The Thesis committee for Sung Ju Hwang
Certifies that this is the approved version of the following thesis:

Reading Between The Lines: Object Localization Using Implicit Cues from Image Tags

APPROVED BY

SUPERVISING COMMITTEE:

Kristen Grauman, Supervisor

Matthew Lease, Supervisor

Reading Between The Lines: Object Localization Using Implicit Cues from Image Tags

by

Sung Ju Hwang, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Arts

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2010

Acknowledgments

It is my privilege to have Professor Kristen Grauman as my advisor for my graduate work. Without her generous support, guidance, and patience, this work would not have been possible.

I would also like to thank Professor Matthew Lease for his support and helpful discussions.

Last but not least, I thank my family and friends. This thesis is a product of their unconditional love, support, and encouragement throughout the years. Words alone cannot convey my appreciation and love for all of you.

Reading Between The Lines: Object Localization Using Implicit Cues from Image Tags

Sung Ju Hwang, M.A.
The University of Texas at Austin, 2010

Supervisors: Kristen Grauman
Matthew Lease

Current uses of tagged images typically exploit only the most explicit information: the link between the nouns named and the objects present somewhere in the image. We propose to leverage "unspoken" cues that rest within an ordered list of image tags so as to improve object localization. We define three novel implicit features from an image's tags—the relative prominence of each object as signified by its order of mention, the scale constraints implied by unnamed objects, and the loose spatial links hinted by the proximity of names on the list. By learning a conditional density over the localization parameters (position and scale) given these cues, we show how to improve both accuracy and efficiency when detecting the tagged objects. We validate our approach with 25 object categories from the PASCAL VOC and LabelMe datasets, and demonstrate its effectiveness relative to both traditional sliding windows as well as a visual context baseline.

Table of Contents

Chapter 1

Introduction

Photo sharing web services, captioned news photo archives, and social networking websites all offer an abundance of images that have been manually annotated with keywords ("tags"). Often tags mark physical things shown in the photo (such as names of objects, locations, landmarks, or people present), which allows users to retrieve relevant photos within massive collections using a simple text-based search. Today millions of people provide such tags, and many more benefit from them when organizing their photos or searching for images. Computer vision researchers in particular regularly exploit tagged images, harvesting datasets that can then be pruned or further annotated to train and test object recognition systems [6, 11].

Those image tags that are nouns serve naturally as "weak supervision" for learning object categories: they flag the presence of an object within the image, although which pixels actually correspond to the object remains ambiguous. A number of techniques have been developed to learn from such loosely labeled data, typically by designing learners that can cope with high label noise [13, 21, 25, 30], or can discover the correspondence between multiple words and the image's regions [2, 3, 18].

Figure 1.1: **Main idea:** the list of tags on an image may give useful information beyond just which objects are present. The tag lists on these two images indicate that each contains a mug. However, they also suggest likely differences between the mug occurrences—even before we see the pixels. For example, the relative order of the words may indicate prominence in location and scale (mug is named first on the left tag list, and is central in that image; mug is named later on the right tag list, and is less central in that image), while the absence of other words may hint at the total scene composition and scale (no significantly larger objects are named in the left image, and the mug is relatively large; larger furniture is named on the right, and the mug is relatively small).

In this work we introduce the idea of "reading between the lines" of image tags. We propose to look beyond image tags as merely offering names for objects, and consider what *implicit* cues a human tagger additionally gives (perhaps unknowingly) based on the way he or she provides the tags. The intuition is that a number of factors besides object presence influence how a person looks at an image and generates a list of tags—for example, the semantic importance of the objects or their centrality in the image can affect which is mentioned first; the spatial proximity of objects can affect their sequence in the tag list; low-level attentional cues can steer gaze patterns. While the existence of such behavior effects has been studied to some extent in the user interface and psychology communities [1, 9, 10], object detection methods have

yet to capitalize on them.

Our main idea is to learn a model to predict how likely a given object location and scale are given an image's ordered tag list. To do this, we define three new implicit features computed directly from the image's tags—the relative prominence of each object as signified by its *order* of mention, the scale cues implied by *unnamed objects*, and the loose spatial links hinted by the *proximity of names* on the list (see Figure 1.1). Having learned how these unspoken cues map to object localization, we can prime the object detectors to search the most likely places first in a novel tagged image, or prefer detections that are plausible according to both the subregion's appearance as well as its agreement with the tag-based predictor. In this way, we intend to improve both the accuracy and efficiency of object localization within weakly human-labeled images.

We demonstrate the effectiveness of our approach on a wide variety of categories in real images tagged by anonymous annotators. Our results show good gains relative to both a traditional sliding window method as well as an alternative location priming baseline that uses visual cues.

Chapter 2

Related Work

Sliding window object detectors test subregions at multiple scales and locations to find a target object, classifying each image window as to whether it contains the category of interest. Due to the expense of classifying each possible window, some techniques aim to reduce the number of windows scanned, either by priming the detector based on global context [22, 29], or directing the search with cascades [31] or branch-and-bound techniques [20]. We also intend to prioritize scanning of those regions that are most likely to contain the object of interest, however we do so based on priming effects learned from implicit image tag features.

Visual features can provide a form of scene-level context that improves the detection of foreground objects [8, 16, 17, 22, 29], and recent work shows how to improve object detection using learned inter-object co-occurrence or spatial relationships [7, 14, 15]. Our approach also seeks cues about total scene composition and layout; however unlike previous work, our information is based solely on implicit associations learned from seeing many tagged images with ground truth bounding boxes, not from visual cues. Aside from the potential advantage of bypassing image feature computation, our tag-based

features can also capture correlations with localization parameters not always evident in the visual scene context, as we will discuss below.

