Large-scale Knowledge Transfer for Object Localization in ImageNet

Matthieu Guillaumin
ETH Zürich, Switzerland

Vittorio Ferrari
University of Edinburgh, UK

Abstract

ImageNet is a large-scale database of object classes with millions of images. Unfortunately only a small fraction of them is manually annotated with bounding-boxes. This prevents useful developments, such as learning reliable object detectors for thousands of classes. In this paper we propose to automatically populate ImageNet with many more bounding-boxes, by leveraging existing manual annotations. The key idea is to localize objects of a target class for which annotations are not available, by transferring knowledge from related source classes with available annotations. We distinguish two kinds of source classes: ancestors and siblings. Each source provides knowledge about the plausible location, appearance and context of the target objects, which induces a probability distribution over windows in images of the target class. We learn to combine these distributions so as to maximize the location accuracy of the most probable window. Finally, we employ the combined distribution in a procedure to jointly localize objects in all images of the target class. Through experiments on 0.5 million images from 219 classes we show that our technique (i) annotates a wide range of classes with bounding-boxes; (ii) effectively exploits the hierarchical structure of ImageNet, since all sources and types of knowledge we propose contribute to the results; (iii) scales efficiently.

1. Introduction

Bounding-boxes on objects provide a valuable estimation of the regions of interest in an image. Therefore, bounding-box annotations are essential for several important applications, such as object class detection and recognition [5, 14] or image segmentation [18, 22, 26]. In turn several other computer vision applications depend on these methods, *e.g*. object tracking [17], viewpoint classification [15] and human pose estimation [2]. However, manually annotating images with object bounding-boxes is tedious and very time consuming, which prevents the above applications from going large-scale. For example, a good object detector requires at least 1000 bounding-boxes for training. Therefore, learning detectors for 10k classes would require 10 millions of bounding-boxes!

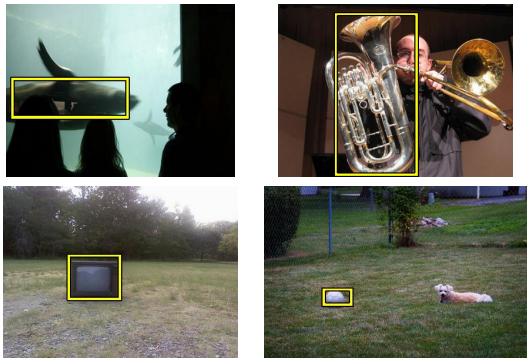


Figure 1. Examples of automatically populated bounding-boxes using our approach on 0.5 million images of ImageNet.

On the other hand, we have recently witnessed the advent of large scale datasets for other computer vision applications, including image search [10] and object classification [6, 31, 33]. In this paper, we want to bridge the gap between these domains by automatically populating the large-scale ImageNet [7] database with bounding-boxes (fig. 1). ImageNet contains millions of images, only a small fraction of which is already manually annotated with bounding-boxes. Moreover, most annotations are concentrated in few classes, leaving many classes completely unannotated. The key idea of our method is to exploit the semantic hierarchy of ImageNet to localize objects in novel target classes by transferring knowledge from related source classes with available annotations. Below in sec. 1.1 we sketch our approach and review related work in sec. 1.2.

1.1. Overview of our method

Goal. The objective of this work is to automatically *localize* objects in images of classes without available bounding-box annotations, by *transferring knowledge* from classes which do have them. The semantic hierarchy of ImageNet naturally suggests which source classes are most likely to help localizing a certain target class. Indeed, classes closer in the hierarchy are semantically more related. Cats and dogs are closer (they are both *carnivores*) than cats and trees (they are both *organisms*, a more general concept).

Knowledge sources and types. To support localizing objects of a target class T, we consider two types of related classes as sources: ancestors (A) and siblings (S) (fig. 2).

