Building text features for object image classification

Gang Wang[1] Derek Hoiem[2] David Forsyth[2]

[1] Dept. of Electrical and Computer Engineering
University of Illinois Urbana-Champaign (UIUC)
gwang6@uiuc.edu

[2] Dept. of Computer Science
University of Illinois Urbana-Champaign (UIUC)

Abstract

We introduce a text-based image feature and demonstrate that it consistently improves performance on hard object classification problems. The feature is built using an auxiliary dataset of images annotated with tags, downloaded from the internet. We do not inspect or correct the tags and expect that they are noisy. We obtain the text feature of an unannotated image from the tags of its k-nearest neighbors in this auxiliary collection.

A visual classifier presented with an object viewed under novel circumstances (say, a new viewing direction) must rely on its visual examples. Our text feature may not change, because the auxiliary dataset likely contains a similar picture. While the tags associated with images are noisy, they are more stable when appearance changes.

We test the performance of this feature using PASCAL VOC 2006 and 2007 datasets. Our feature performs well, consistently improves the performance of visual object classifiers, and is particularly effective when the training dataset is small.

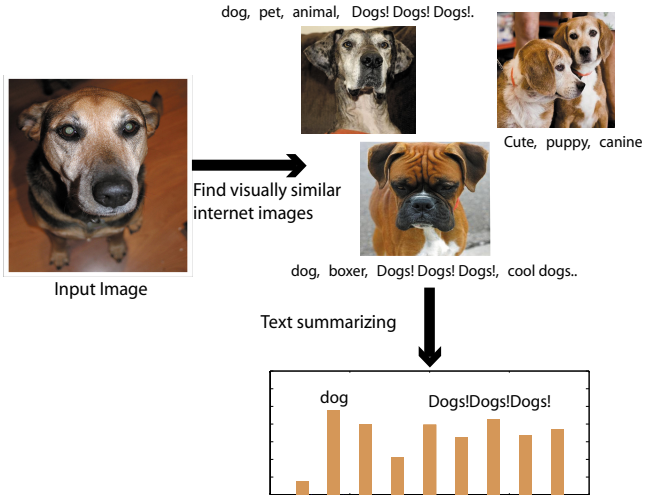

Figure 1. The illustration of our approach. For the input image, we find its similar internet images (downloaded from Flickr). The text associated with these internet images are summarized to build the text feature representation, which is a normalized histogram of text item counts. The Flickr text items can be tags such as "dog", and can be group names such as "Dogs!Dogs!Dogs!".

1. Introduction

With the advent of the digital camera and the popularity of internet photo sharing sites, we now have billions of images at our fingertips. But what can we do with them? These images are not annotated in a way that makes them easy to use for training data, but they are surrounded with a great deal of text, tags, and other supplemental information that are indicative of their content.

The main idea of this paper is to determine which objects are present in an image based on the text that surrounds similar images drawn from large collections. Object-based image classification is extremely challenging due to wide variation in object appearance, pose, and illumination effects. Low-level image features like color, texture, and SIFT [10] are far removed from the semantics of the scene, making it difficult to use them to infer object presence. If we had millions of training examples, these low-level features may be sufficient, but it is unrealistic to expect such large training sets for every object. On the other hand, we do have millions of internet images. While these are not labeled for our task, the text associated with them provides a more direct gateway to image analysis. The image feature representations that were too low-level for modeling objects from hundreds of images are sufficient for finding very similar images among millions. Their associated text can be transferred to our input image, making it easier to infer the scene content.

Our work builds on two insights. First, it is often easier to determine the image content using surrounding text than with currently available image features. State-of-the-art methods in computer vision [3] are still not capable of

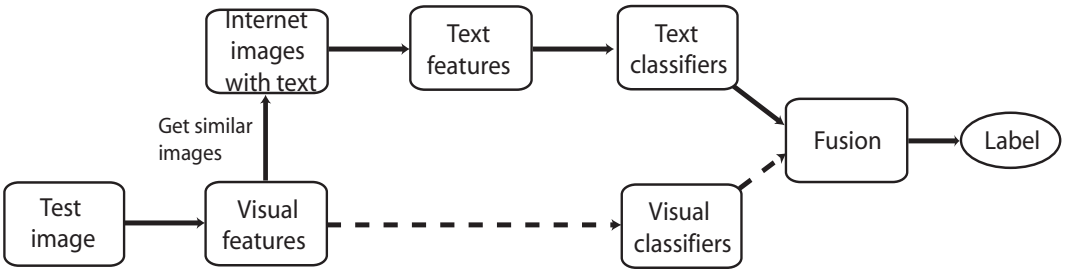

Figure 2. The framework of our approach. We have training and test images (here we only show the test image part). We also have an auxiliary dataset consisting of internet images and associated text. For each test image, we extract its visual features and find the K most similar images from the internet dataset. The text associated with these near neighbor internet images is summarized to build the text features. Text classifiers which are trained with the same type of text features are applied to predict the object labels. We can also train visual classifiers with the visual features. The outputs from the two classifiers are fused to do the final classification.

