge

The Mediamill challenge [29] is a benchmark video concept detection data set derived from the TRECVID video retrieval evaluation. We use it here for image annotation experiments with 101 concepts or tags that have been manually annotated. The first challenge experiment trains per-tag SVM visual detectors that are optimized using cross validation to compare a "large number of SVM parameter combinations." Their performance provides a high performance per-tag benchmark on a data set and tag set of reasonable scale. We compare our approach directly against the benchmark by randomly splitting the challenge training set into two partitions. The first is comprised of 22,625 keyframes and is used to construct spatial trees for nearest neighbor search. The second contains 8,368 shots and is used to train per-tag boosted classifiers via RankBoost. Results appear in Table 3 and show that our method performs competitively. We include MAP results for both the full 101 tag challenge and the smaller lscom-lite tag subset [24]. While we have designed our approach to scale to larger training and test sets, it achieves competitive performance on this smaller data set without requiring costly off-line training or parameter optimization. We also use standard low-level visual features in contrast to the specialized features of [29] which are detailed in [34]. Although the data set is smaller, the results demonstrate both efficiency and accuracy over a larger set of tags.

4.4 Scalability

Finally, we examine complexity and performance tradeoffs as the number of training images used for model-free annotation is varied. The number of training photos used in the construction of the spatial trees for weak classification ranges roughly two orders of magnitude from 24,999 to 2,186,736. Over this range, the cost of tree construction scales approximately linearly. Again this cost is amortized over all tags considered, and wholly accounts for the added complexity with scale. Tree search (weak annotation) doesn't increase significantly, due to both the design of the spatial trees and the use of a truncated priority search. Because the boosted classifier uses the same set of photos (with different weak annotations) for training, its computational costs also are stable. Thus test complexity for the entire system is largely unchanged. The additional per-tag training cost incurred by RankBoost is substantially less than conventional per-tag classifiers.

At the same time, MAP shows consistent improvement

Table 3: Experimental results using the Mediamill Challenge data set: The weak annotation trees are constructed using a set of 22,625 photos. The boosted classifiers are trained using a set of 8,368 photos. Training times are averaged over 101 tags and indicate the average time for learning a boosted classifier for a single tag. Testing times in both cases are averaged over 12,914 test keyframes and 101 tags. The MediaMill benchmark is MAP=0.216 for 101 concepts and MAP=0.260 for the 36 lscom-lite concepts. Times are in seconds.

Trees (#)	Weak Annotation		Boosted Annotation		MAP	MAP
	Train	Avg. Test	Avg. Train	Avg. Test	101	LSCOM
EHST (17)	40.750191	0.005036	1.275239	0.000245	0.124785	0.169756
Spa. EHST (25)	42.5561	0.007856	1.620391	0.000401	0.148716	0.183475
CLRM (6)	13.485891	0.0017304	0.866012	0.000104	0.10787	0.158066
Spa. CLRM (25)	17.5564	0.007693	1.724719	0.000373	0.12814	0.171708
Spa. EHST + Spa. CLRM (50)	60.1125	0.0155497	2.741734	0.000196	0.18844	0.238749
ALL (73)	114.348582	0.0190105	3.674249	0.000260	0.20343	0.250324

with the use of additional training data for weak annotation. The increase in MAP using "ALL" trees is 14.92% when increasing from 24,999 to 249,988 training images, and 8.98% when increasing from 249,988 to 2,186,736 training images exhibiting a log-linear trend. Table 5 shows detailed per-tag performance results for these three training set scales.

5. CONCLUSION

We have presented an approach to image categorization using approximation methods to scale neighborhood-based classification to higher dimensional features and larger training sets without sacrificing efficiency. Incorporating boosting for lightweight per-tag modeling enables our approach to perform competitively with per-tag modeling using SVMs. Our short term future work will examine performance and efficiency tradeoffs with various parameter settings. A variety of other choices or combinations of low-level features are also worth evaluating. We believe that our approach is far from optimized in the annotation context and substantial improvements remain possible.

There are several hierarchical variations of the method that are of interest. One first uses boosting to combine trees within a feature modality. The second step would combine the per-tag boosted classifiers from each modality in a second per-tag boosting step. In a complementary extension, we can use boosting to examine inter-tag relationships. As shown in [33, 39], boosting naturally lends itself to this objective. It is also interesting to consider the ability of different feature modalities to capture such inter-tag relationships. Boosting's application as a feature selection mechanism can be exploited here [31]. Because the design of our weak classifiers does not depend on the number of tags, considering additional feature modalities and larger tag sets remains computationally tractable.

