Large-Scale Image Annotation using Visual Synset

David Tsai[1,2]
caihsiaoster@gatech.edu

Yushi Jing[2]
jing@google.com

Yi Liu[2]
yliu@google.com

Henry A.Rowley[2]
har@google.com

Sergey Ioffe[2]
sioffe@google.com

James M.Rehg[1]
rehg@gatech.edu

[1] Computational Perception Lab, School of Interactive Computing
Georgia Institute of Technology, Atlanta, GA, USA
[2] Google Research, Mountain View, CA, USA

Abstract

We address the problem of large-scale annotation of web images. Our approach is based on the concept of visual synset, which is an organization of images which are visually-similar and semantically-related. Each visual synset represents a single prototypical visual concept, and has an associated set of weighted annotations. Linear SVM's are utilized to predict the visual synset membership for unseen image examples, and a weighted voting rule is used to construct a ranked list of predicted annotations from a set of visual synsets. We demonstrate that visual synsets lead to better performance than standard methods on a new annotation database containing more than 200 million images and 300 thousand annotations, which is the largest ever reported.

1. Introduction

Many previous works in image annotation have addressed the problem of predicting generic labels corresponding to parts or objects within the scene, using standard datasets such as Corel5Kx or IAPR TC12. Most of these datasets contain on the order of thousands of images and hundreds of labels. In this paper, we address the annotation problem in the context of web image search, where the goal is to predict the labels that a user might employ in searching for a given image. For this purpose we have constructed from the web an image annotation dataset of unprecedented size, containing more than 200 million images and 300 thousand labels.[1]

Recently, nearest neighbor-based methods have gained significant attention and have achieved state-of-the art results for image annotation tasks. While the nearest neighbor role is appealing due to its conceptual simplicity, in order to make the method scalable for large datasets it is necessary to employ techniques like KD-tree or metric tree to group related images. In contrast, we propose to construct *visual synsets*, which are collections of images that are visually similar and share a consistent set of labels.[2] We show that concept class defined by a visual synset is easier to learn than the concept class defined by a single label.

Each of the leaves in Figure 1 is a visual synset. Our visual synset representation is inspired by the recent work of ImageNet [8], which builds a large scale ontology of images by manually associating a well-chosen set of images with each synset in the WordNet ontology. In contrast to this work, our goal is to automatically generate visual synsets from a very large collection of images and annotations collected from the web. Thus rather than using a fixed ontology and a hand selected set of images, we automatically identify a large set of images which are associated with a particular concept, and then cluster those images into groups which are visually similar. For each of these groups we then learn an associated set of labels which can be used to predict annotations.

We compare our visual synset approach to two standard methods which define the current state-of-the-art. The first method uses linear SVM [8, 7], in which a model is trained for each pre-defined attribute label. We refer to this as the "category level" approach. This method achieved first place in the ImageNet Large Scale Visual Recognition Challenge 2010 [2]. Another competitor is nearest neighbor matching with label transfer, which uses a non-parametric data-driven model [23, 24]. We refer this as the "instance level" approach. We explore how a linear classifier, combined with

[1] This derives from Google Image Swirl project, see [16] for details.

[2] Our concept of visual synset is a natural visual extension of the original concept of synset, which groups English words into sets of synonyms. In contrast, [27] uses the term "visual synset" to describe a concept which is different from ours, essentially an extension of bag of words.

a simple voting scheme, can leverage the visual synset representation to achieve superior performance in comparison to these other representations This paper makes three contributions:

- We propose “visual synset” representation for large scale image annotation which models a collection of images and their associated annotation labels.
- We demonstrate the superior performance of the visual synset approach on an annotation task using a large scale dataset.
- We introduce a large-scale annotation database containing 200 million images and 300 thousand labels. This is the largest and most complex dataset currently available for image annotation research.

2. Related Work

The goal of image annotation is to assign a set of labels, or keywords, to an image. The annotations can be defined in different ways. In [3], the goal is to associate words with specific image regions. In [10], attributes are learned to describe the objects. [24] and [25] have the most similar goal to ours, which is to provide a “description” of the image which could help with visual search. Our work covers a larger label space than these previous methods.