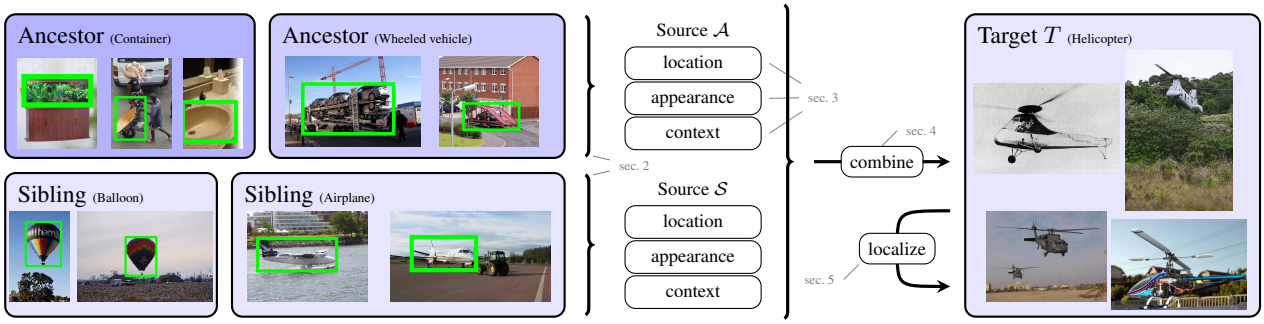


Figure 2. Our knowledge transfer pipeline for localizing objects in ImageNet. We transfer three types of knowledge (location, appearance and context) from two sources (siblings $\mathcal{S}$ and ancestors $\mathcal{A}$) to a target class T (without bounding-box annotations). Multiple knowledge sources and types are combined optimally to drive the localization of objects at T.

Each source provides several types of visual knowledge: the *appearance* of its objects, their *location* distribution within the image, and the *context* in which the objects are embedded. Knowledge of the appearance of objects is useful because semantically related classes typically have more similar appearance than unrelated classes [6, 9]. Location knowledge provides information about the expected scale, aspect-ratio and position of the objects in their images. Context knowledge models the appearance of background patterns in the images of the source class. Since related classes are likely to occur against similar backgrounds, this helps suppressing background in images of T. The remaining regions are more likely to contain instances of T. In sec. 2 and sec. 3 we detail all pairs of knowledge sources and types and explain how to learn them.

Window distributions. We use each pair of knowledge source r and type t to evaluate windows in an image of the target class T. This induces a probability distribution p_{rt} over windows in the image, which helps *reducing the uncertainty in the location* of objects of T. More precisely, the probability $p_{rt}(w|I)$ of a window w in an image I indicates how likely it is to contain an instance of T according to knowledge source r and type t. By combining the distributions induced by several knowledge sources and types we can further reduce the location uncertainty at T (fig. 3). In sec. 4 we explain how to find the optimal combination of distributions so that the window with the highest probability localizes an instance of T as accurately as possible.

Localizing objects of the target class. We employ the combined distribution in a procedure to jointly localize objects in all images of the target class. The procedure iterates two steps. The first step selects the window with the highest probability in each image. From this initial estimation of localization of instances of T, the second step learns an appearance model of the target class. This generates probability distributions over windows in *all images*, which are fed to the next iteration. The key aspect of this procedure is that knowledge about the target class is transmitted between images over the iterations. As we demonstrate in the experiments, this improves localization accuracy (sec. 5).

1.2. Related work.

Transfer learning. Our work is related to previous works on transfer learning [29] in computer vision, where learning a new class (target) is helped by labeled examples of other related classes (sources) [3, 8, 13, 16, 20, 24, 25, 27, 28, 30]. The goal of these works is to reduce the number of examples necessary to learn the target, improving generalization from a few examples. Many methods use the parameters of the source classifiers as priors for the target model [3, 13, 27, 30]. Other works [16, 25] transfer knowledge through an intermediate attribute layer, which captures visual qualities shared by many object classes (e.g. 'striped', 'yellow'), or through prototypes [24]. A third family of works transfer object parts between classes [20, 28], such as wheels between cars and bicycles or legs between cows and horses.

In this paper we have a different goal. By trying to populate ImageNet with bounding-boxes, we effectly want to reduce the *degree of supervision* necessary to learn models requiring them [14, 18] to just image labels. These are cheaper to annotate and are available for all ImageNet images. To achieve this goal, we employ the source classes in a different manner than transferring priors of model parameters, attributes or parts. Instead, we transfer probability distributions over windows in images of the target class. These reduce the uncertainty in the location of the target class. We transfer in this manner several types of knowledge from the source classes. In addition to the appearance of the source objects [3, 30], we also transfer knowledge of their location within an image, and of the background context in which they are embedded. Finally, note how we automatically select the relevant source classes among thousands, thanks to the semantic hierarchy of ImageNet. In contrast, in many previous works the overall pool of source classes is small and manually defined (*e.g.*, up to 40 [16, 25]).

