To Search or To Label?

Predicting the Performance of Search-Based Automatic Image Classifiers

Lyndon S. Kennedy
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
lyndon@ee.columbia.edu

Shih-Fu Chang
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
sfchang@ee.columbia.edu

Igor V. Kozintsev
Intel Labs
Intel Corporation
Santa Clara, CA 95052
igor.v.kozintsev@intel.com

ABSTRACT

In this work we explore the trade-offs in acquiring training data for image classification models through automated web search as opposed to human annotation. Automated web search comes at no cost in human labor, but sometimes leads to decreased classification performance, while human annotations come at great expense in human labor but result in better performance. The primary contribution of this work is a system for predicting which visual concepts will show the greatest increase in performance from investing human effort in obtaining annotations. We propose to build this system as an estimation of the absolute gain in average precision (AP) experienced from using human annotations instead of web search. To estimate the AP gain, we rely on statistical classifiers built on top of a number of quality prediction features. We employ a feature selection algorithm to compare the quality of each of the predictors and find that cross-domain image similarity and cross-domain model generalization metrics are strong predictors, while concept frequency and within-domain model quality are weak predictors. In a test application, we find that the prediction scheme can result in a savings in annotation effort of up to 75%, while only incurring marginal damage (10% relative decrease in mean average precision) to the overall performance of the concept models.

Categories and Subject Descriptors

H.3.1 [**Information Search and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Performance, Experimentation

Keywords

performance prediction, search-based concept models

MIR'06, October 26–27, 2006, Santa Barbara, California, USA.

1. INTRODUCTION

The automatic annotation or tagging of visual concepts in image and video databases will be a key technology for managing and searching the multimedia collections of the future. Past research has laid out workable general framework for automatically classifying images for any arbitrary concept. The framework is shown in Figure 1. First of all, (Figure 1a), system designers must define a set of concepts to be detected. To learn a model for any particular concept, a number of positive and negative examples must be obtained. These examples are typically obtained by having humans annotate a set of training images as either having or not having the concept, though recent research suggests that it might be possible to substitute this process with simply automatically obtaining images from the web by searching for the concept using tools such as Google or Yahoo! image search and taking the returned results to be pseudo-positively labeled training images (Figure 1b). Once the training images are obtained and labeled, various low-level features, such as color distributions, textures, edge locations, or spatial-appearance features of interest points, are extracted (Figure 1c) and statistical classification models are learned over the low-level features using the training labels (Figure 1d). These models can then be applied to unseen images to provide automatic concept annotations (Figure 1e).

The primary focus of this work is an exploration of the trade-offs that occur when choosing an approach for training data acquisition (Figure 1b). On the one hand, we have *annotation-based models*, which use labels provided by human annotators. On the other hand, we have *search-based models*, which use images and labels acquired automatically through web search. The key trade-offs are the amount of human effort that is required in annotation-based models and the expected decrease in classification performance that will result from search-based models.

In a typical application, thousands of examples are needed to learn reliable models for a concept and a lexicon of hundreds or thousands of concepts is probably necessary to adequately annotate most multimedia collections. The task of acquiring enough manual annotations over a reasonably-sized lexicon of concepts is prohibitively expensive in terms of human labor. For example, in one recent effort, nearly 450 concepts were annotated over a collection of 62,000 video keyframes, at a cost of 60,000 hours of labor [2]. Furthermore, annotations made on one domain, such as consumer photos, are not likely to yield models which will general-

ize to other domains, like broadcast news, so annotation efforts may be need to be repeated for every new application. Search-based models provide an interesting solution to the cost of human labor in annotation-based models. Given just the name of the concept to be annotated, such as "car," "bicycle," or "Statue of Liberty," we can feed the name into a web image search engine and take the top returned results to be pseudo-positive examples of the concept, while pseudo-negative examples can be sampled randomly [6, 11].

We are motivated by prior work which shows that search-based models may achieve good performance for some concepts, resulting in a major reduction of effort in manual annotation; however, the performance of search-based models can be quite poor for other concepts. We set out to determine when and why search-based models will succeed or fail and design a system for predicting the performance trade-off between annotation- and search-based models.

The variation in performance can be caused by a number of factors: web-based image searches may provide a lot of false-positives; true-positives may be visually quite different from the application domain; or the number of available images may simply be too small. All of these factors may be measured and used to predict the performance difference through metrics based on either the *data* or the *model*. *Data* quality might be measured by evaluating the number of available training examples and the similarity between the web image domain and the application domain. *Model* quality can be measured by performing cross-validation within the training set or by estimating the generalization from the training domain to the test domain.

We propose a framework, illustrated in Figure 2, for predicting the gain in performance from annotation-based models using the model- and data-quality prediction features described above. We define the gain in performance as the absolute change in average precision (AP) between search- and annotation-based models and learn the prediction model over a set of training concepts (with both annotation- and search-based models) using support vector machine (SVM) regression. Given a new concept, we can then extract the performance predictors and apply the learned model to recommend whether to "search" (apply the search-based model) or to "label" (apply an annotation-based model). We also apply a greedy iterative feature selection algorithm to evaluate the relative contribution of each of the prediction features, giving insight into the success and failure cases of search-based models.

We test the system over a large set of consumer photos using 15 named location concepts related to New York City. We find that prediction metrics related to cross-domain similarity between images and model generalization are most effective, while metrics related to concept frequency and within-training-set model quality are less effective. The prediction framework gives good performance. If we are limited with resources to only annotate 4 of the 15 concepts, the framework can predict the 4 most worthwhile concepts to annotate with 100% accuracy. This results in a decrease in annotation effort of almost 75% with only a relative loss of mean average precision (MAP) of 10%, when compared to annotating all 15 concepts.

In Section 2, we review prior work and summarize the proposed system. In Section 3, we describe the components and experiments. In Sections 4 and 5, we provide analysis and conclusions.