Researchers frequently use keyword-based search as a first step to collect candidate images for datasets [6, 11]. Given the expense of hand-annotating images, a number of techniques have been designed to learn visual category models directly from Web images with no human intervention [13, 21, 25, 30]. To exploit images associated with multiple words or captions, methods have been developed to automatically recover correspondences between the words and image regions [2, 3, 18].

Also relevant to this project is work studying how people look at images, what affects their attention, and what factors determine the words they will generate if asked to tag a photo [1, 9, 10, 27, 28, 33]. Saliency operators use bottom-up visual cues to find interesting image points, e.g. [19]. Such low-level cues have been shown to coincide often with those objects that people find interesting and therefore choose to label [10], though the top-down saliency of recognized objects also plays a role [9]. The authors of [27] explore the notion of "importance" in images, as reflected by what objects people tend to name first. They design a statistical model for the naming process, and demonstrate a regressor that takes hand-segmented images and predicts a list of the most important keywords based on the visual cues. We are essentially tackling the inverse problem—given the human-generated keywords, we want to localize (segment) the objects. For our proposed method to work, it must be the case that people often agree on what objects to name in an image, and

in what order; the success of the ESP Game [32] is encouraging evidence for this premise.

No previous work considers exploiting the information implied by how a person assigns words to an image for the sake of actually strengthening object detection, as we propose here. Perhaps most related to our theme in spirit is the Bayesian expert system for medical diagnosis designed in [23] that captures biases in how patients report symptoms. The authors note that faster diagnoses can be made if inference relies on both what is and is not reported to a doctor, and in what order. They therefore adjust the model to reflect the human knowledge that people prefer to report present symptoms over absent ones, and more severe problems before less severe ones. We see some neat (rough) analogies between this doctor-patient interchange and our tagger-image interchange, since both can benefit from relative importance and noticeability of symptoms/objects. However, in our approach these patterns are learned from data, and of course the image annotation problem has unique challenges.

Chapter 3

Approach

We aim to exploit implicit tag-based features to strengthen object detectors. First, we collect tagged images online, and encode features for each image's tag list. Then we model the conditional probability distribution for each object category's image position and scale given the cues implied by the tags. Separately, we train appearance-based object detectors using state-of-the-art techniques [5, 12]. Given a novel image tagged by an unknown user, our method can perform in one of two modes: either we (1) prioritize search windows within the image based on the learned distribution, thereby speeding up the search relative to the usual sliding-window detector, or else we (2) combine the models to perform more accurate object localization based on both the tags and the pixels.

Below we first define our implicit tag features (Section 3.1), and then describe how we represent the conditional densities (Section 3.2); finally, we describe how we integrate them into the detection process (Section 3.3).

3.1 Implicit Tag Feature Definitions

We propose three implicit features that can be extracted from an image's tags. We specify each descriptor and what it is intended to capture in turn below.

Word Presence and Absence. This feature is a traditional bag-of-words representation, extracted from a single image's list of tags. An image is mapped to a histogram $\mathbf{W} = [w_1, \ldots, w_N]$, where w_i denotes the number of times that tag-word i occurs in that image's associated list of keywords, for a vocabulary of N total possible words. We assume that synonyms and misspellings are resolved to map to the same token (e.g., `car` and `auto` map to a single word). For most tag lists, this vector will consist of only binary entries saying whether each tag has been named or not.

While this feature certainly specifies what *was* said about the image—which words were named—it also indirectly implies attributes for the named objects based on what was *not* said. The known presence of multiple objects serves as a sketch of the total scene composition, which constrains the type of layout or scales those objects may have. Further, those objects that are not named suggest what the total scale/scope of the scene may be, given the tendency to name prominent or large objects in favor of smaller ones [27]. For example, it is more likely that when one tags "`flower, spider`", the flower is prominent in the field of view; whereas if one tags "`flower, garden, wheelbarrow`", it is likely the flower region is smaller. Thus we get information from what is not reported by the human labeler that may aid in localization.

Note that the tag list need not completely cover all objects present for such correlations to be discovered. Additionally, the vocabulary need not contain only nouns. The presence or absence of certain adjectives and verbs could also convey composition and relative scale; however, we have not yet tested this, primarily since our data happens to consist of nouns.

In a sense, the word-count feature explicitly states that which global image descriptors designed to prime object detectors hope to capture indirectly [22, 29]. Both say something about total scene content. However, tag-based features can actually reveal correlations with object placement or scale not captured by a visual scene feature: people may provide similar keywords for images where the localization parameters are common, yet the surrounding visual context varies (for example, consider an image like the left image in Figure 1.1, and a second image where the mug is at a similar scale, but set amidst other randomly placed desk clutter). At the same time, the scene structure revealed by global visual cues can offer better location cues in cases where the tags are less reliable. Thus the two channels can be complementary; we demonstrate this in experiments.

Tag Rank. Our second feature captures the prominence of the named object as implied by its order of mention in the list. The idea is that people do not suggest tags in an arbitrary order; rather, multiple factors bias us towards naming certain things before others, including relative scales and centrality within the image, object significance, and attentional cues [9, 10, 28, 33].