Our work is related to [8], which builds a fully connected CRF to localize objects in images of a class. It uses objectness as a unary potential and appearance similarity between windows as pairwise potentials. We go beyond [8] by transferring multiple knowledge types from multiple sources and making it much more efficient to handle large-scale data.

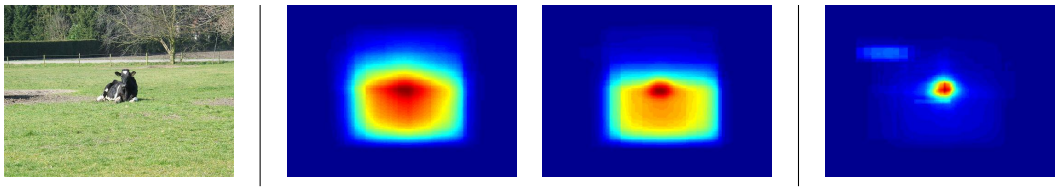


Figure 3. An image I of a cow (left) and some window distributions $p_{rt}(w|I)$ for it. To illustrate a distribution, we color a pixel by accumulating the probabilities $p_{rt}(w|I)$ of each window w containing it. Individual source and type pairs such as scale from siblings ($p_{sl}(w|I)$, middle-left) and appearance from ancestors ($p_{ao}(w|I)$, middle-right) already induce valuable information about the location of the target object in I. The optimal combination $p_c(w|I)$ of all location, appearance and context cues from both sources further reduces the uncertainty in the location of the object (sec. 4, right).

ImageNet. ImageNet [7] is a recent large-scale hierarchical database of images. ImageNet forms a directed acyclic graph (DAG) where the classes are vertices linked by directed edges that represent parent-child relations: *Aircraft* is a parent of *Airplane* because an airplane "is an" aircraft. As of the summer of 2011, ImageNet contains 12 million images for 17000 classes. Most previous works on ImageNet study the relation between visual appearance and semantics [9], image classification [6, 19] or object detection in the fully supervised setting.[1] To our knowledge, ours is the first work trying to automatically populate ImageNet with new bounding-box annotations.

2. Knowledge sources

A knowledge source for a target class T can be formally defined as the output of a sampling process that returns a set of windows from the images of a set of classes. Seeing, *e.g.*, *airplane* as a knowledge source for *helicopter* will return a (large) set of windows from images of airplanes.

Because we aim at large-scale experiments, the first step of the sampling process is to reduce the number of windows considered in an image. For this we use the "objectness" technique of [1] to sample $N = 1000$ windows per image that are likely to cover all objects in the image. See fig. 4(a) for examples of objectness windows. This will greatly reduce the computational complexity of learning the knowledge types and applying them to images of the target class, compared to a standard sliding-window approach.

Although the 1000 windows from the objectness measure are biased towards objects, many of them also cover background regions. This enables to uniformly obtain both kinds of training windows when learning the various knowledge types (*e.g.*, our novel context model is based on background windows, sec. 3.3).

We explain below the later steps of sampling, which are specific to the two knowledge sources we consider. The set of initial 1000 objectness windows sampled from an image I are denoted $\mathcal{I}_O$.

Siblings. Sibling classes S are closely semantically related to the target class T as they share a parent node. For example, *giroplane*, *balloon*, *helicopter* and *airplanes* are siblings, as they are all children of *aircraft*. *Aircraft* is itself a sibling to *wheeled vehicle* (*car, bicycle, skateboard, ...*), as they are both children of *vehicle*, etc.

The siblings source $\mathcal{S}$ is a large set containing all positive and negative windows sampled from each image of S that has a ground-truth bounding-box annotation. The positive windows for an image I are all windows in $\mathcal{I}_O$ that overlap more than 80% with a ground-truth bounding-box. The overlap is measured by the PASCAL VOC criterion, i.e. the area of intersection divided by the union. We also flip all images horizontally and repeat the process. This procedure delivers many small variations of the ground-truth bounding-box, which reflect the level of localization error of the initial objectness samples. Therefore, they represent well what we will observe later in images of the target class.

Negative windows are uniformly sampled among all windows in $\mathcal{I}_O$ having no overlap with any ground-truth.

Ancestors. Ancestors A of T include all parents of T, up to the root, and their descendants (*excluding* the subtree rooted at T itself). For example, the sequence of ancestors of *jet* is *airplane, aircraft, vehicle, conveyance, instrumentation, etc.* An ancestor conveys rather generic knowledge which applies to all descendant classes, including T. Note how ImageNet is divided into a small number of separate DAGs, each with its own root, because it contains only physical objects (as opposed to all synsets of WordNet). Therefore, the most general ancestor we consider for T is the root of the DAG it belongs to. In our experiments all target classes have either *animal* or *instrument* as their root.