6. REFERENCES

[1] J. Adcock, M. L. Cooper, and J. Pickens. Experiments in interactive video search by addition and subtraction. In *ACM Conf. on Image and Video Retrieval*, 2008.

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

[3] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: An embedding method for efficient nearest neighbor retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):89–104, Jan. 2008.

[4] K. Barnard, P. Duygulu, N. d. Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[5] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, volume 20, Cambridge, MA, 2008. MIT Press. to appear.

[6] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[7] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(3):394–410, March 2007.

[8] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, volume II, pages 886–893, 2005.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.

[11] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. E. Abbadi. Approximate nearest neighbor searching in multimedia databases. *Data Engineering, International Conference on*, 0:0503, 2001.

[12] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, New York, NY, USA, 2003. ACM.

[13] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, November 2003.

[14] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, March–April 2009.

Table 4: Time complexity and performance variations with scale: The weak annotation trees are constructed using the number of photos indicated in the "Training Images" column. The boosted classifiers are trained using a set of 90,062 photos. Training times are averaged over 42 tags and indicate the average time for learning a boosted classifier for a single tag. Testing times in both cases are averaged over 90,029 test photos and 42 tags. All times are in seconds.

Trees (#)	Training Images	Weak Annotation		Boosted Annotation		MAP
		Train	Avg. Test	Avg. Train	Avg. Test	
Spa. EHST (25)	24,999	47.45025	0.007315	55.665894	0.0002033	0.066846
Spa. CLRM (25)	24,999	18.39254	0.006963	60.398320	0.0001891	0.0594876
Spa. EHST + Spa. CLRM (50)	24,999	65.84278	0.014278	66.178117	0.0002763	0.0848826
ALL (73)	24,999	132.9060	0.020798	77.822631	0.0005982	0.0883646
Spa. EHST (25)	249,988	502.15563	0.007894	50.663604	0.000127	0.080459
Spa. CLRM (25)	249,988	195.5374	0.0072788	51.291555	0.000173	0.072695
Spa. EHST + Spa. CLRM (50)	249,988	697.6930	0.0151728	44.943356	0.0002424	0.103808
ALL (73)	249,988	1353.9565	0.022811	66.874881	0.0002950	0.1038658
Spa. EHST (25)	499,971	1085.7674	0.007997	53.621025	0.000221	0.083982
Spa. CLRM (25)	499,971	423.6503	0.00731	55.036825	0.0002091	0.0758579
Spa. EHST + Spa. CLRM (50)	499,971	1509.4177	0.015306	63.365708	0.0002857	0.107585
ALL (73)	499,971	2916.4976	0.023452	80.019395	0.0004684	0.107909
Spa. EHST (25)	999,943	2697.4888	0.008423	45.733651	0.0001693	0.086893
Spa. CLRM (25)	999,943	966.36723	0.007382	46.611440	0.000194	0.077701
Spa. EHST + Spa. CLRM (50)	999,943	3663.8560	0.015805	60.601824	0.000265	0.110819
ALL (73)	999,943	7135.4805	0.024707	92.323878	0.000521	0.111027
Spa. EHST (25)	2,186,736	6158.5278	0.09044	35.470965	0.000183	0.089818
Spa. CLRM (25)	2,186,736	2087.448867	0.007595	35.977581	0.000179	0.078994
Spa. EHST + Spa. CLRM (50)	2,186,736	8245.9767	0.016639	48.955567	0.0002608	0.113434936
ALL (73)	2,186,736	15937.9699	0.027226	72.350157	0.0005344	0.114114

[15] A. Hauptmann, R. Yan, and W.-H. Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634, New York, NY, USA, 2007. ACM.

[16] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[17] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, New York, NY, USA, 2003. ACM.

[18] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 607–610, New York, NY, USA, 2006. ACM.

[19] X. Li, C. G. M. Snoek, and M. Worring. Learning tag relevance by neighbor voting for social image retrieval. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval*, pages 180 – 187, Vancouver, Canada, October 2008.