To encode the named objects' order and relative ranking simultane-

ously, we map the tag list to the vector $\mathbf{R} = [r_1, \ldots, r_N]$, where r_i denotes the percentile of the rank for tag i in the current image, relative to all previous ranks observed in the training data for that word (note that i indexes the vocabulary, not the tag list). The higher the value, the more this word tops the list relative to where it typically occurs in any other tag list; if the tag is not present, the percentile is 0. Some objects have context-independent "noticeability"—such as `baby` or `fire truck`—and are often named first regardless of their scale or position in that particular image. Thus, by using the tag-specific percentile rather than raw rank on the list, we attempt to account for semantic biases that occur in tagging.

For this cue, we expect to benefit most from the central fixation bias [28] and the fact that something named sooner than usual may be atypically prominent in this view. Put simply, we expect bigger or more centrally located objects to often be named first, which should help the windowed detector home in on proper scales and positions.

Mutual Tag Proximity. When scanning an image, people generally do not systematically move their eyes from one corner of the image to another. In fact, their sequence of attention to multiple objects is influenced in part by the objects' spatial proximity [10]. Thus, an image tagger may name a prominent object first, and then, as her eyes travel, note some other objects nearby.

Our third implicit cue therefore attempts to capture the rough layout and proximity between objects based on the sequence in which tags are

given. We map the tag list to a vector encoding the mutual nearness of each pair of words: $\mathbf{P} = [\frac{1}{p_{1,2}}, \frac{1}{p_{1,3}}, \ldots, \frac{1}{p_{1,N}}, \ldots, \frac{1}{p_{2,3}}, \ldots, \frac{1}{p_{N-1,N}}]$, where $p_{i,j}$ denotes the (signed) rank difference between tag-words i and j for the given image. The entry is 0 when the pair is not present. (Dimensionality: $\mathbf{P} \in \mathbb{Z}^{\frac{N^2}{2}}$.) Whereas the tag rank feature $\mathbf{R}$ defined above captures the individual (word-normalized) orders of mention, this one records nearness in the list and relative order between words.

3.2 Modeling the Localization Distributions

Having defined the features, next we describe how to relate them to the object detection task. Localization entails three parameters—the scale of the window, and its center position in image coordinates. Denote this as $X = (s, x, y)$. We want to model the conditional probability density $P_o(X|T)$, where O denotes the target object category, and T denotes one of the tag-based features defined above: $T = \mathbf{W}$, $\mathbf{R}$, or $\mathbf{P}$ (or some combination thereof). That is, we want to estimate the probability a given window contains the object of interest, conditioned only the image's tags.

We model this as a mixture of Gaussians, $P_o(X|T) = \sum_{i=1}^{m} \alpha_i \, \mathcal{N}(X; \mu_i, \Sigma_i)$, since we expect most categories to exhibit multiple modes of location and scale combinations. We compute the mixture model parameters α_i, μ_i, and Σ_i using a Mixture Density Network (MDN) [4] trained from a collection of tagged images with bounding box ground truth for the target object. The MDN is a neural network trained with instances of tag-list features $\{T_1, \ldots, T_M\}$ and

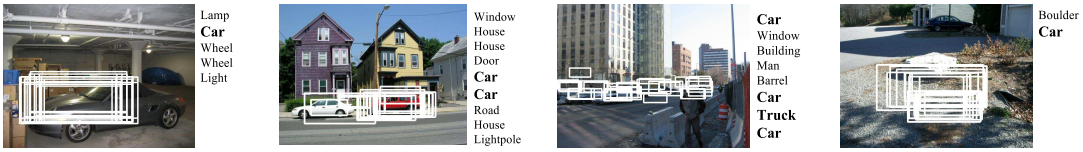


Figure 3.1: The top 30 most likely places for a `car` in several tagged images, as computed by our method. These bounding boxes are sampled according to $P_o(X|T)$; the actual image appearance is *not* yet being used. Note how our predictions change depending on what the tag list implies. The bottom right example shows a failure case, where the absence of larger objects leads our method to overestimate the scale.

their associated target parameters $\{X_1, \ldots, X_M\}$ to output the mixture density model parameters that define $P_o(X|T)$. Given a novel tagged image that lacks bounding boxes, the MDN provides a mixture model representing the most likely locations for the target object. This allows us to prime a detector based only on what the tags suggest. Other models are of course possible; our choice is motivated by MDNs' efficiency, as well as their past successful use for primed detection in [22].

3.3 Modulating or Priming the Detector

Once we have the function $P_o(X|T)$, we can either combine its predictions with an object detector that computes $P_o(X|A)$ based on appearance cues A, or else use it to rank sub-windows and run the appearance-based detector on only the most probable locations ("priming"). The former has potential to improve accuracy, while the latter will improve speed.

We can integrate any existing window-based detector into our method; we experiment with two state-of-the-art methods: the HOG detector of [5],

which works well for rigid textured objects, and the part-based detector of [12], which can also accommodate deformable or articulated objects. Both detectors perform multi-scale windowed detection and return an SVM decision value $d(x, y, s)$ for the input window. We use a sigmoid function to map the score to a probability: $P_o(X = (x, y, s)|A) = \frac{1}{1+\exp(-d(x,y,s))}$, where the score d is computed at the window centered at (x, y) and with diagonal length s.

Modulating the detector: To balance the appearance- and tag-based predictions so as to improve detection *accuracy*, we treat the component conditional density estimates as scalar features and train a logistic regression classifier:

$$P_o(X|A, T) = \sigma\left(w^T \begin{bmatrix} 1 & P_o(X|A) & P_o(X|T) \end{bmatrix}\right). \quad (3.1)$$

Here $P_o(X|T)$ is as defined in the previous section; to use all our tag cues in combination, this breaks out into $P_o(X|T) = [P_o(X|\mathbf{W})\ P_o(X|\mathbf{R})\ P_o(X|\mathbf{P})]$, and a weight is learned for each component feature. Similarly, to optionally incorporate an external context cue, we expand the vector; in some experiments below we insert $P_o(X|G)$ for the Gist descriptor G to compare against the global scene visual context [29]. To learn the weights w, we use the detection scores for the true detections in the training set together with an equal number of randomly sampled windows pulled from the background.