handling the unpredictability of object positions and sizes, appearance, lighting, and unusual camera angles that are common in consumer photographs, such as those found on Flickr. Determining object presence from the text that surrounds an image (tags, discussion, group names) is also far from trivial due to polysemy, synonymy, and incomplete or spurious annotations. Still, the text provides valuable information that is not easy to extract from the image features. The second insight: given a large enough dataset, we are bound to find very similar images to an input image, even when matching with simple image features. This idea has been demonstrated by Torralba et al. [14], who showed that matching tiny (32x32) images using Euclidean distance of intensity leads to surprisingly good object recognition results if the dataset is large enough (tens of millions of images). Likewise, Hays and Efros [7, 8] showed that simple image matching can be used to complete images and to infer world coordinates. Our approach is to infer likely text for our input image based on similar images in a large dataset and use that text to determine whether an object is present.

Others have attempted to leverage internet image collections to assist in image search [4, 1, 15] or recognition [5, 16, 9, 13, 12, 2]. The most common strategy is to improve annotation quality or filter spurious search results, gathering a new collection of images that can be used for training [5, 16, 9, 13]. While intuitive, this is a difficult way to use internet images because the noise or ambiguity in annotations can easily nullify any benefit resulting from the additional data. By contrast, we propose to use the online image collections to provide an alternate *representation* for our input image – one that we believe more directly reflects the semantics of the scene. Along these lines, Quattoni et al. [12] use captioned images to learn a more predictive visual representation. Our work is related to this in that we learn a distance metric that causes images with similar surrounding text to be similar in visual feature space. But our representation is ultimately textual, rather than visual, which we believe makes it more straightforward to infer object presence. Note that our textual representation is implicitly defined through visually similar images: no text is provided with the input image.

Section 2 presents our approach. The experimental results are shown in section 3. Some conclusions are made in section 4.

2. Approach

Our approach is to build text features for object image classification. The text features are expected to capture the semantic meaning of images and provide a more direct gateway to image analysis. Fig. 2 shows the feature extraction and classification procedure. We have a dataset with training and test images. We also have an auxiliary dataset of internet images (downloaded from Flickr), which have associated text. For each training image, we extract visual features and find its K nearest neighbor images from the internet dataset. Text associated with these near neighbor internet images is used to build the text features. Text classifiers are then trained on the text features. For a test image, we follow the same procedure to construct its text features, and use the trained text classifiers to predict the category labels. We also train a separate classifier on the visual features. We obtain the final prediction from a third classifier trained on the confidence values returned by the text and the visual classifiers.

2.1. Visual features

We use five types of features to find the nearest neighbor images and train visual classifiers.

The **SIFT** feature [10] is popularly used for image matching and object recognition. We use it to detect and describe local patches. We extract about 1000 patches from each image. The SIFT features are quantized to 1000 clusters and each patch is denoted as a cluster index. Each image is then represented as a normalized histogram of the cluster indices.

The **Gist** feature has been proven to be very powerful in scene categorization and retrieving [11]. We represent each image as a 960 dimensions Gist descriptor.

We extract **Color** features in the RGB space. We quantize each channel to 8 bins, then each pixel is represented as a integer value range from 1 to 512. Each image is represented as a 512 dimensional histogram by counting all the pixels. The histogram is normalized.

We also extract a very simple **Gradient** feature, which can be considered as a global and coarse SIFT feature. We divide the image to 4*4 cells, at each cell, we quantize the gradient to 16 bins. The whole image is represented as a 256 dimensional vector.

The **Unified** feature is a concatenation of the above four features. We learn weights for different feature types to make the unified feature discriminative. Write the four features introduced above as f_1, f_2, f_3 and f_4 respectively, our new feature is $[w_1 f_1, w_2 f_2, w_3 f_3, w_4 f_4]$, w_j is a non-negative number to indicate the importance of the jth feature.

We learn the weights from the training images. We aim to force the images from the same category to be close, and images from different categories to be far away in the new feature space. We randomly select N pairs of images from the training set. For the ith pair, $S_i = 1$ if the two images share at least one same object class, otherwise, $S_i = 0$. We calculate the chi-square distance with f_j for the ith pair as d_i^j. Then we learn feature weights by minimizing the following objective function:

$$\sum_i (e^{-\sum_j w_j d_i^j} - S_i)^2 \quad (1)$$

This optimization problem can be straightforwardly solved using the "fmincon" function in Matlab.

2.2. Internet dataset

The auxiliary internet dataset provides association between images and text. With this dataset, we can build text features for the images which do not have text by nearest neighbor matching.