Many algorithms have been proposed to solve the image annotation task. Some works use generative models such as latent Dirichlet allocation [3], probabilistic latent semantic analysis [19], and hierarchical Dirichlet processes [26]. It is unclear how these methods can be scaled to achieve good performance on web images, as they need to learn the joint distribution over a very large space of features and labels. In contrast, our method is easy to parallize and thus is suitable for web-scale applications. Another nice work [25] learns a low-dimensional joint embedding space for images and annotations. Our work uses a similar “embedding” spirit, but the explicit grouping of images and annotations in our visual synset representation is more interpretable.

Our visual synset representation is similar to discriminative models using winner-takes-all among all 1-vs-all classifiers [8], in which the model is learned for each pre-defined category. Such a representation has two potential drawbacks. First, a category level representation is too coarse and many categories are ambiguous and diverse, which leads to a difficult learning problem if we train a single model for each category. Second, the winner-takes-all strategy requires the true model be more confident than all other models, which becomes very difficult as number of labels increases. In contrast, by constructing visual synsets and using a voting scheme, we obtain a representation with better discriminative power.

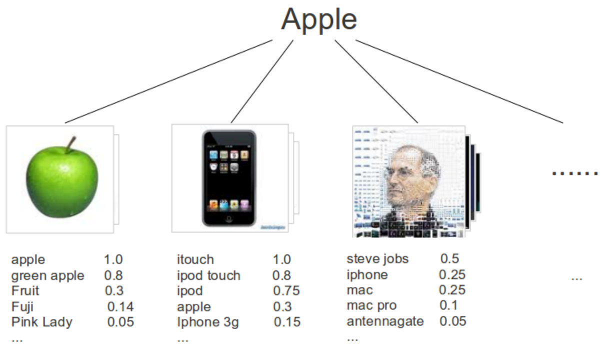


Figure 1. **Illustration of Visual Synsets**. An object class can be divided into various elementary partitions which share compact visual similarity, each associated with a set of key words to describe the semantic meaning.

Our problem formulation is also closely related to many multi-label prediction works from the machine learning community. In these works, hierarchical structures such as tree or DAG are used to model the dependencies between labels [5, 4]. Another related representation is “instance level” prediction based on k-NN and label transfer [24, 18]. Basic nearest neighbor search does not scale to large database sizes, thus approximation is inevitable and is usually based on grouping images. Our method is similar in that it also involves grouping images. However, our grouping process leverages structure in both the label space and the feature space. We demonstrate experimentally that visual synsets are superior to standard approximate k-NN.

3. Visual Synset Representation

We now describe our method for learning visual synsets. The first step is to identify visually-similar groups of images. The next step is to associate each image grouping with a weighted set of labels. Examples of these weighted label sets are provided in Figure 1 for three different image groupings. We now describe these steps in more detail.

3.1. Automatic Image Collection for Visual Synsets

Our goal is to automatically collect a set of images and cluster them to provide the basis for defining a group of visual synsets. Each synset should be associated with a well-defined visual concept, so the problem of predicting whether a query image is a member of the synset is straightforward. However, it is not sufficient to generate candidate synsets by simply clustering a large set of images based on visual feature similarity alone. This is because we also want each visual synset to correspond to a well-defined semantic concept, as defined by the weighted set of labels associated with the synset. Intuitively, each synset is voting for a relatively small set of labels with high weights, and the collection of synsets spans the total label space for the annotation problem.

To accomplish this goal, we adopt the strategy in [16] to collect images for visual synsets. We first use the existing label space in conjunction with image information to partition the data: For each of our 300K possible labels, we use Google Image Search to retrieve up to 1K images that are associated with that label. Note that it is quite possible that the same image will be returned in multiple searches and included in multiple synsets. For each set of returned images (sharing the same label), our goal is to partition them into groups that are visually compact. The starting point for clustering is the computation of a pairwise similarity measure between each pair of images, obtained from a linear combination of various image features such as color, shape, local features, face signatures, etc, as well as text features. We follow the approach of [12] to learn the linear weights.

We cast the problem of finding sets of visually distinctive images as a clustering problem, given computed pairwise image similarity. Formally, we denote the training set indices as $X = \{x_1, x_2, ..., x_N\}$, and an exemplar or prototype set indices $\mathcal{C} = \{c_1, c_2, ..., c_K\}$,where $\mathcal{C} \subset X$. Each image x_i is associated with an exemplar c_k, or $L(x_i) = c_k$. We use affinity propagation [11] (AP) for clustering, as it is straightforward to incorporate prior information, such as the relevance ranking of the retrieved images, into the clustering framework (by adjusting the initial preference scores).

