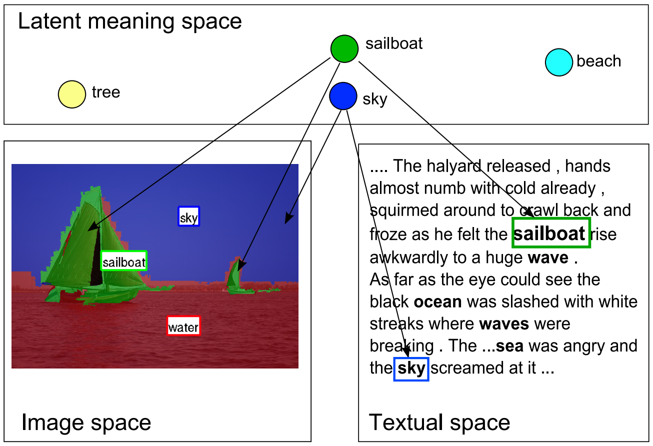
Connecting Modalities: Semi-supervised Segmentation and Annotation of Images Using Unaligned Text Corpora

Richard Socher   Li Fei-Fei
Dept. of Computer Science
Stanford University
richard@socher.org, feifeili@cs.stanford.edu

Abstract

We propose a semi-supervised model which segments and annotates images using very few labeled images and a large unaligned text corpus to relate image regions to text labels. Given photos of a sports event, all that is necessary to provide a pixel-level labeling of objects and background is a set of newspaper articles about this sport and one to five labeled images. Our model is motivated by the observation that words in text corpora share certain context and feature similarities with visual objects. We describe images using visual words, a new region-based representation. The proposed model is based on kernelized canonical correlation analysis which finds a mapping between visual and textual words by projecting them into a latent meaning space. Kernels are derived from context and adjective features inside the respective visual and textual domains. We apply our method to a challenging dataset and rely on articles of the New York Times for textual features. Our model outperforms the state-of-the-art in annotation. In segmentation it compares favorably with other methods that use significantly more labeled training data.



.... The halyard released , hands almost numb with cold already , squirmed around to crawl back and froze as he felt the **sailboat** rise awkwardly to a huge **wave** .
As far as the eye could see the black **ocean** was slashed with white streaks where **waves** were breaking . The ...**sea** was angry and the **sky** screamed at it ...

Figure 1. Our method maps image regions and words to a latent meaning space using context and adjective features. For instance, the word *sailboat* co-occurs with *water, wind and the adjective white* while image regions of a *sailboat* tend to also co-occur with these objects and inside white image regions. If mappings are close in meaning space, the items are likely to be instances of the same underlying semantic concept and can be used for segmentation and annotation. (Image is a test result. Text is extracted from training corpus.)

1. Introduction

In many domains of human cognition, we use context to disambiguate the meaning of items. For instance, in a text corpus we might interpret the word *blast* as an explosion of dynamite or as a generally exciting experience, depending on whether it co-occurs with the word *TNT* or *fun*. As pointed out by [18] among many others, context also helps people and computers with searching and recognizing objects more efficiently. Several methods in computer vision exploit context to recognize objects in scene images (annotation) and to provide a pixelwise localization of these objects (segmentation) [23, 11, 13, 19].

These approaches use context mostly to relate objects to each other inside the visual domain. Our model connects visual and textual modalities through a common latent meaning space. Image regions (which are clustered into visual words) and words extracted from semantically related corpora are mapped to this space using context and feature similarities that are observable inside the respective domains. The model uses the fact that words tend to have the same co-occurence patterns in the textual and visual domains. For instance, the word *sailboat* co-occurs with *water*, *wind* and the adjective *white* while image segments of a *sailboat* tend to also co-occur with segments of these objects and white image regions (Fig. 1).

The input to our algorithm is a set of images of a sports category, only a handful of which need to be labeled, and news articles that mention this sport. We use kernelized canonical correlation analysis (kCCA) to learn a mapping between textual words and visual words (clusters of precomputed image segments). kCCA maps both word types into a low dimensional meaning space. If mappings are close in this space, the words are likely to be instances of the same underlying semantic concept. The learned mapping is then used to annotate new images and label all its segmented regions. With this model only very little human labeling is needed to automatically annotate and retrieve images and specific object regions in photo collections.

Related Work. The model is based on the probabilistic interpretation of canonical correlation analysis [12] by Bach et al. [1] and therefore in a line of work that relates items in different domains. Hardoon et al. [10] use kCCA to retrieve images given a multiple word text query and without using any labels around the retrieved images. Though this work is similar in that it connects visual and textual modalities, it is different in several aspects. Foremost, it uses a parallel image and text corpus and it provides only approximate global image labels without a real understanding of objects and their spatial relationships. Recently, [3] used kCCA to do unsupervised clustering of images and text in latent meaning space. Again, there is no object level understanding of images and the used corpora were aligned. A fascinating cross-modality application of CCA is presented in [14], which uses CCA to relate pixels that are connected to different sounds in short videos.