Generally we expect this combination to eliminate false positives that may occur when using an appearance-based detector alone, particularly for objects whose texture is less distinctive. Similarly, we hope to correct false negatives, particularly when the target object occurs at low resolution or is

Dataset	LabelMe	PASCAL VOC 2007
Number of training images	3799	5011
Number of testing images	2553	4952
Number of classes	5	20
Number of keywords	209	399
Number of taggers	56	758
Average number of tags / image	23	5.5
x variance	0-98.8% (23.8%)	1.3-99.6% (23.5%)
y variance	0.5-90.6% (12.9%)	1.5-98.3% (17.9%)
s variance	0.9-77.1% (11.7%)	11.6-99.8% (25.2%)

Figure 3.2: Dataset statistics. Last three columns show the ranges of positions/scales present in the images, averaged per class, as a percentage of image size.

partially occluded. With a strong enough model for the tag-based cues, we will prefer only those detections that seem plausible according to both what is seen as well as what the human tagger has (implicitly) reported.

Priming the detector: To improve the detector's *efficiency*, we let the implied tag cues prime the search for the target object. Unlike typical detection tasks, we have tags on the test images, so we presume that the object is indeed present; what's left to estimate is the best set of localization parameters (x, y, s). Thus, instead of scanning the whole image, our method prioritizes the search windows according to $P_o(X|T)$, and stops searching with the appearance-based detector once a confident detection is found. (See Figure 3.1 for real examples of locations we'd search first.)

The idea of learning to constrain search for objects based on *visual* features has been explored previously [17, 22, 29], and while we exploit novel tag-based features, the technical machinery in our method draws inspiration

from that work. The important distinction, however, is that while the visual context challenge considers how much information can be taken from the image itself before running an object detector, our approach considers what can be predicted before looking at the image at all.

We envision two scenarios where our method can be applied. The first is the "weak supervision" scenario, where an image has been tagged intentionally to list objects that are present (perhaps by hire, e.g. [6, 26]). The second is the "unaware tagger" scenario: the method processes publicly available images from sites such as Flickr, where users have tagged images for their own purposes, but rarely draw a box around the specific instances. Training requires obtaining manual bounding boxes in either case. At test time, however, images are tagged but foreground pixels are not demarcated; from that minimal human input, our method helps to rapidly and accurately localize named objects.

Chapter 4

Results

We evaluate our method on two datasets: LabelMe [24] and the PASCAL VOC 2007 [11]. Both collections provide realistic snapshots taken by a variety of people and containing various scenes and combinations of objects; we use tags provided by (in total) hundreds of anonymous annotators who are entirely unaware of the experiments we are doing.

We report results for both tasks described in Section 3.3, and also include comparisons with a visual scene-based context model for reference.

Implementation details. We extract our proposed tag features as defined above for each image. Figure 3.2 gives the dataset statistics, including a summary of the ranges of the localization parameters (to show that they do vary significantly per target object). We use the LabelMe tools to resolve synonyms and purify labels. We fix the number of components in the mixture models to $m = 12$ and 8, on LabelMe and PASCAL, respectively, and use 10 hidden units for the MDNs (we leave all parameters the same for all categories, and did not attempt to tune them for better performance). We use Netlab code for the MDNs.

On LabelMe we use the HOG detector for the base appearance-based

detector [5], since the objects are generally amenable to the HOG descriptor; on PASCAL we use the part-based detector [12], since it has been shown to provide state-of-the-art results on that dataset. In both cases we use the authors' code[1], only modifying it to optionally search windows in an arbitrary order as specified by $P_o(X|T)$. We use the standard definition to evaluate detections: there is a "hit" if its area of overlap with the ground truth box normalized by their union exceeds 50%.

4.1 LabelMe Dataset

In LabelMe, annotators label images online with both keywords and object outlines, and the system maintains the order in which tags are added to each image. We downloaded images for the `person`, `car`, `screen`, `keyboard`, and `mug` categories—all of which show the object at a variety of scales and positions. We report the average results across five runs with random train/test splits.

Priming Object Search: Increasing Speed. First, we compare our detector's speed to the standard sliding window baseline, priming as described in Section 3.3. We measure performance by the portion of the windows that must be scanned to obtain any detection rate while allowing a reasonable number of false positives. Figure 4.1 (left) shows the results. Our method significantly reduces the number of windows that must be searched; e.g., for a detection rate of 0.6, our method considers only $\frac{1}{3}$ of those scanned by the sliding window. In fact, our method primes as well as the Gist visual scene