Our goal is to obtain set of exemplars $\mathcal{C}$ and their associated images that maximizes the following energy function:

$$F(\mathcal{C}) = \sum_{i=1}^{N} S(x_i, L(x_i)) + \sum_{i=1}^{N} \delta_i(\mathcal{C}) \qquad (1)$$

where $S(x_i, x_j)$ is the pairwise image similarity, and $\delta_i(\mathcal{C})$ is a penalty term that equals $-\infty$ if some image x_k has chosen c_i as its exemplar, without c_i having been correctly labeled as an exemplar:

$$\delta_i(\mathcal{C}) = \begin{cases} -\infty, \text{if} \quad L(c_i) \neq c_i \ \ but \ \exists k : L(x_k) = c_i \\ 0 \ \ \text{otherwise.} \end{cases} \qquad (2)$$

Since the labels obtained by image search are inherently noisy, not all of the images are good exemplar candidates. We therefore compute an *a prior preference* score of each image being an exemplar given this query based on its relative ranking in the search engine results, and use this to initialize preference scores for clustering.

This clustering method requires that we determine the correct number of clusters automatically. For visual categories, image similarity varies substantially and it's hard to decide a universal K that is suitable for all categories. We address this problem by using the minimum distance between the clusters to automatically select K. We observed that the minimum distance is a more robust criteria than K. In our experiments this global threshold was set manually through a separate validation set. After clustering, we removed clusters that have too few images.

Table 1. **Example of labels in the same visual synset**

label	**other labels that appear in the same synset**
touch screen phone	philips, nokia touch, iphone 3g, mobile, garmin, gps, gsm
egypt air	airlines, embraer 170, airbus, 737, boeing, star alliance
tiger tattoo	dragon tattoo design, traditional japanese tattoos
facebook	facebook profile layout, facebook page, facebook screenshots

We use the images in the same cluster to form a visual synset. Note that in general we do not expect each visual group to uniquely define a single semantic concept. There can be multiple groups representing the same concept, and their outputs will be combined through a voting process.

3.2. Label Sharing for Visual Synsets

Given a set of image clusters that form the basis for visual synsets, the next step is to assign labels and weights to each one. Just as images can be shared by different visual synsets, labels can be shared as well. This is important for robustness, as it allows multiple visual synsets to contribute votes for a particular label. The first step is to identify the total set of labels that are associated with each image in each synset, as a result of the image search process that initialized the clustering stage. We then assign weights based on the intuition that the most frequently-occurring labels across a cluster of images are the most important labels for the visual synset, and should therefore receive the highest weight.

To obtain the weights, we calculate the term frequency-inverse document frequency (TF-IDF). In each synset, the term count TF is given by the number of times a given label appears in that synset. It is normalized to prevent bias towards synsets that have more images: $TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$,where $n_{i,j}$ is the number of occurrences of the label l_i in synset S_j. The inverse visual synset frequency IDF is calculated as: $IDF_i = log\frac{|I|}{1+|\{S:l_i \in S\}|}$,in which $|I|$ is the number of synsets in the corpus, and $|\{S : l_i \in S\}|$ represents the number of visual synsets where the label l_i appears. Then the final score is $S_{i,j} = TF_{i,j} * IDF_i$.

Figure 1 gives several examples of synset labels and their associated weights. Although the label "apple" appears in all three synsets, it has a different weight in each one, due to the differences in the visual concept which is being modeled. Each synset is described by a ranked, weighted list of labels. Note that this is a distinction between our approach and previous works which use only a single label or multiple equally-weighted labels to describe a class [8, 7]. Table 1 gives additional examples of synset labels, demonstrating that labels tend to be associated with a consistent visual concept which is shared by a set of images.

4. Annotation using millions of Visual Synsets

Given the ability to automatically generate visual synsets by clustering annotated web images, we now describe a procedure for using millions of visual synsets to predict annotations for unseen images using linear SVM classifiers and a simple voting scheme.

4.1. Learning Linear Model

After we generate the visual synset, the goal is to learn a discriminative model for each one to predict the visual synset membership. We focus on linear SVM classifiers since the scale of our problem makes it impractical to use nonlinear models.