The method of [9] uses standard probabilistic CCA in an EM algorithm to iteratively learn a translation between two natural languages such as Spanish and English. The E-step finds word pairs that are likely to be 1-to-1 translations of each other and the M-step uses CCA to determine the probability of these words being actual translations. We adapt this idea to the image-text setting by (i) describing image segments as visual words, (ii) allowing n-to-1 mappings in the model and inference to account for many visual words being mapped to the same textual word and (iii) introducing a domain specific adjective kernel for both modalities. Other differences to this approach arise during inference due to the kernelization, approximations and the challenging setting of mapping an unrestricted number of image segments to words.

There is a myriad of different approaches for image annotation and segmentation which we will not explain in detail here [2, 20, 8, 19, 16].

Contributions. The proposed model is the first to *learn segmentation and annotation with words from large unaligned text corpora*. Inspired by ideas from language translation we first develop a new discrete image representation called *visual words*. Each visual word is a concatenation of single feature clusters. We translate them to textual words via a latent meaning space. Both types of words are mapped to this space by kernelized Canonical Correlation Analysis, making this the first usage of kCCA in segmentation. To improve the mapping, we introduce an *adjective kernel* that uses visually observable similarities between segments and distributions of co-occurring adjectives in text. Matlab code of the learning algorithm can be downloaded at www.socher.org.

2. Generative Model for Region-Text Translations

We propose a model that recognizes and localizes objects in scene images using semantically related but unaligned text corpora and a handful of training images. Fig. 2 (left) shows the graphical model representation of our kCCA-type model. The main idea is that there is a latent item from which we can sample the feature vectors in the visual and textual modalities.

The model inputs are (i) a set of discrete visual words with features that describe them contextually and visually and (ii) a set of words from a natural language corpus and their context and adjective features. We first introduce these representations before we describe the full generative process.

2.1. Words and Features

In order to obtain discrete visual words that represent large parts of objects, we segment all images using the segmentation algorithm of Felzenszwalb [7]. Next, we extract four types of visual features (see [22]) from these segments: **Color** features are simple RGB histograms, **texture** features are the mean responses of filterbanks in each segment. **Position** is described as the location in an 8×8 grid and **shape** is a binary histogram over the centered segment mask downscaled to 32×32. Each of these feature spaces is clustered separately with k-means. A visual word is defined as a unique sequence of feature cluster assignments and is represented as the following string:

$$C_{\texttt{color}} - C_{\texttt{position}} - C_{\texttt{texture}} - C_{\texttt{shape}} \qquad (1)$$

For all experiments we used $k = 40$ for color, $k = 8$ for position, $k = 20$ for shape and $k = 25$ for texture. For a throrough analysis of this representation see Sec. 4.2.

Context features for these words are normalized co-occurrence counts of neighboring segments' words. In order to relate the resulting visual words, we use the original feature vectors of the corresponding centers. Sec. 4.3 describes the used similarity kernel in detail.

Text features are co-occurrence counts of words, collected over the entire corpus (i.e. ignoring document boundaries).

2.2. The Generative Process

Words are described by a feature vector in their respective domains. Let $V = (v_1, \ldots, v_n)$ denote the set of visual