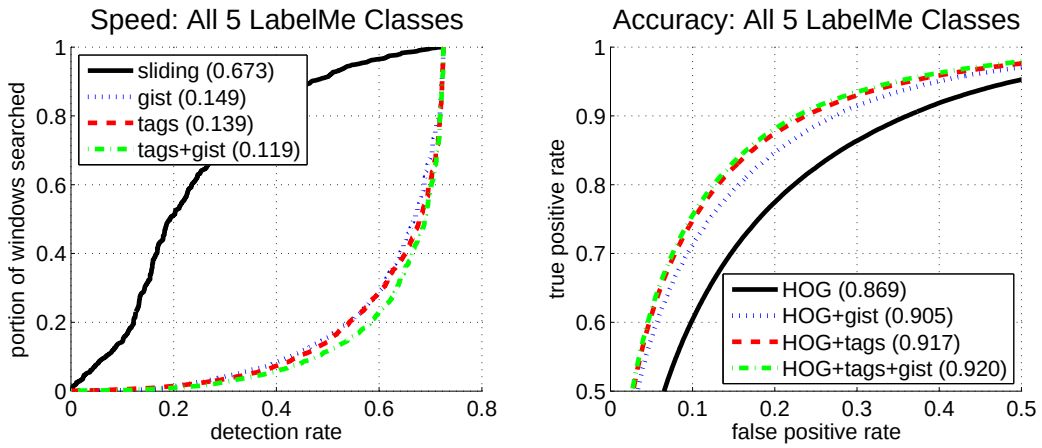


Figure 4.1: LabelMe results. **Left:** Percentage of windows searched as a function of detection rate, for all five categories. The numbers in the legend indicate the portion of the windows searched averaged over all detection rates. **Right:** Localization accuracy when the HOG detector is modulated with the proposed features. The numbers in the legend indicate the AUROC. Adding our implicit tag features (bold dotted lines) improves detection accuracy relative to the appearance-only HOG detector (dark solid line). Accuracy can be further boosted in some categories when the visual Gist context is also included (light dotted lines). Plot focuses on top left quadrant of ROC.

context, which is known to be strong for this dataset [22]; with tags and Gist combined, results are even a bit faster.

Modulating the Detector: Increasing Accuracy. Next, we evaluate how our learned features can improve localization accuracy on LabelMe. In this case, we search all windows, but modulate the scores of the HOG detector according to $P_o(X|T)$ (see Eqn. 3.1).

Figure 4.1 (right) compares the accuracy of the detector when run alone (HOG), the detector when augmented with our tag features (HOG+tags), and when further augmented with the Gist context (HOG+tags+gist). Overall,

class	HOG [5]	+Gist	+W	+R	+P	+Tags	+Tags+Gist
screen	0.866	0.897	0.906	0.903	0.898	0.913	0.916
keyboard	0.890	0.912	0.922	0.916	0.916	0.929	0.932
person	0.855	0.886	0.877	0.870	0.871	0.881	0.884
mug	0.863	0.874	0.892	0.881	0.882	0.898	0.897
carside	0.879	0.913	0.906	0.901	0.903	0.912	0.919

Figure 4.2: LabelMe localization accuracy (as measured by the AUROC) of the detectors modulated with each of the proposed feature types, compared with the raw detector and Gist.

our features make noticeable improvements in accuracy over the raw detector. This is exciting given the nature of the cues, which do not require even seeing the test image itself to compute. On three of the five categories our features are stronger than Gist, while for `person` and `car` (which occur in outdoor scenes) Gist is slightly better, again indicating that our tag-based features are actually quite competitive with a state-of-the-art representation for *visual* context for this data. We find our method's strength is often due to its accurate object scale prediction, which especially helps in indoor scenes in the LabelMe images.

Figure 4.2 summarizes the results when using each of the proposed features separately, showing that each one is indeed informative. For some objects we see further accuracy improvements (though small) when combining our features with Gist, suggesting the potential for using complementary visual and tag cues in concert. Figure 4.3 shows images with example detections.