[20] X. Li, C. G. M. Snoek, and M. Worring. Annotating images by harnessing worldwide user-tagged photos. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009. Invited paper.

[21] X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: a text-like paradigm. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 603–610, New York, NY, USA, 2007. ACM.

[22] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching, version 1.1.1. http://www.cs.umd.edu/∼mount/ANN/.

[23] M. Muja and D. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, Intl. Conf. on Computer Vision Theory and Applications (VISAPP'09), 2009

[24] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia Magazine*, 13(3), 2006.

[25] M. R. Naphade and J. R. Smith. On the detection of semantic concepts at trecvid. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 660–667, New York, NY, USA, 2004. ACM.

[26] A. P. Natsev, M. R. Naphade, and J. TešiĆ. Learning the semantics of multimedia queries and concepts from a small number of examples. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 598–607, New York, NY, USA, 2005. ACM.

[27] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

Table 5: Average precision results for 42 image tags using a varying number of training images. The numbers along the top of the average precision columns indicate the number of training images used for weak annotation. Boosted classifiers using all weak annotations correspond to the MAP results for "ALL" from Table 4. The tags are shown after Porter stemming.

Tag	AP			Tag	AP		
	24,999	**249,988**	**2,186,736**		**24,999**	**249,988**	**2,186,736**
anim	0.081481	0.096594	0.110258	macro	0.123325	0.139275	0.143091
architectur	0.035352	0.039980	0.045584	mountain	0.136290	0.147944	0.163417
babi	0.117361	0.169280	0.182572	natur	0.075036	0.076707	0.082119
beach	0.188340	0.201443	0.213970	night	0.105877	0.111529	0.113912
bird	0.024505	0.035879	0.049002	nyc	0.043537	0.040399	0.039467
birthdai	0.093747	0.130459	0.139600	ocean	0.093425	0.099807	0.104752
blue	0.075480	0.094737	0.102215	parti	0.089457	0.110014	0.118615
boat	0.033744	0.038895	0.041090	plant	0.079647	0.089646	0.091924
build	0.098258	0.119328	0.146527	river	0.099091	0.117856	0.126610
camp	0.026961	0.029140	0.031056	rock	0.027850	0.034165	0.039242
cat	0.122522	0.166816	0.214402	sea	0.094633	0.105338	0.110730
christma	0.055095	0.062296	0.064471	sky	0.127660	0.151477	0.174895
church	0.068088	0.098914	0.105101	snow	0.130771	0.181596	0.209089
citi	0.021497	0.022907	0.024374	street	0.038287	0.044543	0.050583
cloud	0.170389	0.219155	0.257655	sun	0.055298	0.067603	0.068891
festiv	0.053129	0.059857	0.065561	sunset	0.140749	0.179592	0.192572
flower	0.389931	0.413503	0.453333	tree	0.080487	0.088778	0.098177
garden	0.111321	0.115796	0.122873	urban	0.045110	0.044170	0.046220
hike	0.065554	0.078089	0.095323	water	0.092591	0.106896	0.111553
lake	0.028127	0.030593	0.032570	wed	0.068339	0.089879	0.097345
light	0.054506	0.057485	0.056840	white	0.048465	0.054005	0.055205

[28] C. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Trans. on Multimedia*, 9(5):975–986, Aug. 2007.

[29] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM Press.

[30] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, July 2006.

[31] K. Tieu and P. A. Viola. Boosting image retrieval. *International Journal of Computer Vision*, 56(1-2):17–36, 2004.

[32] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Macihine Intelligence*, 30(11):1958–1970, Nov. 2008.

[33] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, May 2007.

[34] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders. Robust scene categorization by learning image statistics in context. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 105, Washington, DC, USA, 2006. IEEE Computer Society.

[35] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Scalable search-based image annotation. *Multimedia Systems*, 14(4):205–220, 2008.

[36] D. Wang and M. Cooper. Image orientation detection using scalable non-parametric classification. *Pattern Analysis and Applications*, (In preparation) 2009.

[37] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *IEEE. CVPR 2006*, pages II: 1483–1490, 2006.

[38] R. Yan, M.-Y. Chen, and A. Hauptmann. Mining relationships between concepts using probabalistic graphical models. In *Proc. IEEE ICME*, 2006.

[39] R. Yan, J. Tesic, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 834–843, New York, NY, USA, 2007. ACM.

[40] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.