# Learning Visual Representations using Images with Captions

Ariadna Quattoni
MIT
CSAIL

Michael Collins
MIT
CSAIL

Trevor Darrell
MIT
CSAIL

## Abstract

*Current methods for learning visual categories work well when a large amount of labeled data is available, but can run into severe difficulties when the number of labeled examples is small. When labeled data is scarce it may be beneficial to use unlabeled data to learn an image representation that is low-dimensional, but nevertheless captures the information required to discriminate between image categories. This paper describes a method for learning representations from large quantities of unlabeled images which have associated captions; to improve learning in future image classification problems. Experiments show that the method significantly outperforms a fully-supervised baseline model as well as models that ignore the captions and learn a visual representation by performing PCA on the unlabeled images alone. Our current work concentrates on captions as the source of meta-data, but more generally other types of meta-data could be used (e.g., video sequences with accompanying speech).*

## 1. Introduction

Current methods for learning visual categories work well when a large amount of labeled data is available, but can run into severe difficulties when the number of labeled examples is small—for example when a user defines a new category and provides only a few labeled examples. Image representations are typically of high dimension, requiring relatively large amounts of training data. When labeled data is scarce it may be beneficial to use unlabeled data to learn an image representation that is low-dimensional, but nevertheless captures the information required to discriminate between image categories.

In some cases unlabeled data may contain useful meta-data that can be used to learn a low-dimensional representation that reflects the semantic content of an image. As one example, large quantities of images with associated natural language captions can be found on the web. This paper describes an algorithm that uses images with captions or other meta-data to derive an image representation that allows significantly improved learning in cases where only a few labeled examples are available.

More specifically, we propose to use the meta-data to induce a representation that reflects an underlying part structure in an existing, high-dimensional visual representation. The new representation groups together synonymous visual features—features that consistently play a similar role across different image classification tasks.

Our approach follows a structural learning framework (Ando Zhang [1]) and exploits *auxiliary problems* which can be created from images with associated captions. Each auxiliary problem involves taking an image as input, and predicting whether or not a particular content word (e.g, *man*, *official*, or *celebrates*) is in the caption associated with that image. In structural learning, a separate linear classifier is trained for each of the auxiliary problems; manifold learning (e.g., SVD) is then applied to the resulting set of parameter vectors, in essence finding a low-dimensional space which is a good approximation to the space of possible parameter vectors. If features in the high-dimensional space correspond to the same semantic part, their associated classifier parameters (weights) across different auxiliary problems may be correlated in such a way that the basis functions learned by the SVD step collapse these two features to a single feature in a new, low-dimensional feature-vector representation.

In a first set of experiments, we use synthetic data examples to illustrate how the method can uncover latent part structures. We then describe experiments on classification of news images into different topics. We compare a baseline model that uses a bag-of-words SIFT representation of image data to our method, which replaces the SIFT representation with a new representation that is learned from 8,000 images with associated captions. In addition, we compare our method to a baseline model that ignores the meta-data and learns a new visual representation by performing PCA on the unlabeled images. Note that our goal is to build classifiers that work on images alone (i.e., images which *do not* have captions), and our experimental set-up reflects this, in that training and test examples for the topic classification tasks include image data only. The experiments show that

our method significantly outperforms both baseline models. The new representation reduces the number of labeled examples required by a large margin: a model trained with just a single positive example using the new representation performs as well as models trained with between 8 and 16 positive examples for the baseline models; a new model trained with 4 positive examples performs as well as models trained with between 16 and 32 positive examples for the baseline models.

## 2. Previous work

When few labeled examples are available most current supervised learning methods [21, 9, 11, 17, 13] for image classification may work poorly. To reach human performance, it is clear that knowledge beyond the supervised training data needs to be leveraged.

There is a large literature on semi-supervised learning approaches, where unlabeled data is used in addition to labeled data. We do not aim to give a full overview of this work, for a comprehensive survey article see [16]. Most semi-supervised learning techniques can be broadly grouped into three categories depending on how they make use of the unlabeled data: density estimation, dimensionality reduction via manifold learning and function regularization. Generative models trained via EM can naturally incorporate unlabeled data for classification tasks [14, 2]. In the context of discriminative category learning, Fisher kernels [10] have been used to exploit a learned generative model of the data space in an SVM classifier.