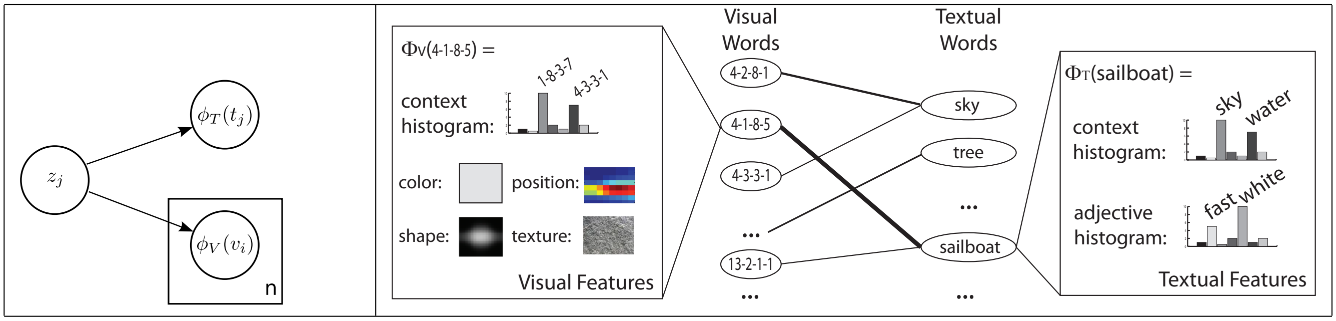


Figure 2. **Left**: Graphical model of the generative process. Nodes are random variables and arrows indicate conditional dependencies. Plates denote repetitions. The latent concept z connects one observed feature value in the textual domain with several in the visual domain. **Right**: The goal of the method is to find the optimal mapping M in the above weighted bipartite graph $B = (V, T, M, C)$ with elliptic nodes denoting words in the corresponding visual (V) and text (T) sets. C is the cost associated with each edge in M, alluded to by different edge widths. Next to the graph, both domain features are illustrated. Histograms depict normalized co-occurrence counts of nouns and adjectives in the text domain. Additionally, each visual word has associated with it a set of three visual features of different dimensionalities: color, position and shape. See text for further details.

word feature vectors: $v_i \in \mathbb{R}^{d_V}$ for all i. Furthermore, there are m textual words with the corresponding set of feature vectors: $T = (t_1, \dots, t_m)$, we define $T \in \mathbb{R}^{d_T \times m}$. Given these two sets, we want to find an n-to-1 mapping M which translates n words in the visual domain to 1 textual word.

We can describe these sets and the mappings between them as forming a weighted bipartite graph $B = (V, T, M, C)$. Edges are formed between translated words. C is the cost associated with each edge and can be interpreted as the inverse of the probability that is learned with the model.

The full generative process is as follows:

1. Sample an n-to-1 mapping $M \sim$ Matching-Prior
2. For each matched edge $(i, j) \in M$, where $i = 1, \dots, |V|$ and $j = 1, \dots, |T|$:
   (a) Sample latent concept: $z_j \sim \mathcal{N}(0, I_d)$
   (b) Sample visual features: $\phi_V(v_i) \sim \mathcal{N}(W_V z_j + \mu_V, \Psi_V)$
   (c) Sample textual features: $\phi_T(t_j) \sim \mathcal{N}(W_T z_j + \mu_T, \Psi_T)$
3. For each unmatched visual word v_i
   - Sample visual features from background distribution: $\phi_V(v_i) \sim \mathcal{N}(0, \sigma^2 I_{d_V})$
4. For each unmatched word t_j
   - Sample textual features from background distribution: $\phi_T(t_j) \sim \mathcal{N}(0, \sigma^2 I_{d_T})$

First, a mapping M is sampled from a prior over mappings in which each visual word indexed by v can occur only once, but each text word may be used in multiple mapping pairs. Changing the prior to allow for n-to-1 mappings is an important distinction to previous methods in language translation [9] which only used priors over 1-to-1 matchings. We assume a uniform prior over such mappings.

At the core of the generative model is canonical correlation analysis in its probabilistic interpretation introduced by Bach [1]. A latent concept $z_j \in \mathbb{R}^d$ is sampled, where $\min\{d_V, d_T\} \geq d \geq 1$ and I_d is a $d \times d$ identity matrix. Intuitively, latent concepts are object categories which are language-independent and abstracted from observed instances. We assume that the mapping from the set of concepts to the set of text words is injective[1]. Given the latent concept, the feature vectors of the indexed words are sampled in their respective domains from a Gaussian distribution. The mean of this distribution is the result of projecting the latent concept into the corresponding feature space via the matrix $W_V \in \mathbb{R}^{d_V \times d}$ and adding the sample mean μ_V (similar for textual features). Sec. 4 describes the employed domain features. The covariance matrices Ψ_V and Ψ_T capture domain specific variations, but are ignored during inference. Unmatched words are assumed to be generated from a background distribution. If a mapping is reasonable, generating the participating words through a latent concept should yield higher likelihood than this background. Fig. 2 (left) shows the corresponding graphical model.

3. Inference

We learn the model described above with a hard EM algorithm that iteratively builds the weighted bipartite graph B. The inputs are similarity matrices of the visual and textual words. The output is the complete weighted graph in which edges indicate mappings between visual and textual words.

E-step: Approximate the posterior over all mappings by finding the optimal, weighted n-to-1 mapping M in the bipartite graph B using weights C.

M-step: Update the weights C between all possible pairs using kCCA trained on the elements that participate in

[1] Therefore, the number of concepts is upper-bounded by the number of text words.

the best k mappings before. The first iteration uses the visual-textual training pairs to compute the M-step.