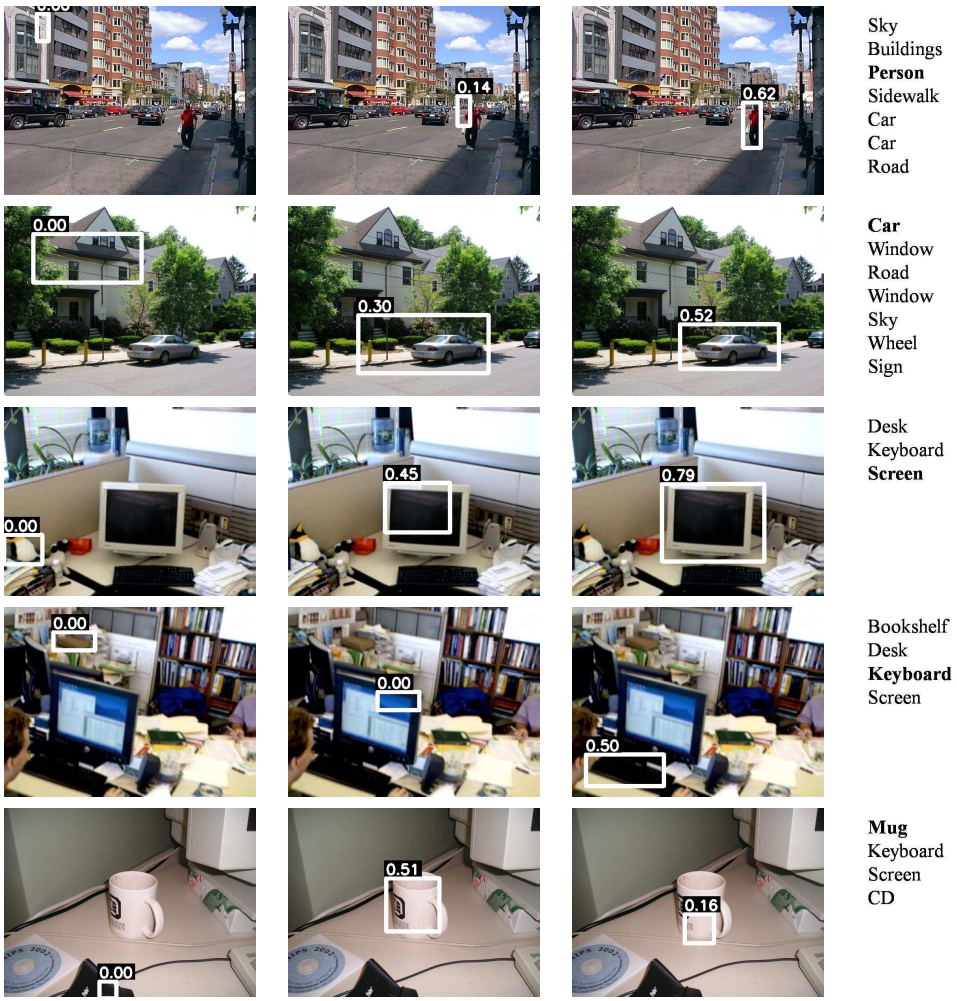


Figure 4.3: Example detections on LabelMe on five different target objects. Each image shows the best detection found; scores denote overlap ratio with ground truth. The raw appearance-only detector is confused by similar textures anywhere in the image (rows 1 and 2), whereas the detectors modulated according to the visual or tag-based context are more accurate. The Gist baseline usually gives good y estimates, while our method often provides a better scale estimate, particularly for indoor objects. When scene complexity (texture) is high throughout the image, Gist tends to be misled into predicting a too-small scale for the target object (row 4). Our approach can be misled on the scale prediction when the tags mention larger objects that are only partly visible (row 5).

4.2 Pascal VOC Dataset

The PASCAL VOC 2007 dataset is a benchmark for object detection systems. It is a challenging and quite realistic testbed for our method, as it consists of real user photos downloaded from Flickr, with a wide variety in composition. From previous work, we expect that the "context" in the VOC images is relatively weaker than LabelMe, meaning that there tends to be less viewpoint consistency or common inter-object statistics across examples [8].

Though the dataset creators gathered the images from Flickr, the original user tags were not kept. Thus, we collected tags using Mechanical Turk: we posted each image online with a nearby textbox, and the anonymous workers were instructed to name the objects or items in the image using nouns. In an attempt at minor quality control, we disabled the textbox until the image had been viewed by the tagger for 7 seconds, and required him/her to submit after 30 seconds had passed. We refined the resulting tags as above, using a spell checker and resolving synonyms. We use the `trainval` set to train the MDN and logistic regression parameters, and test on the standard `test` partition.

Priming Object Search: Increasing Speed. The procedure and evaluation measure is the same here as the previous section, except that we adopt the Latent SVM (LSVM) part-based windowed detector of [12] for this dataset, since it consistently has top results on the VOC. Figure 4.4 (left) shows the substantial speed improvements our method yields, over all 20 categories. The LSVM sliding window is faster here than the HOG's was on LabelMe,

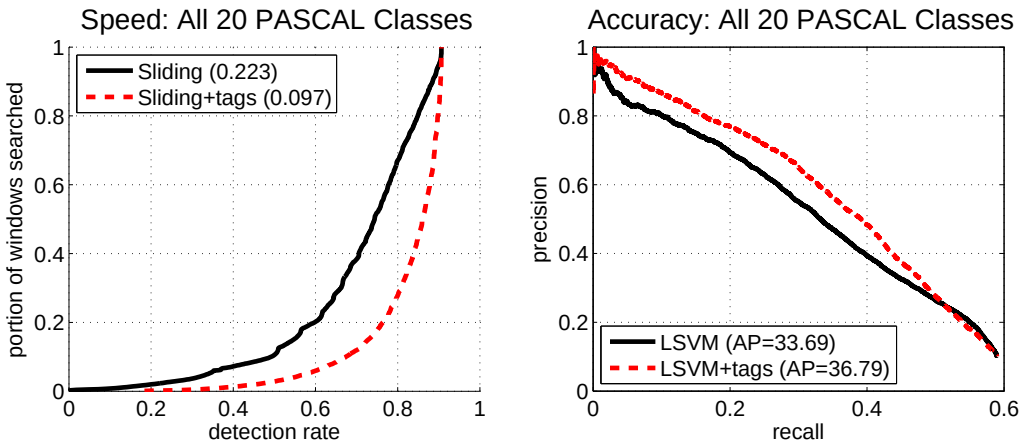


Figure 4.4: PASCAL VOC results. **Left:** Percentage of windows searched as a function of detection rate, for all 20 categories. **Right:** Precision-recall curve drawn by pooling scored bounding boxes from all categories. Augmenting the LSVM detector [12] with our tag features noticeably improves accuracy—increasing the average precision by 9.2% overall.

mainly because PASCAL contains larger object instances, allowing the search to begin at a larger scale.

Modulating the Detector: Increasing Accuracy. Finally, we evaluate the accuracy boosts our features can provide for the state-of-the-art detector. As before, we pose a localization task, where the target object is known to be somewhere in the image, and the system must say where. Please note that this differs from the VOC standard evaluation, which requires methods to also make a decision about object presence. Since our method has access to image tags naming objects, it is trivial for it to reject false detections; thus, for a fair comparison, we score only the images that do have the object.