The internet is rich in multimedia, and there is strong correlation between images and text. This is especially apparent in the photo sharing web sites such as Flickr: users tag images with some keywords, which usually describe the visual content of the images. Users also group images by the content. For example, there is a group called " Dogs! Dogs! Dogs!" which contains dog images. The group name becomes a very strong text cue to indicate the visual content of the images.

Our auxiliary dataset is collected from Flickr, and consists of about 1 million images. About 700,000 images are collected for 58 object categories, whose names come from PASCAL categories such as "car" or Caltech 256 [6] such as "penguin" and "rainbow". The other images are collected from a group called "10 million photos". These images are drawn from random categories.

2.3. Text features

Once the text features are extracted from the auxiliary dataset, they represent the image in a way that more directly reflects the semantics.

For each training and test image in our dataset, we find its K nearest neighbor images from the auxiliary dataset with the visual features. The text associated with these nearest neighbor images is extracted to build the text features. We treat each tag and group name as an individual item in our text feature representation, even though it may include multiple words. For example, the group name "Dogs! Dogs! Dogs!" is treated as a single item. We only use a set of frequent tags and group names (about 6000) in the auxiliary dataset. The other tags and group names are not counted. The text feature is a normalized histogram of tag and group name counts.

2.4. Classifier

The purpose of this paper is to show that a text feature, computed from the auxiliary dataset, is in fact a powerful and general descriptor. Various classifiers could be applied to such a feature. We have chosen to use an SVM classifier with a chi-square kernel for the text features. The same classifier is used for the visual features.

2.5. Fusion

We now have two types of features: the standard visual features and the text features. We do not believe there is likely to be much interaction, in the sense that one feature can tell when the other is unreliable. Therefore, we build two separate classifiers, one for the text features and the other one for the visual features. A third classifier is then trained to combine the confidence values of the two initial classifiers into a final prediction. This final classifier uses logistic regression and is trained on a validation set.

3. Experiment

We perform image classification experiments on two datasets: PASCAL VOC 2006 and PASCAL VOC 2007. The PASCAL 2006 dataset has 10 object categories while the 2007 dataset has 20 categories. The 2007 dataset is more difficult because there is much more variation with the object appearance. To ensure that there were no PASCAL test images in our auxiliary internet dataset, we removed all images from the auxiliary set that had a small image distance (within a threshold) to any image in the test set.

	bicycle	bus	car	cat	cow	dog	horse	motorbike	person	sheep
Gist(KNN)	0.795	0.875	0.885	0.736	0.820	0.674	0.734	0.822	0.605	0.868
Gist(V)	0.825	0.951	0.940	0.861	0.876	0.773	0.845	0.862	0.762	0.914
Gist(T)	0.818	0.915	0.932	0.812	0.843	0.744	0.820	0.878	0.733	0.875
Gist(V+T)	**0.837**	**0.955**	**0.941**	**0.869**	**0.880**	**0.790**	**0.858**	**0.886**	**0.769**	**0.917**
Gra(KNN)	0.734	0.837	0.902	0.743	0.808	0.666	0.743	0.786	0.625	0.799
Grad(V)	0.826	0.933	0.944	0.861	0.842	0.746	0.825	0.863	0.743	0.870
Grad(T)	0.810	0.931	0.935	0.806	0.830	0.725	0.776	0.817	0.722	0.855
Grad(V+T)	**0.834**	**0.941**	**0.947**	**0.864**	**0.850**	**0.766**	**0.831**	**0.878**	**0.756**	**0.877**
SIFT(KNN)	0.735	0.816	0.596	0.684	0.659	0.704	0.561	0.709	0.616	0.732
SIFT(V)	0.886	0.952	0.936	0.857	0.873	0.809	0.799	0.889	0.768	0.874
SIFT(T)	0.837	0.905	0.903	0.827	0.823	0.759	0.742	0.818	0.733	0.826
SIFT(V+T)	**0.889**	**0.953**	**0.937**	**0.861**	**0.877**	**0.812**	**0.805**	**0.896**	**0.776**	**0.897**
Color(KNN)	0.575	0.777	0.686	0.703	0.770	0.626	0.601	0.752	0.574	0.793
Color(V)	0.702	0.840	**0.843**	0.754	0.826	0.721	0.727	**0.864**	**0.703**	0.828
Color(T)	0.666	0.809	0.784	0.740	0.791	0.676	0.691	0.777	0.668	0.834
Color(V+T)	**0.715**	**0.853**	0.835	**0.782**	**0.850**	**0.726**	**0.754**	0.861	0.690	**0.865**
Unified(KNN)	0.794	0.883	0.841	0.794	0.850	0.720	0.695	0.852	0.630	0.866
Unified(V)	0.851	0.948	0.936	**0.885**	0.912	**0.822**	0.883	0.919	**0.800**	0.910
Unified(T)	0.873	0.924	0.933	0.826	0.877	0.788	0.826	0.901	0.785	0.873
Unified(V+T)	**0.901**	**0.959**	**0.944**	**0.885**	**0.922**	0.817	**0.890**	**0.931**	0.773	**0.923**
Combination(V)	0.891	**0.966**	0.953	0.902	0.918	0.823	**0.892**	0.933	0.816	0.917
Combination(T)	0.908	0.965	0.957	0.899	0.916	0.821	0.874	0.929	0.788	0.926
Combination(V+T)	**0.910**	0.965	**0.959**	**0.908**	**0.919**	**0.827**	0.887	**0.938**	**0.824**	**0.930**