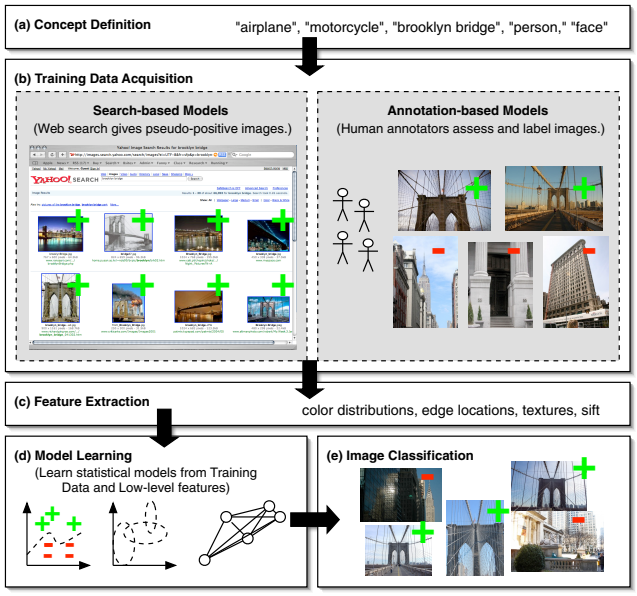

Figure 1: Framework for concept detection. The primary contribution of this work is a system for choosing Search- or Annotation-based models for component (b).

2. AUTOMATIC IMAGE ANNOTATION

2.1 Annotation-based Models

The success of annotation-based models has been aided in many ways by the NIST TRECVID evaluation [1], an annual benchmark which has provided an increasingly large test video dataset (up to 160 hours in 2006) upon which to test video analysis and retrieval algorithms using open metrics-based comparisons. Since its second year, 2002, one of the core components of the benchmark has been a "high-level feature extraction" task, which is essentially automatic annotation. Each year, automatic annotation systems are tested on 10-17 visual concepts over a labeled test set of tens of thousands of video shots. Many of the most successful systems in the recent benchmarks [3, 5] employ the basic framework for automatic concept detection shown in Figure 1. The concept definitions, annotated training data, and a standard test set (components (a), (b), and (e) in the figure) are all provided, so the major grounds for differentiation between various entries are the features used and the types of statistical models applied (components (c) and (d)). The results imply that a suite of features encapsulating color, texture, and edge information are most powerful and that SVMs make for powerful discriminative models. In particular, it also seems that learning models independently on individual feature spaces and performing late fusion on the results gives the best overall performance. The problem of detection in news video is very much scene-based, which is similar to the named-location concepts in our test and we may adopt these features and models for our task.

A major resource provided for this task, of course, is the extensive set of annotations provided over a development data set. In 2005, for example, a set of 39 concepts were annotated by hundreds of participants spread across dozens of research institutions worldwide [10]. In recent parallel ef-

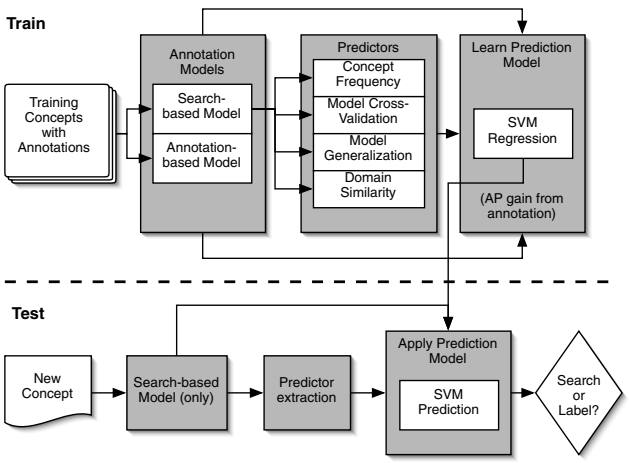

Figure 2: System architecture for predicting performance gain for annotation-based models based on model and data quality predictors. Final output can be used to recommend searching (search-based model) or labeling (annotation-based model).

forts, lexicons of a much larger magnitude — hundreds of concepts — have been designed and annotated across the TRECVID development set at the expense of thousands of hours of labor [2, 8]. The importance of the various annotated concepts can not be understated, of course, as automatic annotation using annotation-based models is still a research area, requiring much more work; however, the expense involved in obtaining realistic and reliable annotations is huge. There is a clear need for methods for acquiring training data with much less human interaction.

2.2 Search-based Models

One proposed solution to the human effort issues introduced by annotation-based models is, of course, the so-called search-based model. The diminished quality of detection results from search-based models has been well-observed by these earlier efforts and handled in some interesting ways.

In [6], the authors implement a search-based model using Google image search and a parts-based model to learn concept models from image search results over a set of concepts, such as "airplane," "car," and "motorbike." It is observed that the results returned by web image search are frequently noisy, containing many false-positives. It is also observed, however, that results at the top of the results lists (within the top 5 or so) are qualitatively "better," with fewer false-positives, than results further down the list. The results retrieved from the web are augmented by translating the concept name into various foreign languages and re-running the image search. The top results from each language are combined into a sort of validation set of images which are more likely to be positive than the other images. This set is then used to select out other images from the search results which are visually similar to the likely positives and therefore likely to positives themselves. So, by smoothing out the false-positive web images, the authors addressed the noisiness problem common to web image search engines and have observed an increase in detection performance. On the other hand, there is still a problem that can arise (though it did not in their experiments) of having typical web search results being highly different from the test set images due to domain differences. The model should also be able to compensate for such a scenario or at least recognize its failure and recommend the user to provide annotations.

In [11], the authors present a search-based model with a slightly different scenario wherein an image which has already been annotated with a single term can have its annotation expanded to many more descriptive terms using a combination of text-based web image search and content-based image search. A set of textually-related images is found from the web through web-based image search given the annotation term associated with the image. From the textually-related images, visually-related images are found through content-based image similarity between the image and the images found on the web. Once images that are both semantically-related and visually similar to the image are found on the web, the associated text on the pages containing the web images are mined to extract salient keywords related to the image. Those mined keywords are then propagated back to the original image as additional annotations. Through the joint use of text and visual features, the authors acknowledge and somewhat address the noisiness of web image searches, though there is still no framework for providing insight into situations in which this framework might fail significantly.