During training, each image is represented by a feature vector. The particular choice of feature representation is not the key focus of this paper. The method we describe can be used with any feature representation. For our experiments, we compute various features including color histogram, wavelet, local binary patterns, and spatial pyramid texton over multiple scales. They are then quantized by clustering on a large corpus of images, which has a sparse vector representation. L1 Hash is then applied to make the sparse features dense, and KPCA with Histogram Intersection Kernel is used to further reduce dimensionality and place the data in a Euclidean feature space.

Given the training data, we train a one-vs-all linear SVM model for each visual synset. Due to the scale of our problem, we need to find an efficient way to train the linear model. Here we adopted the primal estimated sub-gradient optimization algorithm introduced in [22]. The run-time does not depend directly on the size of the training set.

In practice this iterative algorithm converges very fast and the number of iterations required to obtain a solution of accuracy ϵ is $O(1/\epsilon)$ [22]. In our experiments the time bottleneck is not the computational cost of learning, instead it is the time required for loading the data.

4.2. Prediction by Voting

The final goal for annotation is to predict a ranked list of key words for an input image. In our work, the ranking is naturally incorporated in the visual synset representation and can be generated with a very simple voting scheme.

We use a vector $\mathbf{K}$ to denote the label for a visual synset and the length of $\mathbf{K}$ is the number of all possible labels. If label j exists in visual synset i, the jth dimension of $\mathbf{K}$ would be the corresponding S computed in section 3.2, otherwise the value is 0 if the label is not in the synset.

For an input image, we first calculate its feature $\mathbf{x}$ and then pass it to all the visual synset models. If the response is above a threshold T, we accept this synset. Finally we simply do label voting by aggregating the label information associated with all the accepted visual synsets. The label vector $\mathbf{L}$ is defined as:

$$\mathbf{L} = \sum_{i=1}^{n} I(\mathbf{w}_i \cdot \mathbf{x} + b_i > T) \sum_{j=1}^{m_i} \mathbf{K}_{i,j} \tag{3}$$

In which $\mathbf{w}$ and b are parameters learned by linear SVM, n is the number of visual synsets, m_i is the number of labels in each synset. $I(\cdot)$ is the indicator function that only accepts the responses that are above the threshold.

In contrast to previous exemplar-based work using learned distances, we discard the SVM output score information and make a binary decision for each visual synset. It is straight-forward to directly compare the SVM output score of all the 1-vs-all models and predict label ranking purely based on the score. However, as models are trained with different instances, there is no theoretical basis for comparing the SVM scores of different models. Here we avoid this problem by discarding the SVM score and treating all the accepted exemplars equally.

One benefit of our method is that it can boost the ranking of concrete labels (e.g fuji apple) in comparison to general labels (e.g food). In contrast, when using an ontological model like WordNet, as in [23], the nodes at the top of the tree will receive more votes than the nodes at the bottom. As a result, it is necessary to manually select the level in the tree and compare the votes of different nodes at that level. Depending upon the query, the words of interest for annotation might appear in different levels. In our method, by grouping images into specific visual synsets and applying a label weighting technique, we ensure that the concrete labels which appear more frequently in that synset will receive appropriate weight.

5. Experimental Results

We conducted two sets of experiments. The first set was designed to analyze the key assumptions underlying our construction of visual synsets and assess their discriminative power. The second set of experiments addressed the application of the visual synset approach to the large scale image annotation problem. We compare our approach to the two standard methods for annotation: nearest-neighbor classifier, and linear SVMs trained for each annotation label. Our results demonstrate the superior performance of the visual synset approach. Finally, we evaluate the generalization ability by testing on a small scale dataset that was independently collected.

We collected a dataset containing 200 million images and 300 thousand possible labels. We first constructed a dictionary containing 300 thousand common English words which are frequently used in web search. Then for each word we downloaded up to 1000 images from Google Image Search as described in Section 3.1. The images are automatically annotated according their queries, tags, and other processed meta-data. This dataset is the largest ever assembled for the image annotation problem, both in terms of number of images and more importantly in terms of the size of the label space. Detailed description of our dataset can be found from our project website. [3]

We use the same set of visual features in all of our experiments. All of the important parameters are selected using a separate validation set and are fixed in all experiments. We also use the same sub-gradient based optimization algorithm to train all of the SVM classifiers.