Table 1. The AUC values with different settings on PASCAL 2006 for each object category. Take the Gist feature as an example: "Gist(KNN)" denotes the result with a KNN classifier using the Gist feature; "Gist(V)" denotes the result with the visual SVM classifier; "Gist(T)" denotes the result with the text SVM classifier; "Gist(T+V)" denotes the result by fusing the outputs of the text and visual SVM classifiers. Our text classifier outperforms the KNN classifier. The performance of the text features depends on the strength of the visual features. "Unified(T)" usually works best among all the text classifiers; and "Color(T)" usually works worst. We can get better performance in almost all of the categories by combining the text and visual classifiers. The results by combining all the text classifiers, all the visual classifier and all the text and visual classifiers, are indicated by "Combination(T)", "Combination(V)" and "Combination(T+V)" respectively.

According to the standard evaluation measure, the performance is quantitatively measured by AUC (area under the ROC curve) value on the 2006 dataset; and measured by AP (average precision) value on the 2007 dataset. When evaluating the methods, we are interested in the following phenomenons: (1)the performance of text features which are built with different visual features; (2) the effects of combining text and visual classifiers; (3) the effects of varying number of training images; (4)the performance of the text features built with varying number of internet images; (5)the effects of the category names.

3.1. Results: text features built with different types of visual features

We could use different types of visual features to retrieve the nearest neighbor images to build the text features. We use 150 nearest neighbor images in all the experiments. The performance on 2006 and 2007 for each object category is listed in Table 1 and Table 2 respectively. We use a KNN classifier as a baseline in Table 1 for the 2006 dataset. Each internet image is considered to be a positive example of the object categories whose names appear in the associated text. Then a test image can be simply classified by the KNN classifier. Our text classifier significantly outperforms KNN for each individual feature.

The performance of the text features is affected by the strength of the visual features. The better KNN performs, the better the text features are. This is because good visual features can find good nearest neighbor images to build good text features. So the text features built by the unified visual features usually work best; and the text features built by the color features usually work worst on both of the datasets.

3.2. Results: combining text and visual classifiers

Text features don't outperform visual features as shown in Table 1 and Table 2. But text features are quite different from visual features, so they can correct each other, and the