We form the ancestor source $\mathcal{A}$ as a large set containing all positive and negative windows sampled from each image of A that has a ground-truth bounding-box annotation. The positive and negative windows are defined in the same manner as for siblings.

3. Knowledge types

We describe here the three knowledge types we propose, how we learn them for each knowledge source and how they induce window distributions in images of the target class T (sec. 3.1 to 3.3). We propose three knowledge types: loca-

[1] See the ImageNet Large Scale Visual Recognition Challenge 2011.

tion, appearance and context. Each type is in turn composed of several subtypes that grasp specific aspects of the knowledge to be transferred.

3.1. Location

Siblings. We expect siblings to share the typical location at which they appear in the image to some degree. We parametrize a window $w = (x, y, \log(WH), \log(W/H))$ by the (x, y) coordinates of the center, the scale $\log(WH)$, and the aspect ratio $\log(W/H)$ (all measurements are relative to the size of the image). This parametrization offers several advantages over a more classic $w = (x, y, W, H)$. First, aspect-ratio and scale are more statistically independent than width and height. Moreover, as noticed by [14], aspect-ratio is related to rough object shape and to the canonical viewpoints it is imaged in. Therefore, we expect it to transfer well between siblings. Finally, the logarithm distributes the windows more uniformly over the 4D space.

We treat each coordinate in our parametrization as a separate location knowledge subtype l. For each subtype we learn a probability distribution $p_{sl}(w)$ with 1D kernel smoothing density estimation [21] of the positive windows in the sibling source set $\mathcal{S}$. Each density $p_{sl}(w)$ is then discretized into 100 bins for rapid access.

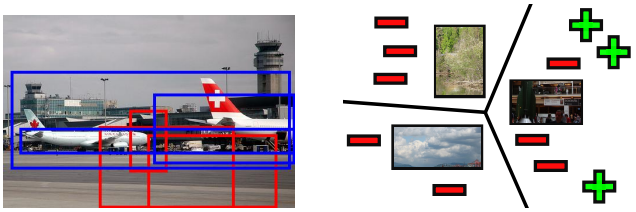
After learning $p_{sl}(w)$, we use it to evaluate each window w in an image of the target class T. For this we read out the probability of the bin w falls into. This results in a window distribution in each image of T. Note how working with 4 separate 1D KDE is substantially faster and less memory consuming compated to a single 4D KDE. Moreover, the probabilities in the $4 \cdot 100$ bins are better estimated from the available training data than 100^4 bins.

Ancestors. In the same way as for siblings, we learn the four subtypes $p_{al}(w)$ for ancestors from the positive windows in the ancestor source set $\mathcal{A}$. After learning, $p_{al}(w)$ is used to evaluate windows of T similarly as for siblings.

3.2. Appearance

Siblings. Because of their semantic similarity, siblings often share parts (*car*, *bicycle* and *skateboard* have wheels) or other appearance characteristics such as texture (many animals are furry) or shape (as *horse*, *zebra* and *donkey*). We expect strong benefits from transferring siblings appearance and train for them discriminative classifiers on a variety of rich, dense image features. For this we use both positive and negative windows from the sibling source set $\mathcal{S}$. We describe them using well-established descriptors:

- *Lab* color histograms quantized into 4000 bins
- Bag-of-word histograms [34] on 2000 visual words based on dense SURF features [4].
- HOG descriptors [5] (8×8 grid of normalized gradient orientation histograms).



(a) Objectness windows: the three windows with the highest (blue) and lowest (red) objectness sampled in an image of *airliner*.

(b) Context knowledge: background prototypes of siblings are shown surrounded by positive and negative windows.

Figure 4. Objectness windows and context knowledge.

We train a large-scale linear SVM [12] on the HOG descriptors using 95% of the windows in $\mathcal{S}$. As reported in [34], the χ^2 kernel leads to better results for Lab and SURF. To retain high computational efficiency, we approximate the χ^2 kernel with the expansion technique of [32], followed by training a linear SVM. After training, we use the remaining 5% of the windows in $\mathcal{S}$ as validation data to fit a sigmoid on the output of the SVM, so as to obtain a probabilitic output. This leads to a function $p_{sd}(w|I)$ for each descriptor d.