Our work is related to work in *transfer* or *multi-task* learning, where training data in related tasks is used to aid learning in the problem of interest. Transfer and multi-task learning have a relatively long history in machine learning [19, 4, 15, 1]. Our work builds on the structure learning approach of Ando and Zhang [1], who describe an algorithm for transfer learning, and suggest the use of auxiliary problems on unlabeled data as a method for constructing related tasks. In vision a Bayesian transfer learning approach has been proposed for object recognition [7] where a common prior over visual classifier parameters is learnt, their results show a significant improvement when learning from a few labeled examples. In the context of multi-task learning, approaches that learn a shared part structure among different classes have also been proposed. In [20] Torralba introduced a discriminative (boosted) learning framework that learns common structure. The paper demonstrated faster learning with better generalization when parts are shared among classes. Epshtein and Ullman [5] have also addressed this goal, presenting an approach which identifies functional parts by virtue of shared context. To the best of our knowledge, no previous approach to learning parts in images has made use of meta-data and structure learning.

Several authors have considered the use of images with associated text data. Fergus et al. [8] developed a method using Google's image search to learn visual categories, and report results comparable to fully supervised paradigms. Other work that has made use of image and/or video caption data includes CMU's Infomedia system [6], Barnards's "Matching Words and Pictures" [3], Miller's and "Names and images in the news" [12].

## 3. Learning Visual Representations

A good choice of representation of images will be crucial to the success of any model for image classification. The central focus of this paper is a method for automatically learning a representation from images which are unlabeled, but which have associated meta-data, for example natural language captions. We are particularly interested in learning a representation that allows effective learning of image classifiers in situations where the number of training examples is small. The key to the approach is to use meta-data associated with the unlabeled images to form a set of auxiliary problems which drive the induction of an image representation. We assume the following scenario:

• We have labeled (supervised) data for some image classification task. We will call this the core task. For example, we might be interested in recovering images relevant to a particular topic in the news, in which case the labeled data would consist of images labeled with a binary distinction corresponding to whether or not they were relevant to the topic. We denote the labeled examples as the core set $(x_1, y_1), \ldots, (x_n, y_n)$ where $(x_i, y_i)$ is the $i$'th image/label pair. Note that test data points for the core task contain image data alone (these images do not have associated caption data, for example).

• We have $N$ *auxiliary* training sets, $\mathcal{T}_i = \{(x^i_1, y^i_1), \ldots, (x^i_{n_i}, y^i_{n_i})\}$ for $i = 1 \ldots N$. Here $x^i_j$ is the $j$'th image in the $i$'th auxiliary training set, $y^i_j$ is the label for that image, and $n_i$ is the number of examples in the $i$'th training set. The auxiliary training sets consist of binary classification problems, distinct from the core task, where each $y^i_j$ is in $\{-1, +1\}$. Shortly we will describe a method for constructing auxiliary training sets using images with captions.

• The aim is to learn a representation of images, i.e., a function that maps images $x$ to feature vectors $\mathbf{f}(x)$. The auxiliary training sets will be used as a source of information in learning this representation. The new representation will be applied when learning a classification model for the core task.

In the next section we will describe a method for inducing a representation from a set of auxiliary training sets. The intuition behind the method is to find a representation which is relatively simple (i.e., of low dimension), yet allows strong performance on the auxiliary training sets. If

the auxiliary tasks are sufficiently related to the core task, the learned representation will allow effective learning on the core task, even in cases where the number of training examples is small.

A central question is how auxiliary training sets can be created for image data. A key contribution of this paper is to show that unlabeled images which have associated text captions can be used to create auxiliary training sets, and that the representations learned with these unlabeled examples can significantly reduce the amount of training data required for a broad class of topic-classification problems. Note that in many cases, images with captions are readily available, and thus the set of captioned images available may be considerably larger than our set of labeled images.

Formally, denote a set of images with associated captions as $(x'_1, c_1), \ldots, (x'_m, c_m)$ where $(x'_i, c_i)$ is the $i$'th image/caption pair. We base our $N$ auxiliary training sets on $N$ content words, $(w_1, \ldots, w_N)$. A natural choice for these words would be to choose the $N$ most frequent content words seen within the captions.[1] $N$ auxiliary training sets can then be created as follows. Define $I_i[c]$ to be 1 if word $w_i$ is seen in caption $c$, and $-1$ otherwise. Create a training set $\mathcal{T}_i = \{(x'_1, I_i[c_1]), \ldots, (x'_m, I_i[c_m])\}$ for each $i = 1 \ldots N$. Thus the $i$'th training set corresponds to the binary classification task of predicting whether or not the word $w_i$ is seen in the caption for an image $x'$.

### 3.1. Learning Visual Representations from Auxiliary Tasks

This section describes an algorithm for learning a representation from a set of auxiliary training sets. We adopt the framework described in [1]. We assume that a *baseline* representation of images $\mathbf{g}(x) \in \mathbb{R}^d$ is available. In the experiments in this paper $\mathbf{g}(x)$ is a SIFT histogram representation [18]. In general, $\mathbf{g}(x)$ will be a "raw" representation of images that would be sufficient for learning an effective classifier with a large number of training examples, but which performs relatively poorly when the number of training examples is small. For example, with the SIFT representation the feature vectors $\mathbf{g}(x)$ are of relatively high dimension (we use $d = 1,000$), making learning with small amounts of training data a challenging problem without additional information.