5.1. Relationship between Semantic Similarity and Visual Similarity

The success of our visual synset approach is based on the ability to construct visually-compact sets of images that share a consistent set of labels. We have designed an experiment to measure the compactness of our synsets. The starting point is the development of similarity measures for visual features and label sets.

Given a set of images associated with a keyword (as used in Section 3.1 to initialize the search for visual synsets via clustering), each image is represented by a dense vector feature $\mathbf{f}$. We compute a measure of visual dissimilarity for the collection of images by summing and normalizing the pairwise Euclidean distances between all feature vectors: $D_v = \frac{1}{K}\sum_{i<j} ||\mathbf{f^{(i)}} - \mathbf{f^{(j)}}||$

For each image, we have a list of labels which can be represented as a binary vector $\mathbf{b}$ indicating whether a specific word is present or not in this image's labels. Likewise for all images in the group, we compute the pairwise inner product for text vectors, and sum them up and normalize it by total word number M to form the semantic similarity of this image group: $D_s = \frac{1}{M}\sum_{i<j} < \mathbf{b^{(i)}}, \mathbf{b^{(j)}} >$. The higher value, the greater the semantic coherence.

We evaluate the compactness of 1074 image clusters, which are generated by starting from a list of 1426 of the most popular words for image searches covering a wide range of content, and then discarding words that have too few images associated with them. From the scatter plot in Figure 2, we can get the general trend: the visual distance decreases with the increase of the semantic similarity. This is verified from the blue curve which shows the cubic regression of the result using all 300K categories. It shows there is positive correlation between semantic similarity and visual similarity. This result is consistent with a recent paper [9] which contains similar experiments using data from ImageNet.

5.2. Discriminative Power of Visual Synset

Before applying visual synsets to the image annotation task, it is important to know the discriminative power of a single visual synset. We evaluate this by looking at the discriminative power of the linear SVM classifier trained on each visual synset. For each visual synset we randomly sample 70% for training and 30% for testing. We randomly sampled negative training images, keeping twice as many negatives as positives. We also sample the same amount of negative testing images as positive testing images. The goal is to test whether a given image should belong to that visual synset or not. We computed AUC (the area under the ROC curve) for each cluster and evaluated the classification results by histogram of AUCs.

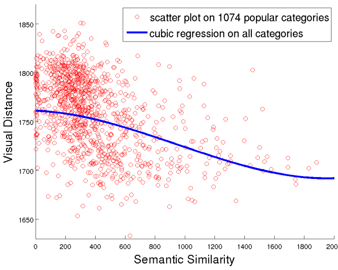


Figure 2. The distribution of visual and semantic similarity pairs on 1074 popular categories(scatter), and cubic regression (blue curve) on all 300K categories.

Since the visual synset leverages both visual and label information, we make comparisons to alternative partitioning strategies based on visual features alone and labels alone. In the case of labels, we directly train an SVM model to learn the category of all images associated with a single keyword. This is analogous to a standard image categorization task, where the keyword can be viewed as a category label. In the case of purely visual partition, we group together the images associated with a set of randomly chosen keywords, and then cluster those images in order to obtain image groups which are based on visual similarity alone from a randomly-chosen set of images.

The results are shown in Figure 3. It is clear that partition based on label information alone leads to the lowest distribution of AUC scores, which reflects the semantic gap. When the data is partitioned based on visual features, the AUC scores are improved. When we use semantic information to help the visual clustering, the resulting visual synsets have the best discriminative power.

5.3. Large-Scale Image Annotation

5.3.1 Evaluation Metric

It is known that for the image annotation task, precision is usually more important than recall. It means we can sometimes accept missing labels, but the labels we predict must be right. In addition, the ranking detail also matters because the users always want the most accurate annotation to appear first. Following the standard metrics [25], we provide precision at the top k of the list ($p@k$) and mean average precision, which favors true positives appearing at the top

[3] http://cpl.cc.gatech.edu/projects/VisualSynset

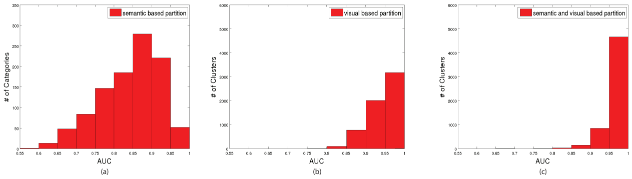


Figure 3. **(a)** Partition based on category. **(b)** Partition based on visual feature. **(c)** Partition based on both visual feature and category.

of the list of annotations. The most frequent annotation in the dataset appears less than 0.01% of the time. We select another 20,000 web images for testing and remove their duplicates in the training set.