It is clear that the success of search-based models varies largely from concept to concept and in order to successfully deploy search-based system, we need to account for the noisiness of web image search results and the possible domain differences between the web and the test set.

2.3 Prediction System

We propose a system for predicting the difference in quality between search- and annotation-based models (Figure 2). Given a set of concepts with full annotations, we can build both annotation- and search-based models and observe the true difference in performance (in terms of AP). We have hypothesized that this difference in performance is due to the noisiness of the images obtained via web image search and the domain differences between the web and the application domain. We have designed a series of predictors, which can be extracted from search-based models, to measure the severity of these disadvantages. We then use SVM regression to learn to predict the ground-truth performance difference from just the predictors. Given a new, unseen concept, we can apply just a search-based model, extract the predictors and use the learned prediction model to predict the expected difference in performance from using an annotation-based model. This predicted performance change can be employed by a user to make a decision on whether or not to invest annotation effort for the new concept.

3. COMPONENTS AND EXPERIMENTS

To implement the performance prediction system, we construct a lexicon of concepts and gather and annotate a large corpus of real consumer photos and general web images. We learn annotation- and search-based models for each concept and engineer a set of metrics that can be used to predict the expected gain in performance that we might achieve by annotating a concept and choosing the annotation-based model instead of the search-based model. Each of these components is described in the following section.

Concept	#(Web)	#(Cons.)	Prec.
bronx-whitestone br.	113	6	1.00
brooklyn bridge	68,059	8,268	0.38
chrysler building	18,515	1,711	0.65
columbia university	163,159	1,572	0.30
empire state building	67,370	9,597	0.18
flatiron building	2,620	252	0.70
george washington br.	7,521	1,057	0.48
grand central	31,956	2,085	0.37
guggenheim	26,683	5,232	0.21
met. museum of art	20,806	2,260	0.02
queensboro br.	625	267	0.38
statue of liberty	48,768	6,631	0.49
times square	70,188	14,136	0.56
verrazano narrows br.	412	89	0.66
world trade center	140,775	4,852	0.13

Table 1: Summary of lexicon of concepts used, including count of matching images found through Yahoo! image search [#(Web)], frequency in consumer collections on Flickr [#(Cons.)], and precision of consumer tags [Prec.].

3.1 Experimental Data

To evaluate the system, we have carefully engineered a collection of concepts to detect, using influences from a number of different sources. We have also amassed a large testing set by leveraging Flickr, a site for sharing personal digital photos online. These components are explained in greater detail in the following sections.

3.1.1 Lexicon Definition

We test the system using a collection of 15 visual concepts pertaining to specific locations in New York City, shown in Table 1. The concepts are selected using influences from a number of sources, such as part-meronyms in WordNet [7], the results of a "related tags" function on Flickr, and a survey of human subjects.

The reader may notice that our choice of named locations as a lexicon of concepts is at odds with the common practice in the computer vision community of using generic objects, such as motorcycles, cars, and airplanes. We make this choice deliberately, since the prospect of obtaining training data for free from the web in some ways turns the task of visual concept lexicon design on its head. In the past, we have been limited by our ability to obtain training data, which has led us to choose generic concepts which may be more practical when annotation resources are limited. However, if we are able to freely obtain noisy annotations for a potentially infinite number of concepts, then it suddenly becomes reasonable to target specific objects and locations, which are less visually diverse, and thus easier model, when compared to generic concepts. Furthermore, it should be fairly easy to map back from specific concepts to their more generic parent concepts, given a sufficient visual ontology: the Statue of Liberty, Michelangelo's *David*, and Rodin's *Thinker* are all obvious instances of "statue" and the Chrysler Building is clearly an instance of a building. On a final note, locations can also be very important for a number of practical application domains, particularly consumer photos.

3.1.2 Data Acquisition

In the course of this experiment, we have produced a rather sizeable and noteworthy test set. It consists of over 38,000 real consumer photos, which have been labeled for the presence or absence of 15 concepts. This is comparable in scope to the TRECVID 2005 high-level feature extraction task, which was conducted over approximately 46,000 video shots for 10 annotated concepts. We were able to acquire this data set rather cheaply by leveraging the power of Flickr by searching for images tagged with words corresponding to the names of concepts in our lexicon. Once we have downloaded several thousand images which have been tagged by Flickr users with a particular concept name, we refine the labels provided by the Flickr users by annotating the subset of images for the presence or absence of the visual concept.

Interestingly, the tags provided by Flickr users are shown to be very imprecise. When the image is tagged with a visual concept by a Flickr user, there is a roughly 50/50 chance that the concept actually appears in the image. This lack of precision seems to come from many sources. In some cases, there are images of professional basketball games or landmarks in Washington D.C. tagged with things like "Brooklyn Bridge" or "Statue of Liberty." These cases are clearly examples of wrong labels, probably induced by improper use of batch labeling tools on the part of the user. On the other hand, there are cases where the disagreement in tags and annotations is more subtle, due to a difference in concept definition, in terms of specificity and granularity, rather than an outright error. Some examples of this case might be shots taken from the observation deck of the Empire State Building being tagged as "Empire State Building," whereas we require that the Empire State Building concept contain shots of the physical building. This lack of precision (or in some cases, simply specificity) on the part of Flickr users indicates a need for automated tools to assist in the annotation and indexing of consumer images.

We also use automated means to acquire images for training search-based models. To do this, we simply feed the concept name as a query into Yahoo! image search. We take the results to be positive examples, without providing any further refinement or annotation. Yahoo! (and any other major image search engine) gives a maximum of 1000 images in the results. After accounting for incorrect URLs and moved images, the figure is closer to 600 or 700 results per query. We choose both Yahoo! and Flickr to obtain images since both offer application programming interfaces, which make programmatic image acquisition easy.