After learning p_{sd}, we evaluate each window w in every image I of T, resulting in a window distribution for I.

This scheme involves computing dense appearance descriptors for 1000 windows in each of 500k images, for a total of half a billion windows. In sec. 6 we provide technical details on how to efficiently compute these descriptors, notably HOG for windows of arbitrary aspect-ratios.

Note how we sample negative training windows from the same images as the positives. This circumvents the lack of a clear concept of negative classes in a hierarchical data set, allowing us to cleanly transfer knowledge from any source to any target. Moreover, in our application the learned SVMs are only used in images containing T, and not in images containing arbitrary negative objects as in [11].

Ancestors. Because the ancestors of T go up to a root of ImageNet, they convey rather generic knowledge that applies to the broad set of their descendant classes, rather than being particularly related to T. Therefore, we adopt the objectness measure [1] as the appearance model of ancestors. This measure tries to distinguish windows containing any object with a well-defined boundary and shape, such as cows and telephones, from amorphous background windows, such as grass and road. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary or being unique within the image. Objectness estimates the probability that a window is covering an object of any class.

We train parameters of objectness using the windows in the ancestor source set $\mathcal{A}$. In practice all our target classes have either animal or instrumentation as their root, and objectness is a generic measure that does not depend on the

exact class distribution in its training set. Therefore, we learn only two sets of objectness parameters overall, one per root, from 1000 randomly sampled images (excluding all our target classes). This results in a function $p_{ao}(w|I)$.

For every image of T, we use the objectness model of the ancestor of T to evaluate every window w, yielding a window distribution for the image.

3.3. Context

Siblings. We expect siblings to occur in front of similar backgrounds. For example, different *fishes* often appear in water, *bovines* on grass and *wheeled vehicles* on roads. We transfer this type of knowledge to suppress windows in images of the target class that are visually similar to backgrounds seen in siblings. This will reduce the location uncertainty of the target class. In some cases this context knowledge might transfer even better than the appearance of the objects themselves (sec. 3.2). Objects in sibling classes might be more different than the backgrounds they stand against (*e.g.*, cows and horses are visibly different, whereas the grass/sky behind them is exactly the same).

We learn context as follows. We first find background prototypes $\{C_k\}_{k=1...K}$ by clustering the negative windows of 50% of the images in the sibling source set $\mathcal{S}$ using k-means on their appearance descriptors ($K = 100$). We then collect both negative and positive windows from the remaining 50% of the images, which we use to identify "good" and "bad" prototypes. We consider the descriptor distance between windows w and their closest prototype C_{k_w}. The key idea of our context modeling is that, among windows assigned to a cluster C_k, background windows w_b are closer to the prototype than objects w_o (fig. 4(b)):

$$\forall w_b, w_o \text{ s.t. } k_{w_b}=k_{w_o}=k, \quad \chi^2(w_b, C_k) \leq \chi^2(w_o, C_k) \quad (1)$$

An object window will typically be relatively far from most background prototypes. However, because the appearance descriptors are imperfect and objects of the same class show a lower visual variability, most object windows are likely to be assigned to a few 'bad' prototypes. Therefore, we want to learn per-prototype thresholds on the distance, to decide whether a new window really is background. We feed a linear SVM with a sparse $(K+1)$-D representation z_w of the cluster assignment and distance for a window w:

$$z_w = [0, \ldots, 0, 1, 0, \ldots, 0, \chi^2(w, C_{k_w})]^\top \quad (2)$$

where the 1 is at position k_w, indicating the cluster assignment. This SVM effectively learns a global threshold and a per-prototype penalty on the distance value to classify windows as being typical background or not. If not, the window has a 'novel' appearance, as it does not look like any background prototype, and is deemed more likely to be an object. The crucial point is that this process *does not entail the appearance of sibling objects themselves*. We then fit a sigmoid on the output of the SVM to obtain a probabilistic function $p_{sc}(w|I)$. After learning, we use p_{sc} to produce a window distribution for each image of T.

Ancestors. Over hundreds of target classes, it is likely that some siblings have specific backgrounds that transfer poorly. Therefore, we learn a context knowledge model from the ancestor source $\mathcal{A}$ alike. In this a way we also learn a generic background model that applies to many classes.

4. Combining window distributions

In the previous sections we have explained how every pair of knowledge source and type induces a probability distribution over windows in an image of the target class T. Here we explain how to *combine* them into a new distribution p_c, which will support localizing objects of T more accurately than any individual distribution p_k (fig. 3).