We found that because the LSVM detector performs so well for this

class	LSVM [12]	LSVM+tags	LSVM+Gist	Our gain
aeroplane	38.86	**39.12**	38.03	**0.67%**
bicycle	64.47	64.31	**64.51**	-0.11%
bird	11.79	**12.74**	12.06	**8.06%**
boat	18.79	**19.65**	18.57	**4.58%**
bottle	**35.67**	35.59	35.38	-0.22%
bus	55.62	**56.96**	55.26	**2.41%**
car	55.10	**55.14**	54.96	**0.07%**
cat	27.01	**29.04**	27.24	**7.52%**
chair	**22.31**	22.23	22.10	-0.36%
cow	32.27	**32.88**	32.70	**1.89%**
diningtable	**44.24**	42.50	43.31	-0.47%
dog	14.04	**16.23**	14.85	**15.60%**
horse	**60.98**	60.07	60.36	-1.49%
motorbike	52.90	**53.30**	52.46	**0.76%**
person	38.95	**39.00**	39.17	**0.13%**
pottedplant	18.81	**21.37**	19.13	**13.61%**
sheep	**31.64**	31.23	31.64	-1.30%
sofa	36.73	**37.72**	37.44	**2.70%**
train	48.77	**49.03**	**49.03**	**0.53%**
tvmonitor	**51.58**	51.34	51.41	-0.47%

Figure 4.5: AP scores on the PASCAL VOC 2007. Our method improves the localization accuracy of the state-of-the-art detector [12] for 13 out of 20 categories. Note the task is localization only, since the tags specify whether the object is present or not.

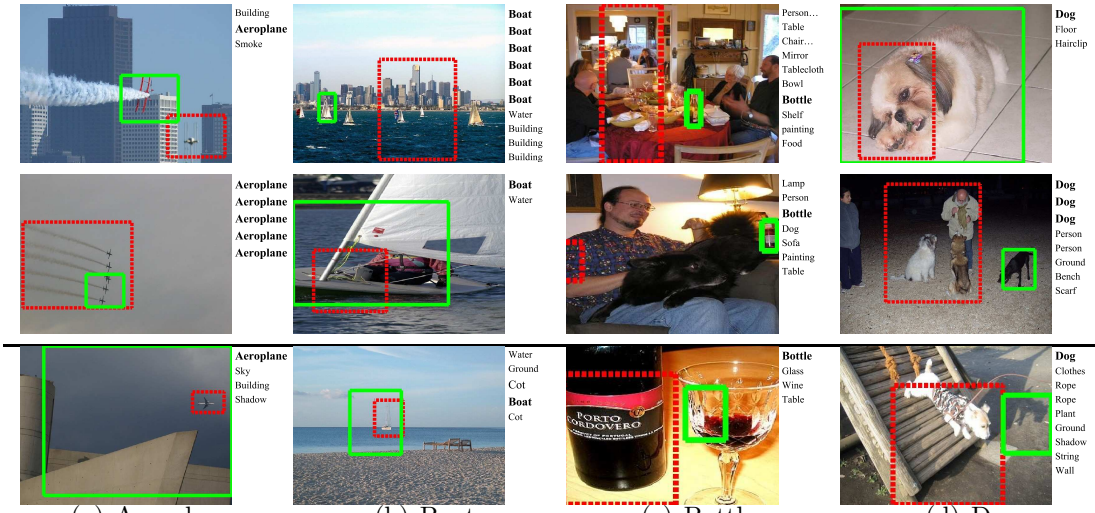
dataset, we can make the most useful accuracy improvements if we allow our features to re-rank the detector's most confidently scored windows, following [8]. Specifically, we rescore the top 500 detections (after non-maxima suppression) in a test image. We measure performance with the average precision (AP), the standard metric used in the VOC challenge [11].

Figure 4.4 (right) shows the precision-recall curve for the test images from all 20 categories, for which we obtain an overall 9.2% gain in AP. Figure 4.5 breaks down the per-class results. Our method improves the LSVM detector on 13 of the 20 categories. For a number of objects (`bird`, `boat`, `bus`, `cat`, `dog`, `pottedplant`, and `sofa`) we see quite good improvements. For others, gains are smaller (or negative). Using previous visual context-based detection results on VOC data as a reference point [7, 8], we believe the magnitudes of the improvements achieved are significant.

Figure 4.6 shows some qualitative examples (good and bad), comparing the top detection window according to LSVM (dotted red) vs. the top detection window according to LSVM+tags (solid green). Often the improvements our method makes are due to its accurate scale prediction.

We also observed some common properties of the categories for which our method works best: they tend to have examples with various other objects present/absent in different scales per instance, which gives context cues to our method about the target. For example, the `dog` image in the first row has the tag `hairclip` but mentions no other major objects, suggesting (through our features) that it may be a closeup shot. In contrast, the `dog` in the second row has tags for `person` and other objects, restricting the likely position and scale of the target. On the other hand, for an object like `diningtable`, our method is not useful—perhaps because the tag list tends not to differ, and the table is generally in the foreground if tagged at all.

We found that Gist does not perform as well on the VOC dataset as it



(a) Aeroplane (b) Boat (c) Bottle (d) Dog

Figure 4.6: Example detections on the PASCAL VOC. Red dotted boxes denote most confident detections according to the raw detector (LSVM); green solid boxes denote most confident detections when modulated by our method (LSVM+tags). For each category, the first two rows show good results, and third row shows failure cases. Most improvements come from our accurate scale prediction for the target object (e.g., see the `bottle` example in top row, or the two `dog` examples). Failure cases occur when the scene has an unusual layout according to those previously tagged in a similar way (e.g., see bottom-left `aeroplane`: we predict a large scale given the `building` tag, which usually implies the plane is close-by on the ground, but is not the case here).

did on the LabelMe dataset, perhaps because most of the images in the VOC are not "scenes", but instead object-centric. Further, the sparser coverage of scales, scene types, and viewpoints may make it more difficult for the single global Gist descriptor to exploit what context is there.