3.2 Concept Models

To create concept models for the items in our lexicon, we adopt a rather standard and straight-forward approach. Given a set of positive or negative examples (obtained by either automated search or manual annotation), we extract a set of three different feature spaces (color, texture, and edge) representing the content of the image. We then learn the mapping from low-level features to high-level annotations with SVMs. Three different SVM models are learned, all over the same training data, but independently for each feature space. Each SVM model is then used to classify any given test image as having (or not having) the specified concept. The scores from each independent SVM model (the distance from the separating hyperplane) are then normalized and averaged to give a final, fused score specifying whether or not the given concept is present in the image.

3.2.1 Features

We use a simple set of color, texture, and edge features to represent the images at a low level. The color features used are grid color moments. The image is segmented into a grid of 5-by-5 equally-sized sections. The first three moments of the distribution of values in each of the three channels (in LUV color space) are then calculated, yielding a 225-dimensional feature vector for each image. The texture features used are Gabor textures, represented by the mean and variance of gabor features in 4 scales and 6 orientations over the grayscale image, yielding a 48-dimensional feature vector. The edge features are represented by an edge direction histogram, which is calculated by finding all edge pixels in an image using the Canny edge detector and then counting edge pixels along various angles from the center of the image. The angles are binned into a 72-dimensional histogram, with an additional bin to count non-edge points, yielding a 73-dimensional feature vector. These three feature sets are chosen since they have been shown to work well for image classification in the past and they are particularly good at characterizing specific scenes, which is very appropriate for our set of location-based concepts.

3.2.2 Models

We use an SVM classifier to learn models for each given concept independently over each of the three available feature spaces and use the independent models to give three predictions of the presence or absence of the concept in unseen images. The three component predictions are combined by a late fusion by simply averaging their scores, giving a final combined score. In principle, feature spaces may be combined prior to learning the SVM model (early fusion) or the late fusion may be more successful with a weighted average; however, we take the score averaging approach since late fusion tends to work better in visual concept modeling and score averaging is a simple approach to late fusion with no parameters to be selected. The training data is largely skewed towards negative samples. We address this by taking all available positively-labeled examples, while the negative examples are randomly chosen to be equal in number to the positive examples. This concept modeling approach (including the features, the use of SVMs, and the decision to use late fusion) is chosen largely due to its proven effectiveness in many other concept modeling applications [5, 3]. Additionally, it has the added benefit of being exceptionally easy to implement using easily downloaded components [4] and we find that the models can typically be trained in a matter of minutes on regular single-CPU PCs.

3.2.3 Training Examples

The quality of the models is highly dependent on the quality of the available training images and their labels. For each concept, we learn models over three different sets of training data: manually annotated images, search results from the web, and roughly labeled images.

In the first case, we implement an annotation-based model by taking the images gathered from Flickr, which have been manually re-annotated to ensure the best possible training data. We learn models over this data using three-fold cross-validation: the set is sectioned into three sets, we train on two of the sets and use the models to classify the left-out set; the process is repeated three times to gather annotation scores over each of the three sets. This should yield the best-possible models since we are using perfectly annotated data from within the same domain as the test set.

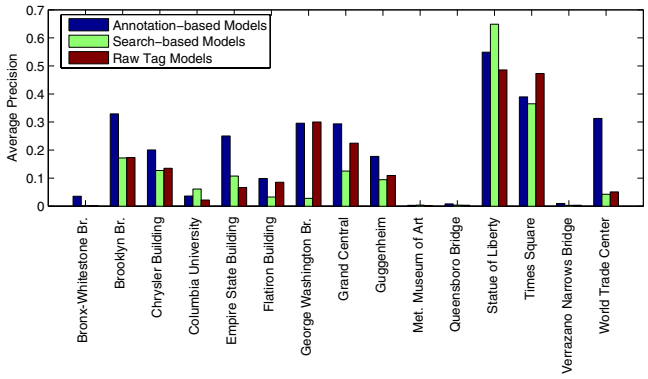

Figure 3: Resulting annotation accuracies for the concepts using annotation-based, search-based, and raw tag-based models.

In the second case, we implement a search-based model by taking the results of a Yahoo! image search to be noisy (but free) training data. The model is learned over the entire set of results returned by the image search and is expected to be inferior to the annotation-based model, due to the noisiness of the training data and expected differences in appearance between the domains of the training and test sets.

In the final case, we implement something in between an annotation-based model and a search-based model by taking roughly annotated images from Flickr (by simply trusting the unreliable tags given by users) as a set of sub-optimal training data. The models are again learned using three-fold cross-validation. Again, the set is sectioned into three sets, we train on two of the sets and use the models to classify the left-out set; the process is repeated three times to gather annotation scores over each of the three sets. This approach should yield an inferior model when compared to the annotation-based model: the training data is much noisier; however, the data is from the same domain as the test set, so the performance may or may not improve upon the search-based model.

Figure 4 shows the top most likely images for two of the concepts as predicted by the learned models. Figure 3 shows the performance of each of the models over the set of concepts, expressed in terms of non-interpolated average precision (AP) at full depth, a common information retrieval metric which approximates the area underneath the precision-recall curve. We can see that the general trend is that the annotation-based model significantly outperforms the search-based model, though, the magnitude of difference varies from concept to concept and there are a few cases (like Statue of Liberty and Columbia University) where the search-based model performs even better. We also see that the search-based model and the rough labels model have fairly similar performance, indicating that, in this case, the examples from a different domain (the web) do not seem to affect performance with any significant consistency.