3.1. M-step

The idea of the M-step is to maximize the likelihood of the mapping pairs:

$$\max_{\xi} \sum_{(i,j)\in M} \log p(v_i, t_j, M_{ij}; \xi), \tag{2}$$

where $\xi = (W_V, \Psi_V, W_T, \Psi_T)$. As [1] showed, this is equal to maximizing the likelihood of the probabilistic interpretation of CCA. We will now shortly revise the intuition and inference of original CCA and its kernel extension. We use the original inference equations for kCCA instead of the maximum likelihood estimator because - as pointed out by [3] - the latter includes the auto-covariance matrices and is therefore more susceptible to noise.

Canonical Correlation Analysis Given two sets of samples such as V and T above, CCA seeks to find basis vectors which (i) project the elements of these sets into the same space and (ii) maximize the correlation of the projected vectors. This can be formulated as

$$\max_{w_v, w_t} \frac{\hat{E}[\langle v, w_v\rangle\langle t, w_t\rangle]}{\sqrt{\hat{E}[\langle v, w_v\rangle^2]\hat{E}[\langle t, w_t\rangle^2]}}, \tag{3}$$

where we use the empirical expectation of the function f over the number of samples N: $\hat{E}(f(x)) = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$. We assumed that the variables have been normalized to zero mean. Hardoon et al. [10] gave a detailed derivation of how this maximization can be cast into a standard eigenvalue problem.

Kernelized Canonical Correlation Analysis Because CCA can only recover linear relationships, it is useful to kernelize it by first projecting the data into a higher-dimensional feature space and then running CCA in this new space. To this end, we define kernels over V and T: $K_V(v_i, v_j) = \langle \phi_V(v_i), \phi_V(v_j)\rangle$ and similarly for T. The idea for kCCA is to search for solutions of w_v, w_t that lie in the span of $\phi_V(v)$, i.e. $w_v = \sum_i \alpha_i \phi_V(v_i)$ and $\phi_T(t)$ accordingly. With these kernels, we wish to optimize over:

$$\max_{w_v, w_t} \frac{w_v^T K_V K_T w_t}{\sqrt{w_v^T K_V^2 w_v \cdot w_t^T K_T^2 w_t}}. \tag{4}$$

As shown by Bach [1] and Hardoon [10], learning needs to be regularized in order to avoid trivial solutions. Hence, we penalize the norms of the projection vectors. We obtain the following standard eigenvalue problem:

$$(K_V + \kappa I)^{-1} K_T (K_T + \kappa I)^{-1} K_V w_v = \lambda^2 w_v. \tag{5}$$

The output of learning are the matrices w_v, w_t which map any vector in V and T to the latent space.

Computational Issues. In order to avoid degeneracy and problems with non-invertible Gram matrices and to increase computational efficiency, we approximate the Gram matrices using the Partial Gram-Schmidt Orthogonalization (PGSO) as suggested by Hardoon [10]. Compared to the Incomplete Cholesky Decomposition used in [1] and [9], PGSO does not depend on a permutation matrix P. With this method we represent the projections with reduced dimensionality. During all following experiments, the regularization parameter of kCCA κ is set to 5 and the parameter controlling the precision of the Gram-Schmidt process η is set to 0.2.

3.2. E-step

The goal of the E-step is to calculate the expected value over all the possible mapping pairs in the bipartite graph which is intractable. We therefore revert to hard EM and approximate only the best k such mapping pairs given the current parameter estimates ξ from kCCA:

$$M_{\texttt{new}} = \underset{M_{1:k}}{\texttt{argmax}} \log p(V, T, M; \xi). \tag{6}$$

We approximate this maximization problem by mapping the vectors in V and T to the latent space and using Euclidean distance between the projected vectors as a similarity score. This distance, which can be interpreted as an approximation to the matching probability, defines the edge weight (or cost) in the bipartite graph B for all possible pairs (i, j).

$$c_{i,j} = \|\phi_V(v_i)w_v - \phi_T(t_j)w_t\|_2. \tag{7}$$

This yields a new cost matrix C for the bipartite graph. We now extract the best possible k mapping pairs[2], add those to the kCCA training set and re-run the M-step. Iteratively enlarging the training set can be seen as a bootstrapping approach.

There are several possibilities to obtain the best n-to-1 mappings in B. First, it could be cast a minimum cost, maximum flow problem of a weighted bipartite graph and solved as a Linear Program[3]. However, given the size of the graph (V can be over 10,000 elements), this is not computationally feasible and we therefore use a greedy approximation. A simple first attempt is to use all pairs of lowest

[2] For all experiments we set $k := 1/10N$, where N is the total number of visual words.