Combination model. To combine the distributions $p_1(w|I) \ldots p_K(w|I)$ representing the K knowledge distributions at image I of T, we consider weighted geometric means, parametrized by a weight vector $\boldsymbol{\alpha} = [\alpha_k]$

$$p_c(w|I) = \prod_k p_k(w|I)^{\alpha_k} \quad (3)$$

where $\sum_k \alpha_k = 1$. This corresponds to weighted averages of log-probabilities: $\log p_c(w|I) = \boldsymbol{\alpha}^\top \boldsymbol{\ell}^I_w$ where $\boldsymbol{\ell}^I_w = [\ell^I_{wk}] = [\log p_k(w|I)]$.

Learning weights. What makes a good combined distribution? Formally, in each image I of T, the window w with the highest probability should be the one which best overlaps with the object bounding-box b. Let the overlap be the PASCAL VOC measure $\mathrm{ovl}_I(w) = \frac{w \cap b}{w \cup b}$. This measure is averaged over all images, so we look for the combination weights $\boldsymbol{\alpha}$ that maximize the following objective function:

$$f(\boldsymbol{\alpha}) = \sum_I \mathrm{ovl}_I \left(\operatorname*{argmax}_{w \in I} \boldsymbol{\alpha}^\top \boldsymbol{\ell}^I_w \right). \quad (4)$$

Of course at test time we do not have ground-truth bounding-boxes b, so we learn $\boldsymbol{\alpha}$ on a separate tree of ImageNet. To ensure complete separation of training and test sets, we train one $\boldsymbol{\alpha}$ for all *animal* classes on 5000 images from *instruments*, and vice-versa.

Since $w \in I$ is restricted to a discrete set of 1000 sampled windows, the objective function (4) is non-differentiable and impractical to optimize directly. We propose to approximate f with $\hat{f}$ by replacing the argmax operation with a softmax, such that $\hat{f}$ is also the expected overlap with the ground-truth bounding-box, under the distribution $p_c(w|I)$:

$$f(\boldsymbol{\alpha}) \approx \hat{f}(\boldsymbol{\alpha}) = \sum_I \sum_{w \in I} \mathrm{ovl}_I(w) \sigma_w(\boldsymbol{\alpha}), \quad (5)$$

$$\sigma_w(\boldsymbol{\alpha}) = \frac{\exp(\boldsymbol{\alpha}^\top \boldsymbol{\ell}^I_w)}{\sum_{w' \in I} \exp(\boldsymbol{\alpha}^\top \boldsymbol{\ell}^I_{w'})} = \frac{p_c(w|I)}{\sum_{w' \in I} p_c(w'|I)}. \quad (6)$$

The approximate objective function $\hat{f}$ is differentiable, and we optimize it using gradient ascent under the constraint $\sum_k \alpha_k = 1$. The partial derivatives of $\hat{f}$ are:

$$\frac{\partial}{\partial \alpha_k} \hat{f}(\boldsymbol{\alpha}) = \sum_I \sum_{w \in I} \mathrm{ovl}_I(w) \sigma_w(\boldsymbol{\alpha}) (\ell^I_{wk} - \bar{\ell}^I_k(\boldsymbol{\alpha})), \quad (7)$$

where $\bar{\ell}^I_k(\boldsymbol{\alpha}) = \sum_{w \in I} \ell^I_{wk} \sigma_w(\boldsymbol{\alpha})$.

Since $\hat{f}$ is not concave, it has potentially several local maxima. To avoid getting stuck in one, we first run a grid search on f over the $K-1$ simplex. We select the best point on this grid as initialization for the gradient ascent on $\hat{f}$.

5. Localizing objects of the target class

At this point we have a combined distribution $p_c(w|I)$ over windows in each image I of the target class T. Now we localize objects by selecting the window w^*_I with the highest probability in each image independently (mirroring the criterion optimized in (4)). The resulting set $\{w^*_I\}_{I \in T}$ of windows is an initial estimate of the location of the objects at T. Next we learn a bag-of-SURF appearance model p_{ta} specific to the target class from $\{w^*_I\}_{I \in T}$ (analog to what is done in sec. 3.2). We use p_{ta} to evaluate all windows in T and combine the resulting distribution to all existing ones from ancestors and siblings as in eq. 3 (giving 0.5 weight to the new term and 0.5 to the sum of all others). Finally, we re-localize objects by selecting the window with the highest probability in each image.