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
Gist(V)	0.575	0.253	0.324	0.512	0.122	0.330	0.561	0.269	0.380	0.121
Gist(T)	0.520	0.207	0.296	0.509	0.089	0.335	0.509	0.227	0.302	0.178
Gist(V+T)	**0.580**	**0.272**	**0.362**	**0.548**	**0.189**	**0.392**	**0.578**	**0.295**	**0.383**	**0.203**
Grad(V)	0.571	0.230	0.238	0.403	0.116	0.333	0.551	0.308	0.397	0.184
Grad(T)	0.548	0.208	0.217	0.352	0.074	0.365	0.554	0.243	0.325	0.169
Grad(V+T)	**0.604**	**0.272**	**0.276**	**0.437**	**0.140**	**0.404**	**0.609**	**0.328**	**0.414**	**0.195**
SIFT(V)	0.510	0.297	0.249	0.412	0.122	**0.243**	0.416	0.330	0.324	0.212
SIFT(T)	0.288	0.254	0.237	0.367	0.104	0.184	0.309	0.320	0.264	0.209
SIFT(T+V)	**0.517**	**0.348**	**0.310**	**0.437**	**0.192**	0.241	**0.431**	**0.365**	**0.336**	**0.240**
Color(V)	0.367	0.124	0.220	0.215	0.112	0.085	0.323	0.134	0.242	0.075
Color(T)	0.400	0.084	0.215	0.215	0.078	0.107	0.332	0.112	0.154	0.098
Color(T+V)	**0.431**	**0.179**	**0.239**	**0.261**	**0.179**	**0.129**	**0.369**	**0.140**	**0.260**	**0.117**
Unified(V)	0.647	0.399	0.450	0.540	0.207	0.425	0.577	0.388	0.439	0.273
Unified(T)	0.580	0.349	0.407	0.545	0.120	0.329	0.565	0.366	0.352	0.170
Unified(V+T)	**0.666**	**0.445**	**0.512**	**0.580**	**0.232**	**0.450**	**0.619**	**0.438**	**0.459**	**0.295**
Combination(V)	0.675	0.407	0.423	0.581	0.239	0.432	0.646	0.421	0.449	0.279
Combination(T)	0.640	0.418	0.459	0.571	0.204	0.436	0.631	0.419	0.402	0.280
Combination(V+T)	**0.684**	**0.481**	**0.497**	**0.593**	**0.253**	**0.481**	**0.673**	**0.476**	**0.469**	**0.327**
	table	dog	horse	motorbike	person	plant	sheep	sofa	train	monitor
Gist(V)	0.289	0.270	**0.652**	0.364	0.679	**0.173**	0.167	0.281	0.541	0.316
Gist(T)	0.144	0.237	0.446	0.331	0.623	0.080	0.141	0.139	0.512	0.228
Gist(V+T)	**0.290**	**0.281**	**0.652**	**0.405**	**0.704**	0.130	**0.170**	**0.284**	**0.586**	**0.335**
Grad(V)	**0.356**	0.248	0.539	0.299	0.662	**0.118**	**0.131**	0.259	0.467	0.286
Grad(T)	0.205	0.179	0.432	0.251	0.601	0.081	0.080	0.171	0.409	0.207
Grad(V+T)	0.316	**0.253**	**0.575**	**0.336**	**0.670**	0.111	0.125	**0.263**	**0.485**	**0.332**
SIFT(V)	0.163	0.284	0.417	**0.243**	0.662	0.114	0.164	**0.196**	0.318	**0.227**
SIFT(T)	0.201	0.201	0.373	0.165	0.635	0.159	0.163	0.097	0.263	0.141
SIFT(T+V)	**0.239**	**0.321**	**0.474**	0.228	**0.687**	**0.182**	**0.255**	0.191	**0.339**	0.216
Color(V)	0.128	0.186	0.442	0.182	0.594	0.146	0.162	0.083	0.243	**0.122**
Color(T)	0.117	0.148	0.451	0.106	0.580	0.085	0.134	0.099	0.118	0.092
Color(T+V)	**0.195**	**0.220**	**0.513**	**0.192**	**0.615**	**0.148**	**0.163**	**0.121**	**0.255**	0.100
Unified(V)	0.373	0.343	0.657	0.489	0.749	0.330	0.324	0.323	0.619	0.322
Unified(T)	0.271	0.271	0.556	0.414	0.691	0.179	0.260	0.202	0.513	0.259
Unified(V+T)	**0.413**	**0.375**	**0.681**	**0.526**	**0.782**	**0.355**	**0.344**	**0.346**	**0.661**	**0.379**
Combination(V)	0.388	0.354	0.704	0.447	0.774	0.245	0.267	0.345	0.619	0.379
Combination(T)	0.336	0.335	0.648	0.484	0.738	0.233	0.305	0.252	0.612	0.295
Combination(V+T)	**0.442**	**0.392**	**0.715**	**0.528**	**0.786**	**0.272**	**0.322**	**0.350**	**0.665**	**0.402**

Table 2. The AP values with different settings on PASCAL 2007 for each object category. Take the Gist feature as an example: "Gist(V)" denotes the result with the visual classifier; "Gist(T)" denotes the result with the text classifier; "Gist(T+V)" denotes the result by combining the text and visual classifiers. The performance of the text features depends on the strength of the visual features. "Unified(T)" usually works best among all the text classifiers; and "Color(T)" usually works worst. We get better performance consistently by combining the text and visual classifiers. The results by combining all the text classifiers, all the visual classifier and all the text and visual classifiers, are indicated by "Combination(T)", "Combination(V)" and "Combination(T+V)" respectively.

combination should result in improvement.

Table 1 and Table 2 show that the combination consistently outperforms separate classifiers (the best performance in each panel is indicated in bold, look for the bold horizontal line). In Fig. 3, we show several examples which are misclassified by the visual classifier, but correctly classified by the text classifier on PASCAL 2006. The objects vary widely. In the first image, the cat is in a sleeping pose, which is unusual in the PASCAL training set. So the visual classifier gets it wrong. However, we may find many such images in the auxiliary dataset (there are several sleeping cat images in the 25 nearest neighbors). Now the text cue can make a correct prediction. The text vector also shows that the group name is an important cue. There are several