[3] The LP has the form

$$min \sum_{i,j} c_{ij} x_{ij}$$

$$\texttt{subject to :} \sum_j x_{ij} = 1 \quad \forall i \in V \texttt{ and } \sum_i x_{ij} \leq n \quad \forall j \in T,$$

where we added a source node of capacity one to each visual node and a sink node to all the textual nodes with an upper-bounded capacity of n. C is the cost matrix and X is the flow from the visual nodes in V to the textual nodes in T. The last constraint allows each textual node to receive the flow from at most n visual nodes.

weight where each visual word occurs only once and textual words may occur multiple times. This, however, tends to degenerate the learning because it favors those words heavily which occurred frequently in the initial training set.

In order to solve this problem we enforce that the newly added k words have the same label ratio as last iterations' set of training words plus a smoothing term to allow for unseen labels. Let L be the total number of possible textual labels and let N_l be the number of associated visual words that appeared in the last iterations' training images ($l = 1, \ldots, L$). At the next iteration we map at most $k\frac{N_l+\delta}{\sum_{i=1}^{L} N_i}$ new visual words to object category l. We set $\delta = 2$ for all experiments.

4. Experiments

This section first describes the data set and demonstrates the flexibility of the visual word representation for object recognition. Then the context and adjective kernels and quantitative results for the two major tasks of annotation and segmentation are given. Furthermore, we analyze the method's behavior under several settings.

4.1. Data Set

We use the image dataset introduced and evaluated in [16] which consists of 8 sports categories: *badminton, bocce, croquet, polo, rock climbing, rowing, sailing* and *snowboarding*. Approximately 800 images were obtained by searching the online photo sharing website flickr.com with these 8 names. As a text corpus we use all articles from the New York Times [6] which mention one of the names of the above 8 sports categories. Unfortunately, *bocce, polo, croquet* and *rock climbing* do not have enough articles and can therefore not be used in the subsequent evaluation.

We use the tree tagger [21] to extract part-of-speech tags and obtain word stems. Words are labeled as nouns (or adjectives), if their most common tag is *noun* (or *adjective*). We then use the first name database Lexique [17] to map people's names to the object category *human*. We use Wordnet to find synonyms and map them to the same unique word. Lastly, we use Wordnet to filter out all nouns which are not *'physical entities'*.

4.2. Analysis of Visual Word Clustering

Visual words represent a cluster of image segments that share certain similarities. Due to the vast amount of variation in object appearance, no algorithm is capable of clustering all segments that depict the same category into one cluster purely based on the region's appearance.

Intuitively, we would like to obtain visual words that (i) only appear with one object category and that (ii) occur frequently in the corpus. These properties can be described by **purity**: visual words should be as pure as possible, i.e. they should depict mostly the same object category. Purity can be computed for different visual words v by:

$$\texttt{purity}(v) \triangleq \max_{c\in C} \frac{A(v,c)}{A(v,\cdot)}, \tag{8}$$

where $A(v,c)$ is defined as the area (sum of pixels) that has been assigned to the visual word v and the object category c and $A(v,\cdot)$ is the total area assigned to this visual word. Second, they should appear as frequently as possible, so robust context features can be extracted and there are less nodes in the bipartite graph. **Frequency** is defined as the number of times a specific region is assigned to a visual word. Certainly, both of these goals are orthogonal to each other and it is easy to obtain a clustering at both extremes. Pure but rare segments are easy to obtain by defining each segment as its own unique visual word. Assigning all segments to the same word results in the opposite.

Clustering image segments has been investigated in the past. [20] uses a topic model to partition the set of segments into visual object classes. Such an approach is computationally very expensive and does not provide the flexibility to modify the purity-frequency trade-off. For a thorough investigation of image segment clustering, we recommend the work of [22].

I.(a) Analysis of visual word clustering. A simple first approach to clustering regions into discrete words is to concatenate their features to one high dimensional vector, run k-means and assign each segment to its cluster. The number of words is then equal to the number of clusters. Fig. 3 (left) shows the purity and frequency results. While frequency is very high, the purity is too low to be useful.

Instead we use the method described in Sec. 2.1. The purity and frequency of the resulting visual words is largely dependent on the number of clusters in each feature space. This representation allows us to define the number of clusters for each feature. If we increase the number of clusters, then visual words will be cleaner but also appear less frequently. It is intuitive that having more distinctive colors is helpful, whereas 8 position clusters (which roughly correspond to 6 different y-ranges and left and right) are sufficient for the position feature.

I.(b) Analysis of visual word clustering. In order to show how flexible this representation is and how easily one can trade off purity and frequency for specific applications, we present 18,265 different combinations of the number of clusters in the four feature spaces. Fig. 3 (right) shows the statistics of frequency vs. purity for all combinations. We select a point in this space for which the average number of occurrences per visual word is 27 and purity is 80%. Lastly, this representation allows for a straightforward encoding of spatial relationships between visual words to learn constraints such as sky is always above water.