5.3.2 Baselines

We compare our proposed approach with the two most popular formulations: category level and instance level representation. Although simple, these representations usually produce the state-of-the-art results on web-scale image label prediction applications.

Category level: We directly train a 1-vs-all classifier for each label. Given a new image, it will get a score from each classifier. Then we will generate the ranking list by sorting the predicted labels according to their scores.

Instance level: Since there is no image structure to model at instance level, we use the nearest neighbor method. We tested approximate k-NN because exact k-NN is clearly not scalable. We adopt the standard metric tree method [6, 17] with a tree of depth p. The tree is balanced by using the median value of data on each node as the threshold, and the leaf node only contains $n/2^p$ of the total points. For a given new point, we first identify the node that contains the point and then compute exact nearest neighbor search within that leaf node. In order to improve the robustness, we use the spill tree [17], which allows overlap on the boundary when partitioning the data. When doing annotations we aggregate the weighted sum of labels from all of the neighbors to predict the final ranked list. In our experiments we used $p = 8$ and 0.1 as the spill factor.

5.3.3 Results

We give quantitative comparison results between our method and the two baselines in Table 2 and Figure 4(a). More qualitative results are shown in Figure 3. Even more examples are included in the supplementary material. We can see that our method can predict higher quality annotations and rank them more accurately in comparison to the two baselines.

The category level approach performs the worst. This is unsurprising due to the increasing difficulty of making correct category label predictions as the number of labels increases. For very large numbers of labels, it is difficult for the right model to beat all of the others. In the first example in Table 3, it is reasonable to have a high score on wrong label "solar power satellite" because the rectangle and metal elements make it look like a solar panel. The instance level approach can achieve better results than category level. However, the distance between images is computed purely from visual features and is not reliable for some far points.

Table 2. **Summary of Results on Web-data**

Algorithm	p@1	p@5	p@10	MAP
Category-level	0.51%	0.36%	0.35%	0.67%
Instance-level	1.13%	0.82%	0.64%	1.86%
Visual synset	1.55%	1.20%	0.97%	2.66%

By constructing visual synsets, our method avoids several problems with the above baselines. In each synset, the visually-similar images and semantically-consistent labels are grouped together, making the output votes more constrained than an instance level representation. With the label sharing and voting scheme, we avoid directly comparing the svm scores. Instead, the right label would appear in many accepted models and will naturally get a higher rank through voting.

Parameter setting and alternative choices: The minimum distance in affinity propagation was selected based on how well the resulting clusters satisfies a set of similar / dissimilar constraints manually collected for a small set of queries, and was set to 13.5 in our experiments.

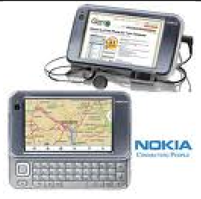
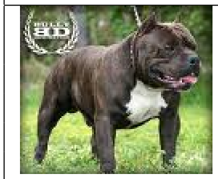
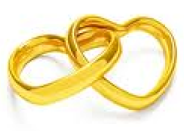
Another choice in the implementation is the clustering technique. Although affinity propagation(AP) is good at selecting exemplars and thus is a good choice for the retrieval task, it is still interesting to know how other techniques perform. We conducted comparisons with hierarchical agglomerative clustering (HAC), spectral clustering (SC)[20] and K-Means, using experiments similar in [15] to test the ability to select proper exemplars. We observed that HAC and SC perform slightly worse than AP, while K-means performs significantly worse than the others.

Computational details: It is a big challenge to deal with such a huge amount of data. In our experiments, we leverage the MapReduce implementation and computer clusters to train visual synset in parallel. We trained about 2 million linear SVM classifiers, one for each visual synset and the mean training time was 3.5s per visual synset. We used a cluster of 2000 nodes and the complete process took less than 45 hours, in which I/O and feature lookup were the biggest sources of overhead.