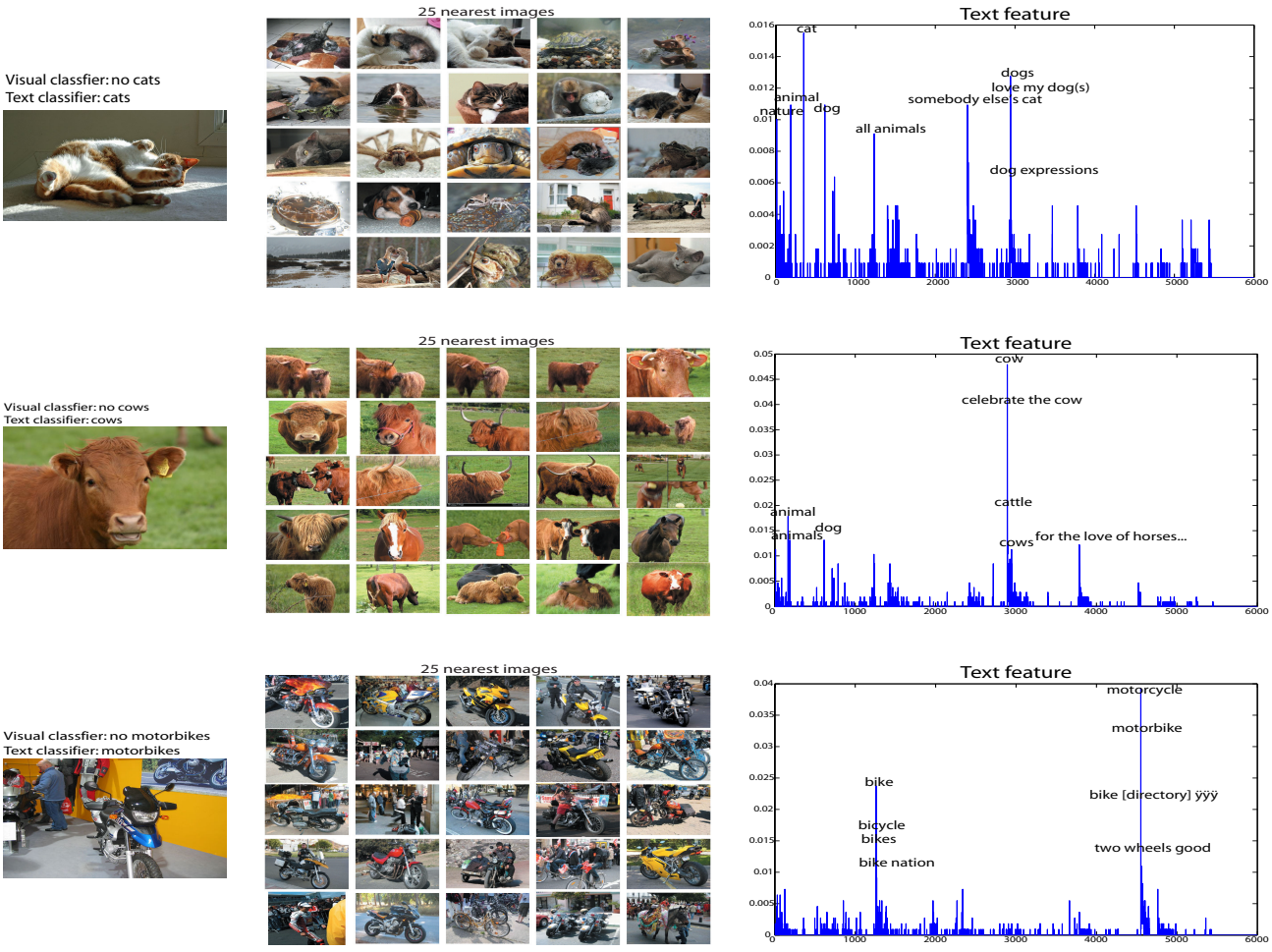

Figure 3. The left column shows the PASCAL 2006 images whose category labels cannot be predicted by the visual classifier, but can be predicted by the text classifier; The center column shows the 25 nearest neighbor images retrieved from the internet dataset; the right column shows the built text feature vectors. In the first image, the cat is in a sleeping pose, which is unusual in the PASCAL training set. So the visual classifier gets it wrong. Some sleeping cat images are retrieved from the auxiliary dataset. Then the text features make a correct prediction.

peaky groups such as "somebody else's cat", "all animals" and so on. In Fig. 4, we also show images which are misclassified by the text classifier but correctly classified by the visual classifier. This happens when we fail to find good nearest neighbor images.

At the bottom of Table 1 and Table 2, we show the performance obtained by combining the different classifiers, which is achieved by training a logistic regression classifier on the validation dataset using the confidence values returned by the individual classifiers as features. Combining all the visual classifiers works better than combining only visual classifiers or text classifiers.