In this scheme, images at T communicate their knowledge of the appearance of the target class through the appearance model. Therefore it enables to localize objects *jointly* over all images. This is computationally much more efficient than building a complex CRF with pairwise potentials between all images, as done by [8]. The complexity of such a model grows quadratically with the number of images and the number of candidate windows, making it impractical in our large-scale setup.

6. Fast feature extraction

In order to apply our large-scale knowledge transfer mechanism, describing a window must be computationally very efficient. As we consider 1000 windows per image in 0.5M images, we have to process half a billion windows. Moreover, we have to recompute descriptors every time they are needed, as they would take too much disk space.

Bag-of-word descriptors. For SURF and Lab histograms one could apply the integral histogram speedup [23]. However, for large codebooks (*e.g*. 2000 for SURF) the integral histogram is too large to store on disk. Instead, we precompute the quantization into an array A of the same size as the image, where each pixel contains the codebook entry and store it on disk. When a window descriptor is needed, it can be efficiently computed by integrating A over its surface. We compute SURF descriptors on a grid of 64×64 points, leading to a very small array A. This algorithm takes only about 0.5s on a standard desktop to compute 2000D bag-of-SURF for 1000 windows (after 3s to precompute A).

HOG. HOG descriptors present an additional challenge. Although windows come in many different scales and aspect-ratios, each window must get a fixed-size descriptor, *i.e*. a fixed grid of HOG cells. In traditional use of HOG for object detection, this is not a problem as the aspect-ratio of all windows is *predermined beforehand*, and the scale dimension is factored out by building a scale-space pyramid [5, 14]. In our scenario instead, aspect-ratios vary arbitrarily. The straighforward but slow solution would be to crop each window, rescale it to a canonical size and aspect-ratio, and compute HOG. Instead, we build a 2D scale–aspect-ratio pyramid. We can now compute a descriptor for an arbirary window w with any given number $a \times b$ of HOG cells. For this we efficiently extract the HOG descriptor of the most overlapping window w^* in the 2D pyramid:

1. Compute the best possible overlap $U(l, m)$ of any $a \times b$ window at each pyramid level (l, m). This can be done in constant time by translating w to the origin and measuring the overlap to the window $(0, 0, a, b)$.
2. Iteratively explore the levels starting from the largest $U(l, m)$, searching for the best overlapping window $w^*(l, m)$. If its overlap $O(l, m)$ improves over the previously explored levels, $O(l, m)$ becomes a lower bound of the best possible overlap. We can then safely discard all levels (l', m') such that $U(l', m') < O(l, m)$ and proceed to the next level if any is left.

The returned window w^* is guaranteed to be the most overlapping to the query w, with a descriptor of fixed size $a \times b$. It takes only about 0.5s to compute descriptors for 1000 windows in an image (after 2s to precompute the 2D pyramid).

7. Experiments

Setup. To quantitatively evaluate our approach we identify target classes in ImageNet with some ground-truth bounding-boxes for themselves and for their siblings. We select them automatically by top-down searching ImageNet until classes have less than 10.000 images and have siblings with annotations. This procedure yields 219 targets (with 609 subclasses overall and an average depth of 12.7 in WordNet) with a total of 502.131 unique images, only 61.786 (12.3%) with ground-truth annotations. For each target, we use the annotations only to evaluate how well our method localizes objects (they are not given as input to the algorithm). Therefore, while we report performance averaged over these 62k images, our method returns a bounding-box on all 0.5M images. The quantitative evaluation we report is a good estimate of the overall annotation quality over the 0.5M images, importantly also for future target classes for which no manual annotation exists at all.

Knowledge types	Location						Appearance						Context		All types		Sec. 5
	aspect-ratio	scale	x	y	*all subtypes*	*all sources*	SURF-χ^2	Lab-χ^2	HOG	Objectness	*all subtypes*	*all sources*	SURF-χ^2	*all sources*	*one source*	*all sources*	*with target*
Siblings	45.0	**49.8**	43.3	45.1	52.6	**52.7**	**51.2**	36.4	39.0	-	48.6	**53.1**	**49.9**	**50.7**	53.9	**54.1**	**55.2**
Ancestor	44.4	44.5	43.6	45.2	50.8		-	-	-	50.5	50.5		49.8		53.3		

Table 1. Localization accuracy averaged over all 219 target classes for all our individual knowledge subtypes/sources, and their combinations. All combinations weights are learnt as described in sec. 4. The last column integrates target-specific appearance models as described in sec. 5. For each knowledge type, we show in bold the best subtype and the best combination of sources.