5.4. Generalization Capability Test

Standard datasets are not suitable to evaluate the web-scale image annotation task because the labels in these datasets are usually too limited and do not contain many

Table 3. **Examples of the annotations of three approaches on web data. Ground truth is in the last column.**

Image	Category level	Instance Level	Visual Synset	Ground Truth
	usina nuclear, rain water harvesting, pressure washing, us military, solar power satellite, water pollution, trichy airport, future computers, herniated disc, pollution pictures, roof construction	india, map of india, india map, india maps, potitical map, maps of india, nokia n810, cartoon, potitical map india, cartoons, n810 internet tablet, nokian810	nokia, nokia n900, n900, nokia internet, gps, qwerty, n810 nokia, keyboard, nokian810, nokia n810, internet tablet, japanese architecture n810 internet tablet	mobile, nokia n93, gps, internet tablet, n810 internet tablet n93, nokia, n810, nokia n810 internet tablet nokia n810...
	cane corso puppies, molosser, baby donkey, pitbull fights, ezel, heste, silverback gorilla, gambar monyet, poney shetland, chien pitbull, pygmy hippo, albino horse	biggest dog, biggest dog ever, biggest dog in the world, caucasian shepherd dog, huge dogs, pitbull, dogs, big dog breeds, donkey	dog, terrier, pitbull, bull terrier, dogs, pit bull biggest dog, american pitfull terrier, pit, puppies, american pitbull dog breeds, pups	pitbull dogs, american bully, pitbull puppies pitbull pups pit bull pit bulls...
	yellow, inele, platano, gold jewellery, maiskolben, lady's gold rings, gold bar, jelly shoes, pasta shapes, magnesio, wedges, gold bangles	bananas, banana, thermal, sopron map taco, wedding rings ring, lemon, salsa, yellow gold, taco pics, taco bell tacos	ring, rings, yellow gold, wedding rings, diamond ring, engagement rings, princess cut engagement ring, gold jewelry, antique rings	jewellery gold, gold jewellery, gold jewellery set, rings, ring, wedding rings engagement ring...

popular web annotations such as celebrities or products. However, since our visual synset representation is designed to be general purpose and is blind to specific datasets, we conducted a challenging test of its generalization capability, by using it to predict annotations for the standard dataset MIR-Flickr-25000 [14]. There are several challenges in this experiment: (1) the standard dataset has a much smaller label space in comparison to our web data. Specifically, all 25000 images are annotated with 25 general topics like sky, people, sunset etc. (2) Image annotations are provided by humans, using a very different process from our work. Note that in this experiment we use the MIR images for *testing only*, and do not include any training step with this dataset.

The goal of this experiment is to provide a challenging test of the generalization ability of our method, by testing on a dataset generated *using a completely different procedure* from our training set. In testing, we simply passed all the images into our system and outputed a set of labels for each image. Since the output label space is significantly larger than the one in MIR-Flickr, we constructed a mapping between the two spaces.

We first created a large database of relationships between labels from webpages and documents using lexico-syntactic pattern matching [21] by looking for patterns such as: "i is a k" or "k is i's parent". For example, i could be "apple" and k could be "fruit". We leverage such a database to do label mapping. For each label in the MIR dataset, we find all its children in the Is-A relationship. If our system produces the child, we will map it to its parent label. For example, if our system produces "Honda Accord", it will be mapped to the label "car" which exists in the ground truth. For the baseline method, we use the same mapping scheme.

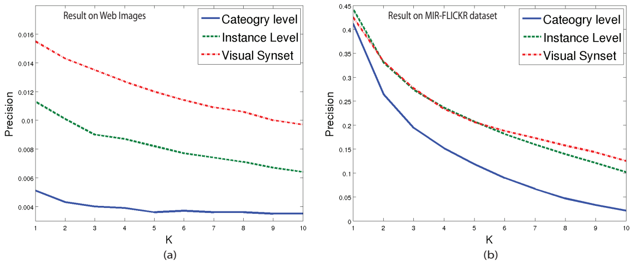


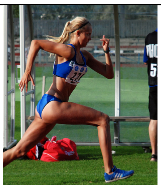
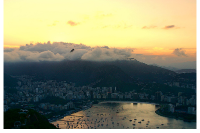
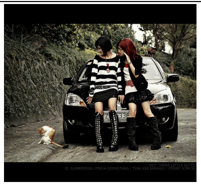
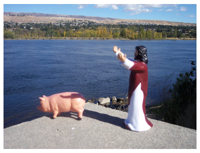
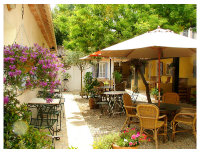
Figure 4. **(a)** Annotation result on web images **(b)** Generalization capability test on MIR-FLICKR dataset