Note that one method for learning a representation from the unlabeled data would be to use PCA—or some other density estimation method—over the feature vectors $\mathbf{g}(x_1), \ldots, \mathbf{g}(x_m)$ for the set of unlabeled images (we will call this method the *data-PCA* method). The method we describe differs significantly from PCA and similar methods, in its use of meta-data associated with the images, for example captions. Later we will describe synthetic experiments where PCA fails to find a useful representation, but our method is successful. In addition we describe experiments on real image data where PCA again fails, but our method is successful in recovering representations which significantly speed learning.

[1] In our experiments we define a content word to be any word which does not appear on a "stop list" of common function words in English.

**Input 1:** Auxiliary training sets $\{(x^i_1, y^i_1), \ldots, (x^i_{n_i}, y^i_{n_i})\}$ for $i = 1 \ldots N$. Here $x^i_j$ is the $j$'th image in the $i$'th training set, $y^i_j$ is the label for that image. $n_i$ is the number of examples in the $i$'th training set. We consider binary classification problems, where each $y^i_j$ is in $\{-1, +1\}$. Each image $x$ is represented by a feature vector $\mathbf{g}(x) \in \mathbb{R}^d$.
**Input 2:** Core training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$

**Structural learning using auxiliary training sets:**

**Step 1: Train N linear classifiers.** For $i = 1 \ldots N$, choose the optimal parameters on the $i$'th training set to be $\mathbf{w}^*_i = \arg\min_{\mathbf{w}} L_i(\mathbf{w})$ where

$$L_i(\mathbf{w}) = \sum_{j=1}^{n_i} l(\mathbf{w} \cdot \mathbf{g}(x^i_j), y^i_j) + \frac{C}{2}||\mathbf{w}||^2$$

(See section 3.1 for more discussion.)

**Step 2: Perform SVD on the Parameter Vectors.**
Form a matrix $\mathbf{W}$ of dimension $d \times N$, by taking the parameter vectors $\mathbf{w}^*_i$ for $i = 1 \ldots N$. Compute a projection matrix $\mathbf{A}$ of dimension $h \times d$ by taking the first $h$ eigenvectors of $\mathbf{W}\mathbf{W}'$.
**Output:** The projection matrix $\mathbf{A} \in \mathbb{R}^{h \times d}$.

**Train using the core training set:**

Define $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$.
Choose the optimal parameters on the core training set to be $\mathbf{v}^* = \arg\min_{\mathbf{v}} L(\mathbf{v})$ where

$$L(\mathbf{v}) = \sum_{j=1}^{n} l(\mathbf{v} \cdot \mathbf{f}(x_j), y_j) + \frac{C}{2}||\mathbf{v}||^2$$

Figure 1. The structural learning algorithm.

Given the baseline representation, the new representation is defined as $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$ where $\mathbf{A}$ is a projection matrix of dimension $h \times d$.[2] The value of $h$ is typically chosen such that $h \ll d$. The projection matrix is learned from the set of auxiliary training sets, using the *structural learning* approach described in [1]. Figure 1 shows the algorithm.

In a first step, linear classifiers $\mathbf{w}^*_i$ are trained for each of

[2] Note that the restriction to linear projections is not necessarily limiting. It is possible to learn non-linear projections using the kernel trick; i.e., by expanding feature vectors $\mathbf{g}(x)$ to a higher-dimensional space, then taking projections of this space.

the $N$ auxiliary problems. In several parameter estimation methods, including logistic regression and support vector machines, the optimal parameters $\mathbf{w}^*$ are taken to be $\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w})$ where $L(\mathbf{w})$ takes the following form:

$$L(\mathbf{w}) = \sum_{j=1}^{n} l(\mathbf{w} \cdot \mathbf{g}(x_j), y_j) + \frac{C}{2}||\mathbf{w}||^2 \qquad (1)$$

Here $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ is a set of training examples, where each $x_j$ is an image and each $y_j$ is a label. The constant $C > 0$ dictates the amount of regularization in the model. The function $l(\mathbf{w} \cdot \mathbf{g}(x_j), y_j)$ is some measure of the loss for the parameters $\mathbf{w}$ on the example $(x_j, y_j)$. For example, in support vector machines $l$ is the hinge-loss, defined as $l(m, y) = (1 - ym)_+$ where $(z)_+$ is $z$ if $z >= 0$, and is 0 otherwise. In logistic regression the loss function is

$$l(m, y) = -\log \frac{\exp\{ym\}}{1 + \exp\{ym\}}. \qquad (2)$$

Throughout this paper we use the loss function in Eq. 2, and classify examples with $sign(\mathbf{w} \cdot \mathbf{g}(x))$ where $sign(z)$ is 1 if $z \geq 0$, $-1$ otherwise.