3.3 Performance Predictors

In the previous section, we observed that the performance of search-based models is, indeed, varied from concept to concept, when compared to annotation-based models. We

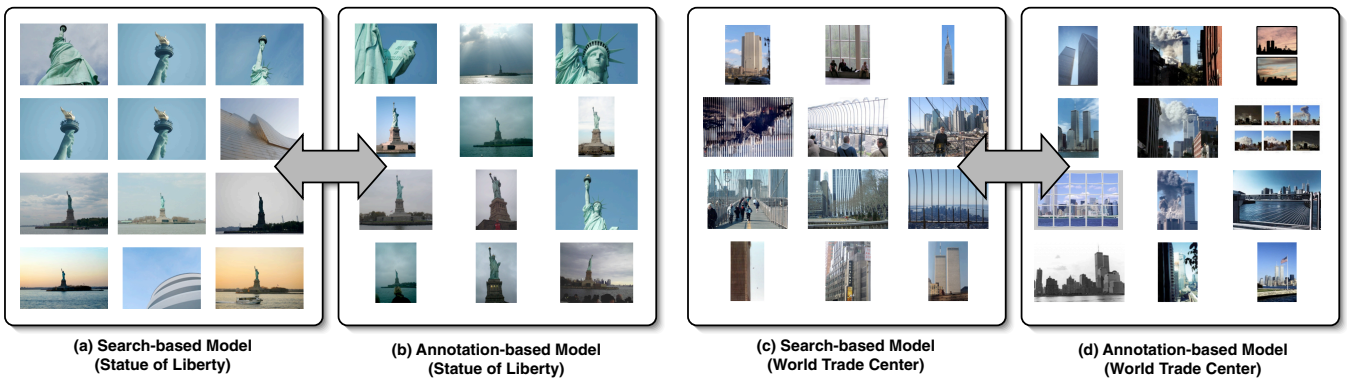

Figure 4: Top results for "Statue of Liberty" (strong search-based model) and "World Trade Center" (poor search-based model).

wish to design a set of metrics for predicting the performance gained by choosing an annotation-based model over a search-based model. These metrics, which are described in detail in the following section, include estimated frequencies of the concept on the internet, the quality of the model learned on the training data, and the estimated similarity between the training and test domains.

3.3.1 *Concept Frequency*

Perhaps the most obvious choice for a performance predictor is the frequency of the concept. We can use two easy methods to get an estimate on the frequency of a concept: taking the total number of matches found in a Yahoo! image search or the total number of images on Flickr tagged with the concept name (note that these are raw labels provided by users, not the explicit refined labels that we obtain through annotation). The frequency on the web may be indicative of the quality of the images which have been obtained. A concept with thousands of matches is likely to have many reasonable examples found in the top 1000 returned results, increasing the likelihood of obtaining reasonably visually consistent training examples, while a concept with only a few hundred matches is likely to have many poor-quality examples in the top 1000 returned results, making it much less likely that the training examples will be visually consistent. On the other hand, the frequency of the images on Flickr should be correlated with the performance of the models in terms of AP, since AP is biased by the number of true positives in the search set.

3.3.2 *Model Quality*

Another predictor of concept model performance is the estimated quality of the model, which can be estimated through cross-validation on the training set or through estimating how well the model generalizes to the test set.

Cross-validation on the training set is a fairly straightforward process. The results from a web image search are all taken to be positive examples. Three-fold cross-validation is then used to see how well a model trained on some of the web images can predict the labels for the left-out web images. The folds are determined randomly and the average precision resulting from this process is then taken as a predictor of the quality of the search-based model. A high AP from cross-validation would indicate strong consistency among the available training images, though it does not guarantee that the training images are at all similar to the test images.

An estimation of how well the model generalizes to the test set is then needed. Such an estimation is more difficult to make. Some recent work in the text retrieval community [12] has indicated that comparing the results of a multi-word text query against multiple other results returned by queries composed of each single word from the multi-word query is a good predictor of the relative difficulty of a text query. This is done be decomposing a multi-word query into many individual-word component queries. The results of the full multi-word query are then compared to the results of each component query. This comparison of results could be done a number of ways, but the intersection of the top-10 ranked documents in each query is the method used. High agreement between the full query and the component queries indicates a fairly successful retrieval result, while low agreement indicates a particularly difficult query with poor retrieval performance. The intuition, then, is that easier queries are composed of complimentary keywords, while more difficult queries have many outlying keywords.

We draw influence from this approach and offer a novel adaptation of it to the multimedia domain. In our case, we have a fused concept model which is represented by the average of three component models: the SVMs learned on individual feature spaces are analogous to single-term component searches and the fused model is analogous to the full multi-word search. We can compare the results of the fused model to the results of each component model to gain an estimate of the difficulty of the concept (or how well the concept was applied to the test set). Since this is still an open research area, we employ a number of different metrics to compare two result lists. The first is simply the intersection between the sets of the top 100-ranked images from each model, much like the top-ten intersection preferred for text retrieval. The second is the pearson correlation between the scores of each image in the test set given by each model. The third is the spearman rank-order correlation between the ranks of each image given by each model. And finally, we also adopt the average dynamic recall (ADR) [9], a metric first adopted by the music retrieval community which measures the performance of retrieval algorithms against multi-valued ground truth relevance labels and gives weight to the

highest-ranked documents. We take the combined model to be the multi-valued ground truth and measure the ADR of each component model against it. With each of these four list comparison metrics, we compare the results of the full fused model with the results of each of the three component models. For each metric, the average of the comparison scores across all three component models is taken as a model generalization predictor.

3.3.3 Domain Similarity

We also wish to include a set of predictors which will explicitly model the similarity between the training images found and the images contained in the test set. If the training images from the web are visually similar to the unlabeled images in the consumer collection, then the search-based model might be quite reliable. To calculate the domain similarity for each concept, we use the results from the web image search as well as the rough labels on the Flickr data (the raw tags provided by the users). We use each low-level feature space (color, texture, and edge) separately and measure the Euclidean distance between each positive image in the web results and each (roughly) positive image in the Flickr set. For each Flickr image, we assign a score of the distance to the closest image in the web results set. We then take the mean, variance, maximum, and minimum across these distance values in each of the three feature spaces to give 12 predictors of domain similarity.

3.4 Performance Prediction

Given our set of performance predictors (described in the previous section), we now need to learn how to predict the relative drop in performance caused by choosing a search-based model over an annotation-based one. We propose to learn these prediction models through support vector machine regression, using a gradient search for feature selection to gain some intuition about the most powerful predictors.