Since such mapping is in general noisy and incomplete, it is possible that some good annotations cannot be converted to any word in the ground truth. However, our method still can produce reasonable annotation results, as shown in Table 4. We also compared our method with the two baselines described in the previous section. The quantitative results are shown in Figure 4. We can see for this dataset with small label size, our method can achieve as good result as instance level reresentation and better results than category level. When using the top 5 labels, it can achieve a MAP of 32.8%. It is unfair to make direct comparisons with other methods because we are using training images from different sources. But to give a general impression, in the photo annotation task in ImageCLEF2010 [1] using this dataset, [13] showed MAP of 36.4% based on visual features. This experiment demonstrates the good generalization ability of our visual synset representation.

6. Conclusion

In this work, we proposed the Visual Synset representation for large-scale image annotation. Visual synsets cap-

Table 4. **Predicted keywords for images from MIR-Flickr dataset**

					
Predicted annotations	people,portrait female,animal plant life	lake,people structures animals,sunset	people,car transport animals,bird	animals,food lake,river people	flower,tree plant life food,bird
Human annotations	people,female structures plant life	clouds,sea sky, sunset structures	animals,car,female people,plant life portrait,transport	animals,clouds,lake plant life,river,sky water	flower,tree plant life structures

ture both low-level visual coherence and high-level semantic consistency, and produce better prediction performance than competing methods based on category-level classification and instance-level nearest-neighbor voting. Our work introduces a dataset collected from the web which contains 200 million images and 300K labels which covers a wide range of real-world searching behavior. We believe this is the largest dataset ever assembled for image annotation. We conducted extensive experiments to characterize the performance of our method.

Acknowledgements. This research was supported by NSF Grant 0960618 and Google Research. We would like to thank Nemanja Petrovic for his help on Approximate k-NN and Samy Bengio for many useful discussions.

References

[1] http://imageclef.org/2010. 7

[2] http://www.image-net.org/challenges/lsvrc/2010/. 1

[3] K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 2003. 2

[4] W. Bi and J. T. Kwok. Multi-label classification on tree- and dag-structured hierarchies. *ICML*, 2011. 2

[5] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *JMLR*, 2006. 2

[6] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. *VLDB*, 1997. 6

[7] J. Deng, A. Berg, K. Li, and F. fei Li. What does classifying more than 10,000 image categories tell us? *ECCV*, 2010. 1, 3

[8] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F.-F. Li. Imagenet: A large hierarchical image database. *CVPR*, 2009. 1, 2, 3

[9] T. Deselaers and V. Ferrari. Visual and semantic similarity in imagenet. *CVPR*, 2011. 5

[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, 2009. 2

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007. 3

[12] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. *ICCV*, 2007. 3

[13] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. *ICCV*, 2009. 7

[14] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. *MIR*, 2008. 7

[15] Y. Jing, M. Covell, and H. Rowley. Comparison of clustering approaches for summarizing large population of images. *ICME VCIDS*, 2010. 6

[16] Y. Jing, H. Rowley, C. Rosenberg, J. Wang, and M. Covell. Visualizing web images via google image swirl. *NIPS workshop on Statistical Machine Learning for Visual Analytics*, 2009. 1, 3

[17] T. Liu, A. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. *NIPS*, 2004. 6

[18] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *IJCV*, 2010. 2

[19] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: Constrining the latent space. *ACM Multimedia*, 2004. 2

[20] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002. 6

[21] S. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. *AAAI*, 2007. 7

[22] S. Shalev-Schwartz, Y. Singer, and N. Srebro. Pegasos: primal estimated sub-gradient solver for svm. *ICML*, 2007. 4

[23] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *PAMI*, 2008. 1, 4

[24] X. Wang, L. Zhang, M. Liu, Y. Li, and W. Ma. Arista - image search to annotation on billions of web photos. *CVPR*, 2010. 1, 2

[25] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *Machine Learning Journal*, 2010. 2, 5

[26] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical dirichlet process model. *MDM*, 2008. 2

[27] Y.-T. Zheng, M. Zhao, S.-Y. Neo, T.-S. Chua, and Q. Tian. Visual synset: Towards a higher-level visual representation. *CVPR*, 2008. 1