In the second step, SVD is used to identify a matrix $\mathbf{A}$ of dimension $h \times d$. The matrix defines a linear subspace of dimension $h$ which is a good approximation to the space of induced weight vectors $\mathbf{w}_1^*, \ldots, \mathbf{w}_N^*$. Thus the approach amounts to *manifold learning in classifier weight space*. Note that there is a crucial difference between this approach and the data-PCA approach: in data-PCA SVD is run over the data space, whereas in this approach SVD is run over the space of parameter values. This leads to very different behavior of the two methods.

Ando and Zhang [1] describe the following method that makes use of the projection matrix $\mathbf{A}$ when training a model for a new problem. The parameter values are chosen to be $\mathbf{w}^* = \mathbf{A}'\mathbf{v}^*$ where $\mathbf{v}^* = \arg\min_{\mathbf{v}} L(\mathbf{v})$ and

$$L(\mathbf{v}) = \sum_{j=1}^{n} l((\mathbf{A}'\mathbf{v}) \cdot \mathbf{g}(x_j), y_j) + \frac{C}{2}||\mathbf{v}||^2 \qquad (3)$$

This essentially corresponds to constraining the parameter vector $\mathbf{w}^*$ for the new problem to lie in the sub-space defined by $\mathbf{A}$. Hence we have effectively used the auxiliary training problems to learn a sub-space constraint on the set of possible parameter vectors.

If we define $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$, it is simple to verify that

$$L(\mathbf{v}) = \sum_{j=1}^{n} l(\mathbf{v} \cdot \mathbf{f}(x_j), y_j) + \frac{C}{2}||\mathbf{v}||^2 \qquad (4)$$

and also that $sign(\mathbf{w}^* \cdot \mathbf{g}(x)) = sign(\mathbf{v}^* \cdot \mathbf{f}(x))$. Hence an alternative view of the algorithm in figure 1 is that it induces a new representation $\mathbf{f}(x)$. In summary, the algorithm in figure 1 derives a matrix $\mathbf{A}$ that can be interpreted either as a sub-space constraint on the space of possible parameter vectors, or as defining a new representation $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$.

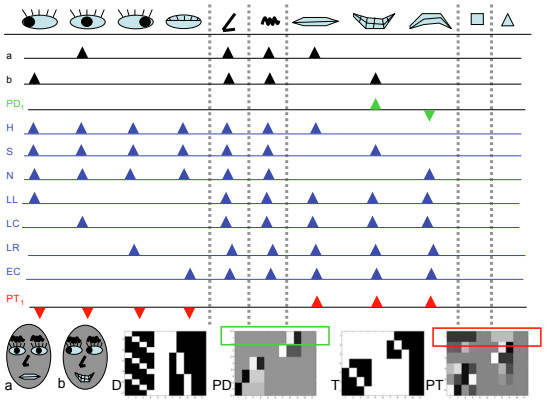


Figure 2. Concept figure illustrating how when appropriate auxiliary tasks have already been learned manifold learning in classifier weight space can group features corresponding to functionally defined visual parts. Parts (eyes, nose, mouth) of an object (face) may have distinct visual appearances (the top row of cartoon part appearances). A specific face (e.g., a or b) is represented with the boolean indicator vector as shown. Matrix D shows all possible faces given this simple model; PCA on D is shown row-wise in PD (first principal component is shown also above in green as $PD_1$.) No basis in PD groups together eyes or mouth appearances; different part appearances never co-occur in D. However, idealized classifiers trained to recognize, e.g., faces with a particular mouth and any eye (H,S,N), or a particular eye and mouth (LL,LC,LR,EC), will learn to group features into parts. Matrix T and blue vectors above show these idealized boolean classifier weights; the first principal component of T is shown in red as $PT_1$, clearly grouping together the four cartoon eye and the three cartoon mouth appearances. Positive and negative components of $PT_1$ would be very useful features for future learning tasks related to faces in this simple domain because they group together different appearances of eyes and mouths.

## 4. Examples Illustrating the Approach

Figure 2 shows a concept figure illustrating how PCA in a classifier weight space can discover functional part structures given idealized auxiliary tasks. When the tasks are defined such that to solve them they need to learn to group different visual appearances, the distinct part appearances will then become correlated in the weight space, and techniques such as PCA will be able to discover them. In practice the ability to obtain such ideal classifiers is critical to our method's success. Next we will describe a synthetic example where the method is successful; in the following section we present real-world examples where auxiliary tasks are readily available and yield features that speed learning of future tasks.