3.3. Results: varying number of training images

In Fig. 5, we show the performance with different number of training images on PASCAL 2006. We randomly select $1/40$, $1/30$, $1/20$, $1/10$, $1/5$, $1/2$ of the positive and negative images respectively in the training data for each category to do the experiments. For comparison, we also show the results with all the training images. The performance is shown by the average AUC values over all the categories. We do experiments by the "Gist" and the "Combination" of multiple classifiers. We observe that the text features outperform the visual features when there are only a small set of training images available. There is always improvement by combining the two type of features, but the gain is not significant when the two classifiers are not comparable.

3.4. Result:varying number of auxiliary images

We also test the performance of the text features built with varying number of internet images in Table 3 on PAS-

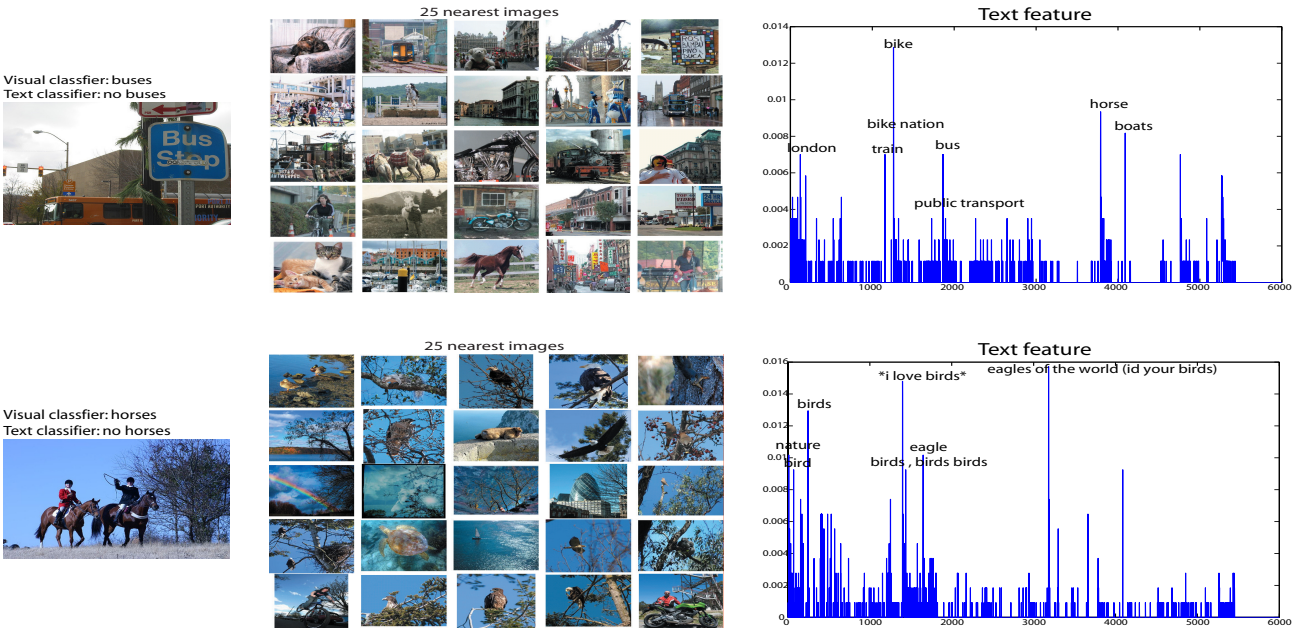

Figure 4. The left column shows the PASCAL images whose category labels cannot be predicted by the text classifier, but can be predicted by the visual classifier; The center column shows the 25 nearest neighbor images retrieved from the internet dataset; the right column shows the built text features of the PASCAL images. The text features do not work here mainly because we fail to find good nearest neighbor images.

	200,000	600,000	1,000,000
Gist(T)	0.7116	0.8297	0.8370
SIFT(T)	0.6975	0.8104	0.8173
Grad(T)	0.7016	0.8093	0.8207
Color(T)	0.6496	0.7370	0.7436
Unified(T)	0.7413	0.8583	0.8606

Table 3. The performance of the text features built with different numbers internet images on PASCAL 2006. We randomly select 200,000, 600,000 images from the collection to construct the text features. The result is based on the average AUC values over the 10 object categories.

CAL 2006. We randomly select 200,000, 600,000 images from the collection to build the text features. The result is based on the average AUC values over the 10 object categories.

Increasing the image number from 200,000 to 600,000 leads to a big improvement, but further increasing to 1 million results in a negligible improvement.

This means that merely increasing the size of the auxiliary dataset may not have much impact. Instead, one should create an auxiliary dataset covering more meaningful images and improve the technique to find good nearest neighbor images.

	bicycle	bus	car	cat	cow
W	**0.818**	0.915	0.932	**0.812**	0.843
WO	0.817	**0.917**	0.932	0.811	**0.848**
	dog	horse	motorbike	person	sheep
W	**0.744**	**0.820**	**0.878**	0.733	**0.875**
WO	0.738	0.816	0.876	**0.734**	**0.875**

Table 4. When we exclude category names and their plural inflections from the text features, there is little effect on the performance. We show results for Pascal 2006: W - with category names; WO - without.