3.4.1 Learning Prediction Models

The final goal of our system is to predict the gain in performance caused by using an annotation-based model for any given concept instead of a search-based one. The prediction task can be defined in a number of ways: predicting the raw AP for the concept, predicting the percentage gain in AP, or predicting the absolute change in AP. We choose the latter approach since it provides the clearest correlation with the perceived difference in quality of the classification that the user might experience. Specifically, predicting the raw AP for the concept is undesirable since this gives us little insight into whether we should search or label. This just tells us the quality of the search-based model without any insight into the gains of an annotation-based one. The percentage gain in AP is slightly better, since it gives insight into the gains expected from annotation, but it can wrongly emphasize the wrong concepts. A change in AP from 0.2 to 0.4 is much more useful than a change from 0.002 to 0.004, though both are equivalent in terms of percentage changes. The absolute change, on the other hand gives proper emphasis to concepts with high-impact gains from annotating.

We can measure this difference in AP empirically using the true performance values for both the search- and annotation-based models over our set of 15 concepts. We see that the range of differences varies quite a lot, from a potential gain of over .25 in AP to a potential loss of almost .10. If we were to attempt to utilize a limited amount of annotation resources (only enough to annotate a subset of our available concepts), we would benefit from annotating those concepts with the highest potential for gain in AP. This would result in the highest gain in our final automatic annotation system.

We implement a system to predict this gain in average precision based on the predictors that we have available prior to investing the labor to acquire fully annotated training data. We do this using SVM regression, with the gains as the target values to predict and the various predictors (or a subset of those predictors) as the input space. Since our sample space for this learning task is rather small (only 15 concepts), we learn the prediction models using leave-one-out cross-validation, training a model on 14 concepts and testing it on the remaining single concept. We repeat this process, leaving each concept out once. The predicted performance-gain values can then be used to recommend which concepts will be most benefited by manual annotation.

3.4.2 Prediction Feature Selection

When learning performance prediction models, our input feature space (the 19 performance predictors) is large compared to our number of samples (only 14 concepts). Having a feature space which is too large for the number of samples can cause overfitting problems and result in seriously degraded performance and often selecting only the most discriminative features as the input space results in serious gains in performance. We therefore need to consider a feature selection method for choosing only a subset of the available performance predictors. This will help in ensuring a reasonably sized feature space and will give us insight into which predictors are most powerful. It may even be possible to discover useless predictors, which need not even be calculated, since their impact is marginal.

We implement feature selection using a gradient search, with oracle knowledge. This corresponds to the so-called wrapper model in which classifiers are placed in the loop of feature selection. In this method, we choose features one-by-one according to the resulting gain in performance that they give. So, in the first step, we train models using only one feature. We try all available features and select the one with the highest performance (in terms of mean squared error between the predicted and true values for the performance loss). In the second step, we try this best-performing feature in combination with each of the remaining features, choosing the single feature resulting in the best performance when used in tandem with the best-overall feature. We iterate through, always greedily selecting the next feature to be the one resulting in the highest gains in prediction accuracy. In the end, we have an ordered list of the most effective features and the resulting performances of the prediction models using varying numbers of features. This gives us insight into the power of various predictors and the limitations incurred by our small number of training examples.

4. ANALYSIS OF RESULTS

We conduct performance-loss prediction over our set of 15 New York City-related concepts and make recommendations for the use of annotation- and search-based models. The results, expressed in terms of Mean Average Precision ("MAP," or the mean of the average precisions for each of the 15 concepts in the evaluation set) versus the number of concepts using manual annotations, are shown in Fig-

ure 5. Recent research suggests that annotation quality is best when one concept is annotated at a time, so the actual number of hours invested is expected to be linear with the number of concepts annotated [10]. The results confirm that wisely selecting which concepts to annotate can result in significantly decreased annotation time, while causing minimal decrease in the average performance of the models. Furthermore, the results demonstrate that the performance vs. annotation time trade-off of our proposed prediction method closely approximates the ideal case. The performance is discussed in greater detail in Section 4.1.

Analysis of the results of the predictor feature selection confirm that many of the performance predictors that we have proposed are, indeed, useful for this task. In particular, we find that many of the predictors related to estimating the similarity between the training and testing domains are quite powerful. Also powerful are the predictors designed to test the quality of generalization of the learned model to test set. The qualities of the performance predictors are discussed in more detail in Section 4.2.

4.1 Performance

The actual and predicted changes in performance between the search- and annotation-based models are shown in Figure 6. In this case, the performance change is expressed as the absolute gain in average precision expected by choosing to conduct annotation. We can see from the figure that the predicted values are often erroneous, but the trend is quite good and we can fairly reliably predict the most and least useful concepts to annotate. Given only a finite amount of time to gather annotations, the system might recommend to gather annotations for concepts to the left of the threshold shown, while simply retaining the results of a search-based model (which has no annotation cost) for the concepts to the right of the threshold.

In Figure 5, we can see the implications of the annotation recommendation system in terms of time spent annotating and the resulting performance of the concept models. In our case, the time spent annotating is quantized into 15 values: we would annotate entire concepts, if they are recommended, and annotating an entire concept might cost 2 or 3 hours of labor. So, in one extreme case, we would annotate none of the concepts and constantly choose to trust the search-based model, yielding a MAP of .121. In the opposite extreme, we would annotate all 15 concepts, taking 30 or 40 hours of labor, and yielding a significantly increase MAP of .198. What we are really interested in though, are the in-between cases, where annotating only a few concepts will result in the highest positive impact on the performance. In the figure, we can see a baseline curve, which shows the effect of randomly choosing which concepts to annotate. The benefits in terms MAP are essentially constant with each added concept and this gives us a suitable lower bound for comparison. We can also see a baseline demonstrating the performance resulting from annotating concepts according to the positive change in AP, given prior true knowledge of the gains that will results. This is an upper bound. Interestingly, we see that fully annotating all 15 concepts does not result in the best possible performance, since search-based models actually perform better for several of the concepts. The curve demonstrating the results of our system lies in between these two bounds and is reasonably close to the best-case scenario for many of the highest-impact concepts.