We now describe experiments on synthetic data that il-

(a)

| | | | | |
|---|---|---|---|---|
| **a** | A | *A* | A | **A** |
| *b* | ***b*** | ***B*** | **b** | *b* |
| **c** | C | *c* | c | ***c*** |
| | | ... | | |
| **j** | J | *J* | J | **J** |

(b)

| | |
|---|---|
| A *b* **c** | *a b* D |
| A **b** *c* | **A** *b* D |
| **a** ***b*** c | a b d |
| **A d** E | *b c* f |
| A D E | ***B*** c *f* |
| **A** D *e* | *b* C *f* |

Figure 3. Synthetic data involving objects constructed from letters. (a) There are 10 possible parts, corresponding to the first 10 letters of the alphabet. Each part has 5 possible observations (corresponding to different fonts). (b) Each object consists of 3 distinct parts; the observation for each part is drawn uniformly at random from the set of possible observations for that part. A few random draws for 4 different objects are shown.

lustrate the approach. To generate the data, we assume that there is a set of 10 possible *parts*. Each *object* in our data consists of 3 distinct parts; hence there are $\binom{10}{3} = 120$ possible objects. Finally, each of the 10 parts has 5 possible *observations*, giving 50 possible observations in total (the observations for each part are distinct).

As a simple example (see figure 3), the 10 parts might correspond to 10 letters of the alphabet. Each "object" then consists of 3 distinct letters from this set. The 5 possible observations for each part (letter) correspond to visually distinct realizations of that letter; for example, these could correspond to the same letter in different fonts, or the same letter with different degrees of rotation. The assumption is that each observation will end up as a distinct visual word, and therefore that there are 50 possible visual words.

The goal in learning a representation for object recognition in this task would be to learn that different observations from the same part are essentially equivalent—for example, that observations of the letter "a" in different fonts should be collapsed to the same point. This can be achieved by learning a projection matrix $\mathbf{A}$ of dimension $10 \times 50$ which correctly maps the 50-dimensional observation space to the 10-dimensional part space. We show that the use of auxiliary training sets, as described in section 3.1, is successful in learning this structure, whereas PCA fails to find any useful structure in this domain.

To generate the synthetic data, we sample 100 instances of each of the 120 objects as follows. For a given object $y$, define $P_y$ to be the set of parts that make up that object. For each part $p \in P_y$, generate a single observation uniformly at random from the set of possible observations for $p$. Each data point generated in this way consists of an object label $y$, together with a set of three observations, $x$. We can represent $x$ by a 50-dimensional binary feature vector $\mathbf{g}(x)$, where only 3 dimensions (corresponding to the three observations in $x$) are non-zero.

To apply the auxiliary data approach, we create 120 auxiliary training sets. The $i$'th training set corresponds to the problem of discriminating between the $i$'th object and all other 119 objects. A projection matrix $\mathbf{A}$ is learned from the auxiliary training sets. In addition, we can also construct a projection matrix using PCA on the data points $\mathbf{g}(x)$ alone. Figures 4 and 5 show the projections learned by PCA and the auxiliary tasks method. PCA fails to learn useful structure; in contrast the auxiliary task method correctly collapses observations for the same part to nearby points.

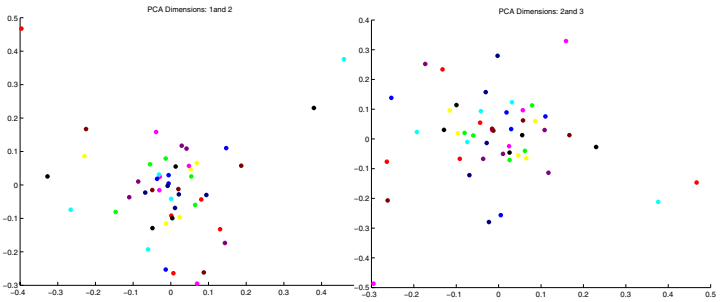


Figure 4. The representations learned by PCA on the synthetic data problem. The first figure shows projections 1 vs. 2; the second figure shows projections 2 vs. 3. Each plot shows 50 points corresponding to the 50 observations in the model; observations corresponding to the same part have the same color. There is no discernable structure in the figures. The remaining dimensions were found to similarly show no structure.