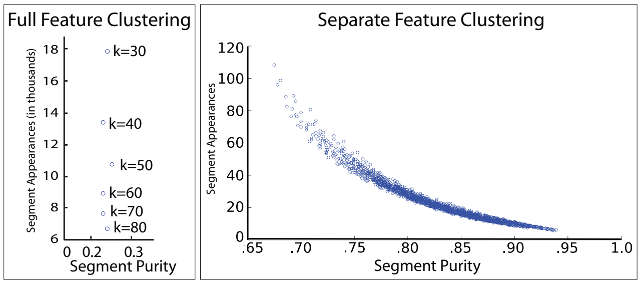


Figure 3. **I. Analysis of visual word clustering.** Each datapoint corresponds to the average statistics of a segment clustering. **Left: I.(a).** All segment features are concatenated and clustered into k clusters. The figure shows the number of times a word of that cluster appears vs how pure the corresponding labels of this cluster are. Notice that no such clustering provides very pure words. **Right: I.(b).** Results from using our method of clustering different feature types separately and concatenating them (see text for details). This region-based representation gives much flexibility in the trade-off between frequency and purity.

4.3. Kernels

Context Kernel For context we use the simple linear kernel of normalized word frequencies that occur in a window of size four around each word (where only nouns are counted). For visual words, we collect such context counts by counting adjacent visual words for each segment.

Adjective Kernel We also include adjective features. For text those are normalized frequencies of co-occurring adjectives. For visual words, we take the values of the corresponding cluster centers of the four feature types mentioned in Sec. 2.1 and compute a χ^2-kernel. Let x and y be one the feature vectors of two visual words[4], the χ^2 kernel normalized by A is computed by:

$$K(x,y) = \exp\left(-\frac{1}{2A}\sum_{i=1}^{d}\frac{(x_i-y_i)^2}{x_i+y_i}\right). \qquad (9)$$

For all experiments, we define the final visual kernel as the product: $K_V = K_{context}K_{color}K_{position}K_{texture}K_{shape}$.

4.4. Annotation

We compare the annotation performance of our model to three state-of-the-art approaches: alipr.com [15], CorrLDA [4] and Towards Total Scene Understanding [16]. We restrict labels to those words used in [16] and [4] in order to allow a direct comparison. This means we do not allow arbitrary words from the text corpus to be used as labels. We investigate model performance with and without this restriction in Sec. 4.6.

II. Annotation Comparison. Comparison to these methods is a difficult undertaking since the amount of training data varies between each of the methods. Alipr uses thousands of training images but also has a large space of textual labels. The model in [16] first uses hundreds of images for training 20 object models which bootstrap the full model and then leverages on about 5000 image-tag pairs from flickr. CorrLDA uses the same number of image-tag pairs without the bootstrap object images. Our method uses the least training data (20 images in total) and only a semantically related but unaligned text corpus. For obtaining visual words we used the same segment features as [16] but we do not use patches.

In the sports dataset of [16] there are 30 images in each of the four sports categories annotated with ground truth segmentations of 20 object categories. In this set of experiments we use only 5 randomly selected images from each sports category for a total of 20 training images. We train our model on all four sports together. The remaining 4×25 images are used for testing and comparing all the models. The tests of all methods were performed on the same 100 test images. Table 1 shows average precision, recall and F-measure for the different models. F-measure is the standard harmonic mean of precision and recall. Even though our model uses the least training data, it achieves a 5% increase in F-measure over the state-of-the-art of [16].

4.5. Segmentation

III. Segmentation Comparison. As mentioned above, our mapping of segments to visual words allows us to provide a pixel level segmentation unlike previously used methods that involved CCA. The experimental setup is equal to the above one in annotation. Precision is calculated by dividing the correctly segmented pixels by the total number of pixels assigned to each object category. Table 1 compares our segmentation method with two related state-of-the-art methods for semantic segmentation: Cao & Fei-Fei, 2007 [5] and Total Scene [16].

IV. Analysis of single category training. In another set of experiments (IV. and V.), we train and test the four sports categories separately. This easier task was motivated in the introduction: All you need to annotate and segment images from a sports event is the name of the sport and a handful of training images. Fig. 4 shows separate precision and recall values for each sport and both tasks when trained with 5 images per category. The last columns of table 1 list the results averaged over the 4 sports. Fig 6 shows several results from the test set.