3.5. Result: excluding the category names

Our text features might be powerful only because our images are tagged with category labels. To test this, we exclude category names and their plural inflections from the text features. This means that, for example, the words "cat" and "cats" would not appear in the features. The effect on performance is extremely small (Table 4). This suggests that text associated with images is rich in secondary cues (perhaps "mice" or "catnip" appear strongly with cats). In future work, we will investigate directly applying semantic measures of similarity to our features.

4. Conclusion

Text produced by matching an image to a large auxiliary collection of images which have noisy annotations is a sur-

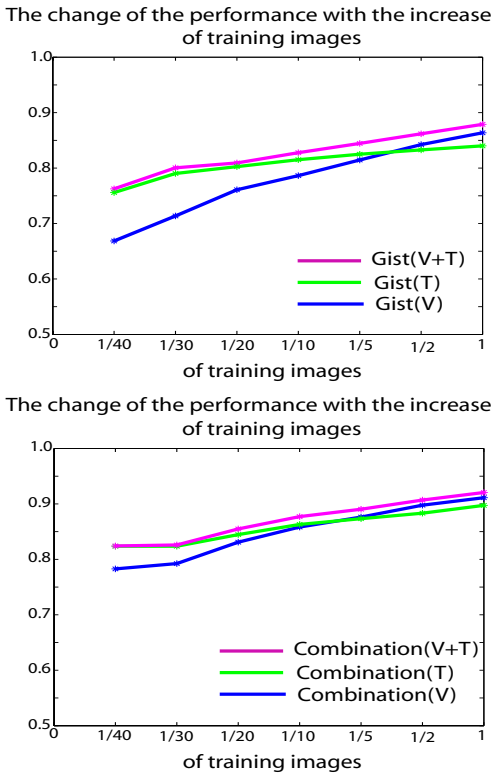

Figure 5. The performance with different numbers training images on PASCAL 2006. We randomly select $1/40$, $1/30$, $1/20$, $1/10$, $1/5$, $1/2$ of the positive and negative images respectively in the training data for each category. The performance is shown by the average AUC values over all the categories. We do experiments by the "Gist" and the "Combination" of multiple classifiers. The text features outperform the visual features when there are only a small set of training images available. There is always improvement by combining the two type of features, but the gain is not significant when the two classifiers are not comparable.

prisingly powerful feature. One caution is necessary. It is unwise to expect that text produced by matching with a relatively weak visual feature will enhance a different, more powerful, visual feature. For example, we have been able to obtain the SVM score produced by the overall winner for Pascal 2007 test images (INRIAGenetic of [3]). We fuse this SVM with a classifier applied to a text feature. We obtain the text features by matching using our unified visual feature (which is not as powerful as the Pascal winner), and we observe no improvement. The situation is analogous to fusing the unified visual feature with text produced by a color matcher (one observes no improvement).

However, our feature is somewhat decorrelated from direct visual features. It can be used to enhance any visual feature capable of producing matches, and doing so has consistently improved recognition performance in our experiments with large standard datasets.

5. Acknowledgement

We would like to thank Julia Hockenmaier and Peter Young for their comments and suggestions. This work was supported in part by the National Science Foundation under IIS - 0534837 and 0803603 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research.

References

[1] N. Ben-Haim, B. Babenko, and S. Belongie. Improving Web-based Image Search via Content Based Clustering. In *Conference on Computer Vision and Pattern Recognition Workshop CVPRW*, pages 106–111, 2006. 2

[2] TL Berg, AC Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E. Learned-Miller, and DA Forsyth. Names and faces in the news. In *CVPR, 2004*, volume 2. 2

[3] M. Everingham, L. Van Gool, CKI Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007. 1, 8

[4] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, 2004. 2

[5] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005. 2

[6] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. 2007. 3

[7] J. Hays and A.A. Efros. Scene completion using millions of photographs. *SIGGRAPH*, 2007. 2

[8] J. Hays and A.A. Efros. IM2GPS: estimating geographic information from a single image. In *CVPR 2008*, pages 1–8, 2008. 2

[9] L.J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning. 2007. 2

[10] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2

[11] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV.*, 42, 2001. 3

[12] A. Quattoni, M Collins, and T. Darrell. Learning visual representations using images with captions. In *CVPR*, June 2007. 2

[13] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007. 2

[14] A. Torralba, R. Fergus, and W. T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 2007. 2

[15] G. Wang and D. Forsyth. Object image retrieval by exploiting online knowledge resources. In *CVPR*, pages 1–8, 2008. 2

[16] K. Wnuk and S. Soatto. Filtering Internet image search results towards keyword based category recognition. In *CVPR*, 2008. 2