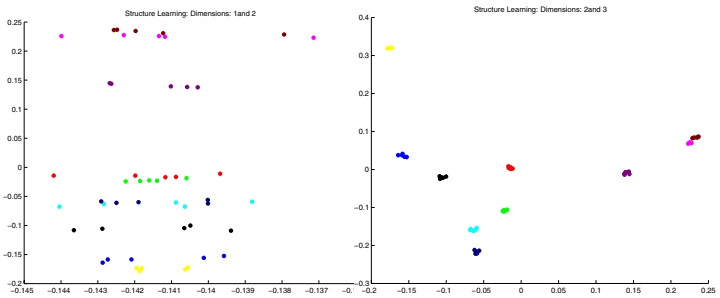


Figure 5. The representations learned by structural learning on the synthetic data problem. The first figure shows projections 1 vs. 2; the second figure shows projections 2 vs. 3. Each plot shows 50 points corresponding to the 50 observations in the model; observations corresponding to the same part have the same color. There is clear structure in features 2 and 3, in that observations corresponding to the same part are collapsed to nearby points in the projected space. The remaining dimensions were found to show similar structure to those in dimensions 2 and 3.

## 5. Experiments on Images with Captions

### 5.1. Data

We collected a data set consisting of 10,576 images. These images were collected from the Reuters news website (`http://today.reuters.com/news/`) during a period of one week. Images on the Reuters website are partitioned into *stories* or *topics*, which correspond to different

Figure 6. Example images from the Golden Globes, figure skating, Grammy and ice hockey, topics.

topics in the news. Thus each image has a topic label—in our case, the images fell into 130 possible topics. Figure 6 shows some example images.

The experiments involved predicting the topic variable $y$ for test images. We reserved $8,000$ images as a source of training data, and an additional $1,000$ images as a potential source of development data. The remaining 1,576 images were used as a test set. Multiple training sets of different sizes, and for different topics, were created as follows. We created training sets $T_{n,y}$ for $n = \{1, 2, 4, 8, 16, 32, 64\}$ and $y = \{1, 2, 3, \ldots, 15\}$, where $T_{n,y}$ denotes a training set for topic $y$ which has $n$ positive examples from topic $y$, and $4n$ negative examples. The 15 topics corresponded to the 15 most frequent topics in the training data. The positive and negative examples were drawn randomly from the training set of size $8,000$. We will compare various models by training them on each of the training sets $T_{n,y}$, and evaluating the models on the 1,576 test images.

In addition, each of the $8,000$ training images had associated captions, which can be used to derive an image representation (see section 3.1). Note that we make no use of captions on the test or development data sets. Instead, we will use the $8,000$ training images to derive representations that are input to a classifier that uses images alone. In summary, our experimental set-up corresponds to a scenario where we have a small amount of labeled data for a core task (predicting the topic for an image), and a large amount of unlabeled data with associated captions.

### 5.2. The Baseline Model

A baseline model was trained on all training sets $T_{n,y}$. In each case the resulting model was tested on the $1,576$ test examples. The baseline model consists of a logistic regression model over the SIFT features: to train the model we used conjugate gradient descent to find the parameters $\mathbf{w}^*$ which maximize the regularized log-likelihood, see equations 1 and 2. When calculating equal-error-rate statistics on test data, the value for $P(y = +1|x; \mathbf{w}^*)$ can be calculated for each test image $x$; this score is then used to rank the test examples.

The parameter $C$ in Eq. 1 dictates the amount of regularization used in the model. For the baseline model, we used the development set of $1,000$ examples to optimize the value of $C$ for each training set $T_{n,y}$. Note that this will in practice give an upper bound on the performance of the baseline model, as assuming $1,000$ development examples is almost certainly unrealistic (particulary considering that we are considering training sets whose size is at most 320). The values of $C$ that were tested were $10^k$, for $k = -5, -4, \ldots, 4$.

### 5.3. The Data-PCA Model

As an additional baseline, we again trained a logistic-regression classifier, but with the original feature vectors $\mathbf{g}(x)$ in training and test data replaced by $h$-dimensional feature vectors $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$ where $\mathbf{A}$ was derived using PCA. A matrix $\mathbf{F}$ of dimension $1,000 \times 8,000$ was formed by taking the feature vectors $\mathbf{g}(x)$ for the $8,000$ data points; the projection matrix $\mathbf{A}$ was constructed from the first $h$ eigenvectors of $\mathbf{F}\mathbf{F}'$. The PCA model has free parameters $h$ and $C$. These were optimized using the method described in section 5.5. We call this model the *data-PCA* model.

### 5.4. A Model with Predictive Structure

We ran experiments using the structure prediction approach described in section 3. We train a logistic-regression classifier on feature vectors $\mathbf{f}(x) = \mathbf{A}\mathbf{g}(x)$ where $\mathbf{A}$ is derived using the method in section 3.1. Using the $8,000$ training images, we created 100 *auxiliary* training sets corresponding to the 100 most frequent content words in the captions.[3] Each training set involves prediction of a particular content word. The input to the classifier is the SIFT representation of an image. Next, we trained linear classifiers on each of the 100 auxiliary training sets to induce parameter vectors $\mathbf{w}_1 \ldots \mathbf{w}_{100}$. Each parameter vector is of dimension $1,000$; we will use $\mathbf{W}$ to refer to the matrix of size $1,000 \times 100$ which contains all parameter values. The projection matrix $\mathbf{A}$ consists of the $h$ eigenvectors in $\mathbb{R}^d$ which correspond to the $h$ largest eigenvalues of $\mathbf{W}\mathbf{W}'$.