Our model does not perform as well in segmentation as the other model when all categories are jointly trained. This is mostly due to the considerably smaller training set that does not provide sufficient information. As a result there is easy confusion between sky and snow, clouds and sky, court and water, etc. In these cases visual and contextual features are very similar and the model lacks a top-down scene influence to counter-balance the bottom up cues. In

[4] For instance, x could be the RGB histogram of the 4'th color center, if the visual word starts with '$C_{color} = 4$'

Annotation	Alipr			Corr LDA			Total Scene			Our Model			(Our Model - Exp. IV)		
Results	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Mean	.17	.25	.20	.17	.37	.23	.29	.76	.42	.35	.71	**.47**	.71	.79	.75
Segmentation				Cao,2007			Total Scene			Our Model			(Our Model - Exp. IV)		
Results				P	R	F	P	R	F	P	R	F	P	R	F
Mean				.35	.32	.33	.45	.43	**.44**	.30	.24	.27	.46	.52	.49

Table 1. **Top: II. Annotation Comparison.** Precision, recall and F-measure for Alipr, Corr-LDA, Total Scene and our model. All models except Alipr were jointly trained on four sports categories. However, our method uses two orders of magnitude less training images. **Bottom: III. Segmentation Comparison.** Results of segmentation averaged over all 20 objects. **Last column: IV. Analysis of single category training.** Average results when each sports category is trained and tested separately.

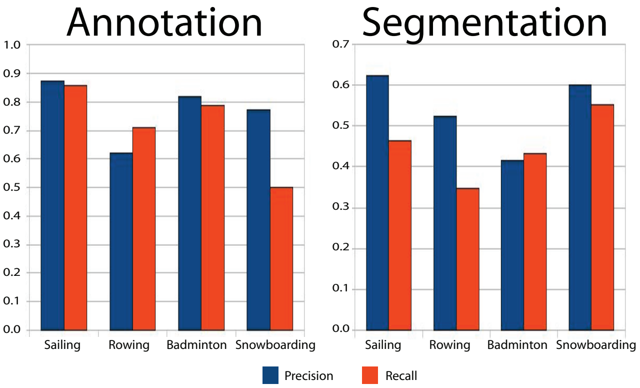


Figure 4. **IV. Analysis of single category training.** Annotation and segmentation results for four different sports categories each trained with only 5 labeled images.

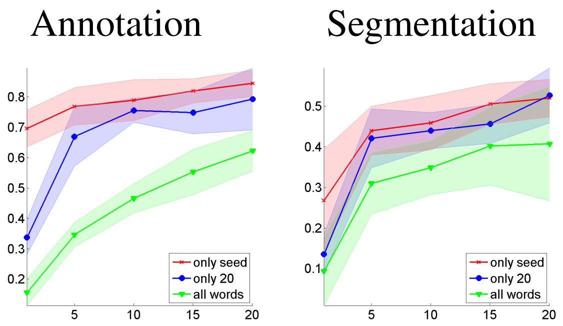


Figure 5. **V. Influence of the number of training images and possible text labels.** Average F-measures and standard deviation for different numbers of training images (x-axis) and different pools of textual words that may participate in the mapping. 5 sets of randomly chosen training images were used for each setting.

experimental setting IV. where more training data is given and less categories are possible, the model and our visual words perform very well in segmentation.

4.6. Evaluation of Different Settings

V. Influence of the number of training images and possible text labels. To provide some insight into model learning and to underline the effectiveness of the model in dealing with little training data, we provide experimental results by varying the following two settings: (i) number of training images $t = 1, 5, 10, 15, 20$ from which mapping seed sets are extracted. Results are then computed for the remaining $30 - t$ images and their visual words. (ii) pool of possible textual words: (**seed**) only words that appeared in training are used. (**20anno**) all 20 words that were used for evaluation in [16]. (**all**) any word from the text corpus of the corresponding sports category may participate in the mapping. Fig. 4 (bottom row) shows the resulting average F-measures. The more seed mappings are extracted, the better the three text settings perform. It is interesting to notice that with increasing numbers of seed mappings, the full text model slowly approaches the other more restricted models. The last row of Fig. 6 shows annotated images, if all corpus words are allowed. This is hard to evaluate but gives very interesting results such as the label *sea* for *water*.

5. Conclusion

We presented a model based on kCCA to segment and annotate images using a handful of labeled images and an unaligned text corpus. We leverage on contextual similarities between scene images and semantically related articles and introduce an adjective kernel. Our semi-supervised method requires very little training data to outperform other state-of-the-art methods in annotation underlining the usefulness of using related text corpora to learn relationships between objects. In settings with less labels (a single sports category), 5 training images are sufficient for good performance. Possible extensions of this work include the incorporation of scene information and geometric constraints (such as sky being above snow) to the model and using the new visual word representation for classifying objects without context.