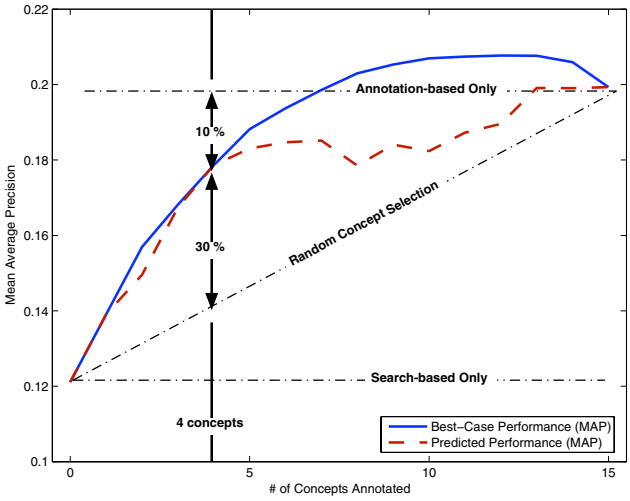

Figure 5: Impact on mean average precision of selecting varying numbers of concepts for annotation-based models. When the number of concepts annotated is 4, the automatic prediction system gives only a 10% relative decrease in MAP, when compared to annotating all 15 concepts, and a 30% relative increase compared to randomly selecting concepts for annotation. This is the best-case performance.

We can see that, given limited annotation resources, we can utilize the power of search-based models to supplement concept models for concepts that have no available annotation. And if we have good tools for predicting which concepts will benefit most from annotation (and which ones will perform equally well without annotation), then we can achieve performances on par with fully-annotated data, while also gaining significant savings in time expended on annotation. Our prediction method achieves very good results in recommending concepts for manual annotation. Out of the 15 concepts, if we are limited to an annotation budget of 4 concepts only, our method achieves 100% accuracy in selecting the most rewarding concepts which benefit most from manual annotation. Compared to a baseline method using random selection, our method is about 30% better in terms of the average AP (as shown in Figure 5).

4.2 Predictor Selection

Table 2 shows the order in which each performance predictor was chosen by the predictor selection algorithm along with the resulting decrease (or increase) in mean squared error. We see that many of the most powerful predictors are from the set of "domain similarity" metrics described in Section 3.3.3 along with the "quality of generalization" metrics described in Section 3.3.2. The power of these predictors indicates that much of predictive power necessary for this prediction task is encapsulated by estimators of the similarity between the training data and the test data. The domain similarity predictors try to estimate this similarity explicitly, while the model generalization metrics address it a bit more indirectly. Interestingly, edge features are most powerful for this domain similarity estimation. This may be due to the man-made structures which dominate our concept set. Color and texture might be more suitable for natural scenes.

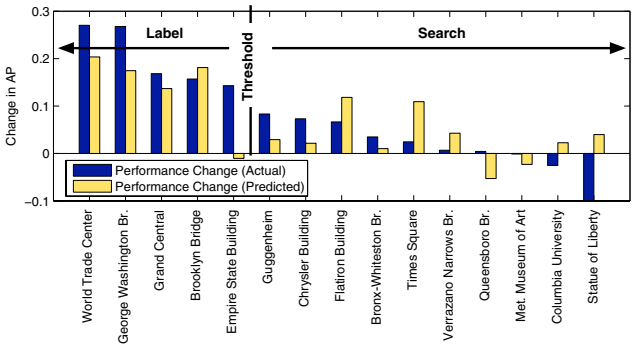

Figure 6: Actual and predicted increases in performance from using annotation-based models ordered by decreasing actual performance increase. A variable threshold can be used to recommend the use of annotation- or search-based models.

On the other hand, the less powerful predictors seem to be the ones expected to be related to raw performance of the concept models, themselves, such as the frequency counts described in Section 3.3.1 (though the frequency in the test domain (Flickr) is much more useful than the frequency in the training domain) or the training set cross-validation described in Section 3.3.2.

4.2.1 *Analysis of Predictor Strength*

The intuition, then, seems to be that the raw quality of the model over the training domain data is less important than the similarity between the training domain and the testing domain. If highly visually consistent images are obtained for a concept by web search, but the images are highly dissimilar from the types of images in the test set, then the result will be a high-quality model, which is unfortunately of little use in the application domain. The right way to estimate if the model will work, then, seems to be ensuring that the domains are similar (such as in the domain similarity metrics) and that the model learned is of high quality on the test set, which is estimated by the generalization quality metrics, as opposed to training set model quality, which is estimated by the training set quality metrics and the concept frequency.

4.3 Concept Evaluation

A final, important area for examination is the comparative performance of annotation- and search-based models on various concepts. Under what conditions do the models succeed or fail? Likewise, what situations cause the performance prediction framework to give erroneous predictions? This will give us insight into the ways in which search-based models need to be improved and areas in which performance prediction still needs work.

When evaluating the performance of the various search- and annotation-based models over various concepts, we can consider the concepts in roughly four different categories, summarized in Table 3. First, there are concepts for which both the search- and annotation-based models perform exceedingly well and we can save time by skipping the annotation process or the search-based model may be fused with the annotation-based model to augment performance (Table 3a). Concepts in this set include "Statue of Liberty" and "Times Square." Both of these concepts have high visual consistency and the types of images that appear on the web and in consumer photos are highly similar. Second, there are concepts for which both the search- and annotation-based models perform exceedingly poorly. Interestingly, these are a lot like the first case, since we can skip the annotation process as well: the expected increase in performance will be marginal (Table 3b). Concepts in this category include: "Bronx-Whitestone Bridge," "Columbia University," "Metropolitan Museum of Art," "Queensboro Bridge," and "Verrazano Narrows Bridge." These concepts tend to be visually very diverse, regardless of whether the images were drawn from the web or from exact annotations and it is very difficult to learn models for them. Third, there are concepts for which the search-based model actually performs reasonably well (with an AP greater than 0.1), but the annotation-based model is still much better, so we would stand to gain a lot by engaging some effort in annotating the concept (Table 3c). Concepts in this set include "Brooklyn Bridge," "Empire State Building," "Chrysler Building," "Grand Central," and "Guggenheim." In each of these, there are many visually consistent images in both the web and consumer photo domains, but there are also many different views of the location, which make it difficult to surmount the noisiness of the web images. Finally, there are the concepts which have poor performance from search-based models but very good performance from annotation-based models, giving the largest overall incentive to gather some annotations (Table 3d). Concepts in this set include "George Washington Bridge" and "World Trade Center." These suffer from the fact that the styles of images found on the web and in consumer collections are very difficult. An interesting side-note is that a few of the concepts, "Statue of Liberty" and "Columbia University," actually perform slightly *better* with search-based models. Examination of the results leads us to believe that this is