### 5.5. Cross-Validation of Parameters

There are two free parameters in the data-PCA and the predictive structure models: the dimensionality of the projection $h$, and the constant $C$ used in Eq. 1. A single topic—the 7th most frequent topic in the training data—was used to tune these parameters for both model types. For each model type the model was trained on all training sets $T_{n,7}$

[3] Content words are defined as any words which do not appear on a "stop" list of common function words.

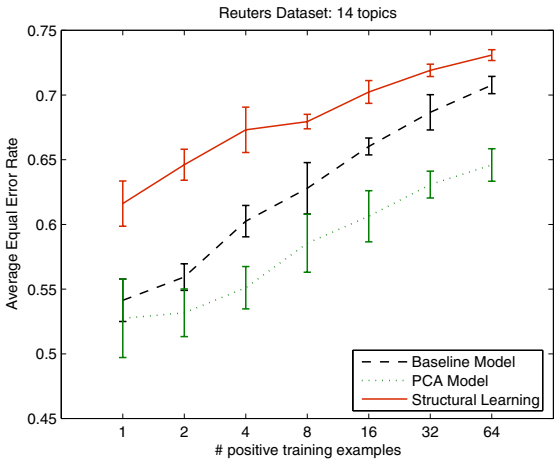


Figure 7. Equal error rate averaged across topics, with standard deviations calculated from ten runs for each topic. The equal error rates are averaged across 14 topics; the 7th most frequent topic is excluded as this was used for cross-validation (see section 5.5).

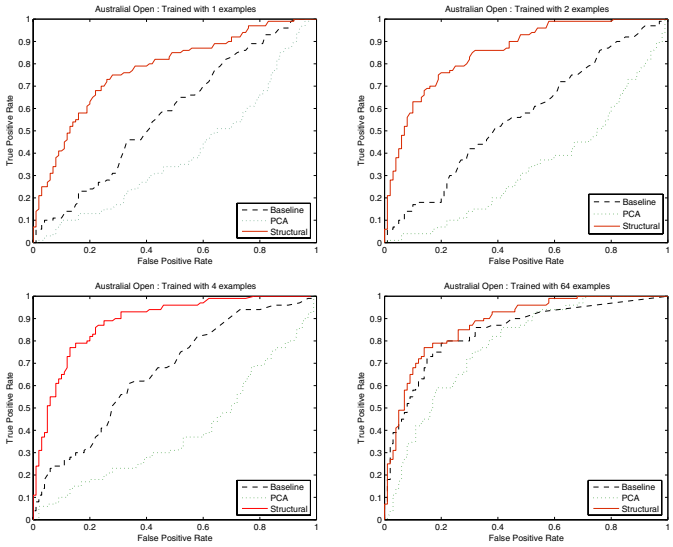


Figure 8. Roc Curves for the "Australian Open" topic.

for $n = 1, 2, 4, 8, ..., 64$, with values for $h$ taken from the set $\{2, 5, 10, 20, 30, 40, 100\}$ and values for $C$ chosen from $\{0.00001, 0.0001, \ldots, 1000\}$. Define $E^n_{h,C}$ to be the equal-error-rate on the development set for topic 7, when trained on the training set $T_{n,7}$ using parameters $h$ and $C$. We choose the value $h^*$ for all experiments on the remaining 14 topics as

$$h^* = \arg\min_h \sum_{i=1,2,\ldots,64} \min_C E^i_{h,C}$$

This corresponds to making a choice of $h^*$ that performs well on average across all training set sizes. In addition, when training a model on a training set with $i$ positive examples, we chose $C^*_i = \arg\min_C E^i_{h^*,C}$ as the regularization constant. The motivation for using a single topic as a validation set is that it is realistic to assume that a fairly substantial validation set (1,000 examples in our case) can be created for one topic; this validation set can then be used to choose values of $h^*$ and $C^*_i$ for all remaining topics.

### 5.6. Results

Figure 7 shows the mean equal error rate and standard deviation over ten runs for the experiments on the Reuters dataset. The equal error rate is the recall occurring when the decision threshold of the classifier is set so that the proportion of false rejections will be approximately equal to the proportion of false acceptances. For example an equal error rate of 70 percent means that when the proportion of false rejections is equal to the proportion of false acceptances 70 percent of the positive examples are labeled correctly and 30 percent of the negative examples are misclassified as positive.