Chapter 5

Conclusions

Our main contribution is to derive implicit features from human-provided image tags, and to demonstrate their power for improving detection speed and accuracy. Overall, our results indicate that there is significant value in reading into the implicit cues found in human-tagged images. Not only can we find objects faster by noting the context hints from how they were named, but we can also improve the final accuracy of the detector itself.

Results on two realistic datasets with 25 diverse categories demonstrate that we can learn the tendencies of real taggers, even when (and perhaps because of the fact that) they are unaware of what the ultimate purpose of their tags will be. When designing our approach, we did not necessarily expect our tag-driven results to be competitive with a context model using visual cues; the fact that our method complements and at times even exceeds the well-known Gist cue is very encouraging.

Our feature design is inspired in part by studies on attentional behavior; the differences between how people scan images and what they decide to tag is not fully teased apart, though work is being done in this direction [1, 9, 10]. Furthermore, while our data contains nouns and objects that

did appear, generic tags can extend to any words or parts of speech, present or not. Interesting future work would be to predict which objects in the tags are most likely to be present, or considering multiple detectors' confidence.

Bibliography

[1] M. Ames and M. Naaman. Why We Tag: Motivations for Annotation in Mobile and Online Media. In *CHI*, 2007.

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching Words and Pictures. *JMLR*, 3:1107–35, 2003.

[3] T. Berg, A. Berg, J. Edwards, and D. Forsyth. Who's in the Picture? In *NIPS*, 2004.

[4] C. Bishop. Mixture Density Networks. Technical report, Neural Computing Research Group, Aston University, 1994.

[5] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[7] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative Models for Multi-Class Object Layout. In *ICCV*, 2009.

[8] S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert. An Empirical Study of Context in Object Detection. In *CVPR*, 2009.

[9] W. Einhauser, M. Spain, and P. Perona. Objects Predict Fixations Better than Early Saliency. *Journal of Vision*, 8(14):1–26, 2008.

[10] L. Elazary and L. Itti. Interesting Objects are Visually Salient. *Journal of Vision*, 8(3):1–15, 2008.

[11] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge, 2007.

[12] P. Felzenswalb, D. McAllester, and D. Ramanan. A Discriminatively Trained Multiscale Deformable Part Model. In *CVPR*, 2008.

[13] R. Fergus, P. Perona, and A. Zisserman. A Visual Category Filter for Google Images. In *ECCV*, 2004.

[14] C. Galleguillos, A. Rabinovich, and S. Belongie. Object Categorization Using Co-Occurrence and Location. In *CVPR*, 2008.

[15] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class Segmentation with Relative Location Prior. *IJCV*, 80(3):300–16, 2008.

[16] A. Gupta, J. Shi, and L. Davis. A Shape Aware Model for Semi-Supervised Learning of Objects and its Context. In *NIPS*, 2008.

[17] D. Hoiem, A. Efros, and M. Hebert. Putting Objects in Perspective. *IJCV*, 80(1), October 2008.

[18] M. Jamieson, A. Fazly, S. Dickinson, S. Stevenson, and S. Wachsmuth. Learning Structured Appearance Models from Captioned Images of Cluttered Scenes. In *ICCV*, 2007.

[19] T. Kadir and M. Brady. Saliency, Scale and Image Description. *IJCV*, 45(2):83–105, June 2001.

[20] C. Lampert, M. Blaschko, and T. Hofmann. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In *CVPR*, 2008.

[21] L. Li, G. Wang, and L. Fei-Fei. Optimol: Automatic Online Picture Collection via Incremental Model Learning. In *CVPR*, 2007.

[22] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. *Towards Category-Level Object Recognition*, chapter Object Detection and Localization Using Local and Global Features. LNCS, 2006.

[23] M. Peot and R. Shachter. Learning From What You Don't Observe. In *Uncertainty in Artificial Intelligence*, 1998.

[24] B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a Database and Web-Based Tool for Image Annotation. *IJCV*, 77(1):157–173, May 2008.

[25] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting Image Databases from the Web. In *ICCV*, 2007.

[26] A. Sorokin and D. Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *CVPR Workshop on Internet Vision*, 2008.

[27] M. Spain and P. Perona. Some Objects Are More Equal Than Others: Measuring and Predicting Importance. In *ECCV*, 2008.

[28] B. Tatler, R. Baddeley, and I. Gilchrist. Visual Correlates of Fixation Selection: Effects of Scale and Time. *Vision Research*, 45:643, 2005.

[29] A. Torralba. Contextual Priming for Object Detection. *IJCV*, 53(2):169–191, 2003.

[30] S. Vijayanarasimhan and K. Grauman. Keywords to Visual Categories: Multiple-Instance Learning for Weakly Supervised Object Categorization. In *CVPR*, 2008.

[31] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *CVPR*, 2001.

[32] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *CHI*, 2004.

[33] J. Wolfe and T. Horowitz. What Attributes Guide the Deployment of Visual Attention and How Do They Do It? *Neuroscience*, 5:495–501, 2004.

Vita

Sung Ju Hwang was born in Seoul, Korea on February 15th, 1982, the son of Buhyun Hwang and Hyun Sook Park. After completing his high school studies in Gwangduk High School in Gwangju, South Korea, he entered the Seoul National University, where he received the degree of Bachelor of Science in 2008. The same year, he entered the Graduate School at the University of Texas at Austin.

Permanent address: Moa Tower Apt. 101-Dong 1202-Ho, Hwajung 4-Dong, Seo-Gu, Gwangju, South Korea

This thesis was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.