#	Category	Metric	MSE	ΔMSE
1	Domain	Edge - mean	0.0112	
2	Domain	Edge - var	0.0095	-0.0017
3	Quality	Intersect@100	0.0104	+0.0009
4	Quality	Pearson	0.0090	-0.0014
5	Freq.	Flickr	0.0095	+0.0005
6	Domain	Edge - max	0.0089	-0.0007
7	Domain	Color - max	0.0083	-0.0006
8	Domain	Texture - max	0.0074	-0.0009
9	Domain	Color - var	0.0075	+0.0002
10	Quality	Spearman	0.0071	-0.0005
11	Domain	Color -mean	0.0074	+0.0003
12	Quality	Cross-validation	0.0073	-0.0001
13	Domain	Edge - min	0.0070	-0.0003
14	Domain	Color - min	0.0068	-0.0002
15	Domain	Texture - min	0.0067	-0.0001
16	Quality	ADR	0.0066	-0.0002
17	Domain	Texture - mean	0.0064	-0.0001
18	Domain	Texture - var	0.0058	-0.0006
19	Freq.	Yahoo	0.0054	-0.0004

Table 2: Order of predictor feature selection, showing general categories as well as resulting mean squared error and change in mean squared error

	SBM	ABM	Label?	Concepts	Properties
a	High	High	Search	Statue of Liberty, Times Square	visually consistent across domains
b	Low	Low	Search	Bronx-Whitestone Br., Columbia Univ., Met. Museum of Art, Queensboro Br.	visually diverse regardless of domain, many view angles
c	Med.	High	Label	Brooklyn Br., Empire State Building, Grand Central, Guggenheim	visually consistent across domains, but many view angles
d	Low	High	Label	George Washington Br., World Trade Center	very different across domains

Table 3: Approximate performance of Search-Based Models (SBM) and Annotation-Based Models (ABM) for various concepts, with recommendations on whether to search or to label (use SBM or ABM, respectively).

due to the fact that for both of these concepts, web-based images are of similar quality to annotated images *and* the web search provides many more training examples than annotation, yielding higher-quality models.

Looking from a different perspective, we also want to evaluate the quality of our performance prediction on a concept-by-concept basis. We can also evaluate the prediction performance in the same four groups of concepts that we used for model performance. The most difficult case for our method to predict is the first case (Table 3a: both search- and annotation-based models have good performance, so there is little incentive to annotate). We do exceedingly poorly at predicting these concepts, this may be due to the fact that there are only a few of them, so we have insufficient data to accurately learn prediction models for these cases. In the second case (Table 3b: both search-based models and annotation-based models are poor), we perform quite well in prediction. These concepts seem to be well covered by the model quality and generalization predictors. In the third case (Table 3c: search-based models are good, but annotation-based models are much better), the prediction is hit or miss. For a few concepts, such as "Brooklyn Bridge" and "Grand Central," the prediction is excellent, while it is abysmal for "Empire State Building." Increasing diversity of views seems to negatively impact the prediction in these concepts. In the final case (Table 3d: search-based models are very poor, but annotation-based models are much better), the prediction is very good. These are characterized by visual coherence within the training and testing domains, but a general lack of coherence across the domains.

5. CONCLUSIONS

We have examined the task of automatically annotating images in a consumer photo library using models trained on positive and negative examples. We have paid particular attention to the implications of obtaining training images from web image searches (as opposed to collecting manual annotations) and have seen that such models tend to have decreased performance, though the magnitude of the decrease varies from concept to concept, while the savings in terms of annotation time are constant. We have, therefore, proposed a framework for predicting the performance loss due to the use of search-based models using only performance predictors which can be measured prior to gathering manual annotations. Using this framework, we can make recommendations on whether to trust the search-based models or to invest the time to gather manual annotations.

We have implemented the framework over a large set of consumer photographs for a set of 15 New York City-related concepts. We find that, indeed, the performance prediction system can result in significant savings in annotation time, while incurring only minor setbacks in the performance of automatic annotation. We analyze the contributions of the many proposed performance predictors and find the most powerful method of performance prediction to be based on estimating the similarity between the training and testing sets as well as the quality of the generalization of the model.

We also find some deficiencies in the prediction of concepts in which both the search- and annotation-based models perform very well and recommend balancing the set of training concepts in this respect to give more balanced performance.

6. REFERENCES

[1] NIST TREC Video Retrieval Evaluation http://www-nlpir.nist.gov/projects/trecvid/.

[2] LSCOM lexicon definitions and annotations version 1.0, DTO challenge workshop on large scale concept ontology for multimedia. Technical report, Columbia University, March 2006.

[3] A. Amir, J. Argillander, M. Campbell, A. Haubold, G. Iyengar, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer. IBM Research TRECVID-2005 Video Retrieval System. In *NIST TRECVID workshop*, Gaithersburg, MD, November 2005.

[4] 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.

[5] S.-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D. Zhang. Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction. In *NIST TRECVID workshop*, Gaithersburg, MD, November 2005.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from googles image search. *ICCV*, 2005.

[7] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[8] C. G. Snoek, M. Worring, D. C.Koelma, and A. W. Smeulders. Learned lexicon-driven interactive video retrieval. In *CIVR*, 2006.

[9] R. Typke, R. C. Veltkamp, and F. Wiering. A measure for evaluating retrieval techniques based on partially ordered ground truth lists. In *ICME*, 2006.

[10] T. Volkmer, J. R. Smith, and A. P. Natsev. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *ACM Multimedia*, 2005.

[11] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. *CVPR*, 2006.

[12] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR*, 2005.