For all training set sizes the structural learning model leads to improved performance. The average performance with one positive training example is around 62% with the structural learning method; to achieve similar performance with the baseline model requires between four and eight positive examples. Similarly, the performance with 4 positive examples for the structural learning method is around 67%; the baseline model requires between 32 and 64 positive examples to achieve this performance. PCA's performance is lower than the baseline model for all training sizes and the gap between the two increases with the size of the training set.

Structural learning improves performance for all but three of the topics. Figures 8 and 9 show equal error rates for two different topics. The first topic, "Australian Open", is one of the topics that exhibits the most improvement from structural learning. The second topic, "Winter Olympics", is one of the three topics for which structural learning does not improve performance. As can be observed from the Australian Open curves the use of structural features speeds the generalization ability of the classifier. The structural model trained with only two positive examples performs comparably to the baseline model trained with sixty four examples. For the Winter Olympics topic the three models perform similarly. At least for a small number of training examples, this topic exhibits a slow learning curve; i.e. there is no significant improvement in performance as we increase the size of the labeled training set; this suggests that this is an inherently more difficult class.

## 6. Conclusions

We have described a method for learning visual representations from large quantities of unlabeled images which have associated captions. The method makes use of auxilliary training sets corresponding to different words in the

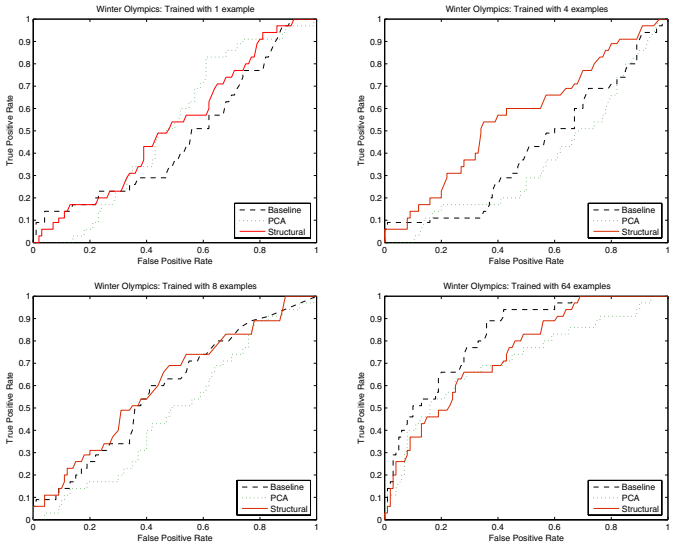


Figure 9. Roc Curves for the "Winter Olympics" topic.

captions, and structural learning, which learns a manifold in parameter space. The induced representations significantly speed up learning of image classifiers applied to topic classification. Our results show that when meta-data labels are suitably related to a target (core) task, the structure learning method can discover feature groupings that speed learning of the target task. Future work includes exploration of automatic determination of relevance between target and auxilliary tasks, and experimental evaluation of the effectiveness of structure learning from more weakly related auxiliary domains. We would also like to explore other manifold learning techniques such as NMF and LDA.

## References


[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005. 1, 2, 3, 4

[2] S. Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *In Neural and Information Processing Systems (NIPS)*, 1998. 2

[3] K. Barnard, P. Duygulu, D. Forsyth, and N. D. Freitas. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003. 2

[4] J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28:7–39, 1997. 2

[5] B.Epshtein and S.Ullmam. Identifying semantically equivalent object fragments. In *Proceedings of CVPR-2005*, 2005. 2

[6] CMU. http://www.informedia.cs.cmu.edu/. 2

[7] Fei-Fei, P. Perona, and R. Fergus. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence*, 28(4), 2006. 2

[8] R. Fergus, F.-F. L., P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. of the 10th Inter. Conf. on Computer Vision, ICCV 2005*, 2005. 2

[9] K. Grauman and T. Darrell. The pyramid match kernel:discriminative classification with sets of image features. In *Proceedings fo the International Conference on Computer Vision (ICCV)*, 2005. 2

[10] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, 1998. 2

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of CVPR-2006*, 2006. 2

[12] T. Miller, A. Berg, J. Edwards, M. Maire, and R. White. Names and images in the news. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[13] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *Proceedings of CVPR*, 2006. 2

[14] K. Nigam, A. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134, 2000. 2

[15] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine learning*, pages 713–720, 2006. 2

[16] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, Univ. of Edinburgh, 2001. 2

[17] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005. 2

[18] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering object categories in image collections. *Proc. Int'l Conf. Computer Vision, Beijing, 2005*, 2005. 3

[19] S. Thrun. Is learning the n-th thing any easier than learning the first? In *In Advances in Neural Information Processing Systems*, 1996. 2

[20] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *Pattern Analysis and Machine Intelligence*, In press, 2006. 2

[21] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Proceedings of CVPR-2006*, 2006. 2