Acknowledgments

We thank for helpful discussions, code, data or comments: Jordan Boyd-Graber, Garrett Marcotte, Aria Haghighi, Li Jia, Sean Gerrish, Tomasz Malisiewicz, Sam Gershman and the Stanford vision lab. This research is partially supported by the ERP fellowship and the Francis Robbins Upton fellowship to RS and the NSF CAREER grant (IIS-0845230), a Google research award, and a Microsoft Research fellowship to L.F-F.

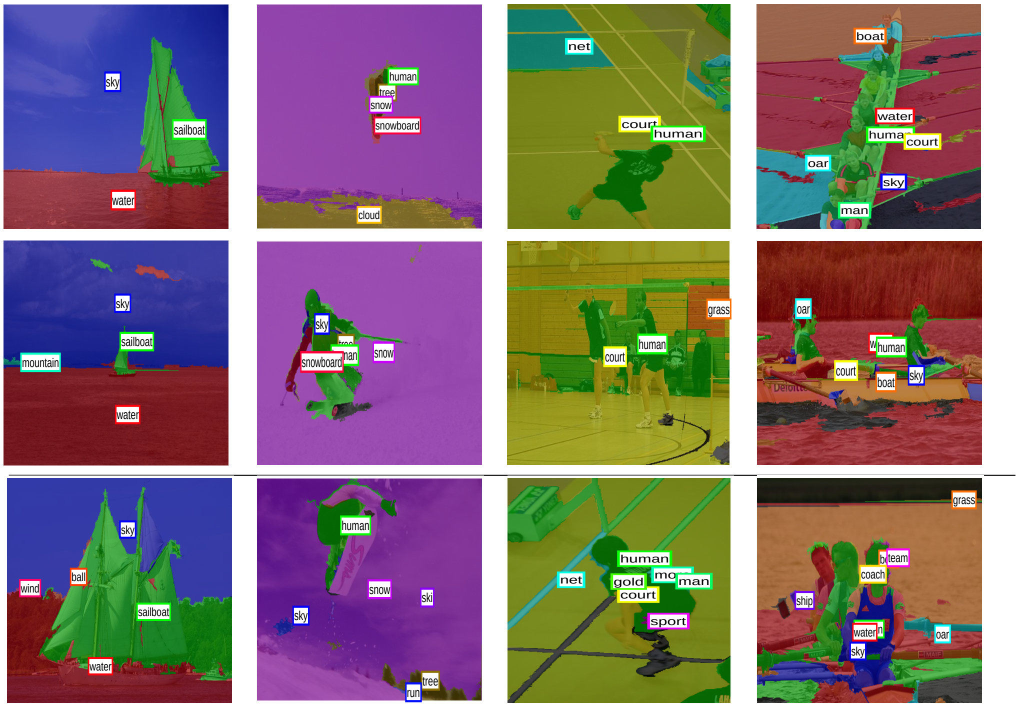


Figure 6. **Top two rows: IV. Analysis of single category training.** Results of annotation and segmentation of the test set. Labels are shown in boxes and the corresponding regions are overlayed with the same color as the boundary box. **Bottom row: V. Results with mappings from all words of the text corpus.** If all words of the text corpus are allowed in mappings the evaluation becomes very hard. *Man* might replace the *human* label in badminton images. *Wind* might show up in front of a *sailboat* etc.

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *JMLR*, 3, 2003. 2, 3, 4

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

[3] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *CVPR '08*, 2008. 2, 4

[4] D. Blei and M. Jordan. Modeling annotated data. *Proc. of ACM SIGIR*, 2003. 6

[5] L. Cao and L. Fei-Fei. Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. *ICCV 2007*. 6

[6] L. D. C. Evan Sandhaus. The new york times annotated corpus. Philadelphia. 5

[7] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 2004. 2

[8] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV 2008*. 2

[9] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. In *ACL-2008: HLT*. 2, 3, 4

[10] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12), Dec. 2004. 2, 4

[11] D. Hoiem, A. Efros, and M. Hebert. Putting Objects in Perspective. *CVPR*, 2006. 1

[12] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. 2

[13] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. *CVPR*, 2006. 1

[14] E. Kidron, Y. Y. Schechner, and M. Elad. Pixels that sound. In *CVPR 2005*. 2

[15] J. Li and J. Wang. Automatic Linguistic Indexing of Pictures by a statistical modeling approach. *PAMI*, 25(9):1075–1088, 2003. 6

[16] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR 2009*. 2, 5, 6, 7

[17] B. New and C. Pallier. Lexique 3. 5

[18] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences, in press*, Nov. 2007. 1

[19] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV 2007*. 1, 2

[20] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *CVPR 2006*. 2, 5

[21] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Sep. 1994. 5

[22] M. Tomasz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR 2008*. 2, 5

[23] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *ICCV '03*, page 273, 2003. 1