We measure localization accuracy by the PASCAL-overlap of the window returned by the algorithm with the ground-truth box, averaged over the images of a target class, and then averaged over the 219 target classes. The average overlap of the best objectness window is 84.7%, out the 1000 we sample (sec. 2).

Results. In tab. 1, we show the performance of all individual knowledge subtypes and all combinations that we learnt (sec. 4). Context models are based on SURF only, as they give the most meaningful background prototypes. In terms of location knowledge, siblings provide a better estimate of the aspect-ratio and scale of the target than ancestors, whereas (x, y) position is transferred equally well from the two sources. Concerning appearance, SURF outperforms the other descriptors, which are too rigid (HOG) or not descriptive enough (*Lab*) to transfer well between classes. Interestingly, the objectness measure performs very well, confirming its value as a generic object bias [1]. Regarding context, both sources perform about as well, indicating that the range of typical backgrounds is limited enough to be modeled well with a few generic prototypes (from ancestors). Most importantly, every combination of (sub)-types and/or sources improve over all its components.[2] This shows the effectiveness of the combination technique we propose (sec. 4), which automatically learns weights for the components. The results confirms that all sources and types of knowledge we propose are worth transferring and contribute to the localization accuracy of target objects.

Using all sources and types leads to our best initial estimation of the location of target objects (54.1%). Using the method of sec. 5 to add a target-specific appearance model learned from the initial localizations, performance improves further to 55.2%. This confirms that *jointly* localizing objects of T over all its images is important, as it benefits from communication between images.

Fig. 5 shows the distribution of overlaps between the windows output by our method and the ground-truth. According to the PASCAL criterion [11], we localize the object correctly in 58% of the images (overlap$\geq$50%). On average, these objects are localized accurately (71.5% overlap). This might be sufficient to train state-of-the-art object detectors such as [14]. Fig. 1 shows qualitative results.

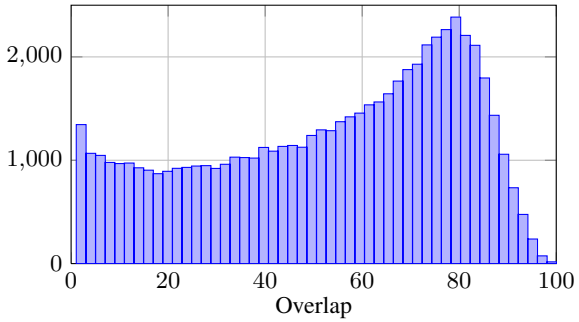


Figure 5. Histogram of overlaps between the windows output by our method and the ground-truth.

As a last experiment, we compare our approach to LocLearn, a state-of-the-art weakly supervised localization method [8]. However, LocLearn is computationally expensive and cannot be run on a large-scale. Hence, we selected 10 random classes (5 *animals*, 5 *instruments*), each with the subset of images with available ground-truth bounding-boxes (in order to evaluate performance; they are not given to the algorithm). For tractability, we were also forced to reduce the number of candidate objectness windows to 100, as done originally in [8]. LocLearn achieves an average localization accuracy of 48.6% compared to our 51.6% on the same classes. This highlights the benefits of transferring knowledge of multiple types and from multiple source classes which is the main contribution of our paper. On average, after the above reductions, LocLearn is about one order of magnitude slower than our approach even though we operate on ten times more images per class and ten times more windows per image.

8. Conclusion

We have proposed a large-scale knowledge transfer mechanism to exploit the semantic hierarchy of ImageNet to support localizing instances of new target classes. We proposed two sources and three types of knowledge and a technique for learning their optimal combination so as to maximise localization accuracy in images of a target class.

[2] Except for all siblings appearance subtypes vs. SURF alone. We believe this is due to a bad local maximum when learning the weights $\boldsymbol{\alpha}$.

We use an iterative scheme to integrate a target-specific appearance model to further improve performance. On experiments over 219 classes and 0.5M images, we have shown that all knowledge sources and types help localizing the objects in the images and that the learned combinations always improve over their components. We release online[3] the codes for objectness and fast HOG, as well as the 0.5M bounding-boxes produced by our method.

We believe this can be seen as a landmark towards the ultimate goal of populating the entire ImageNet (>10M images) with high quality bounding-boxes.

In future work, we would like to exploit descendant classes as an additional, special source. The key idea is that visual variability decreases with the depth of the class in the tree. Hence, localizing objects in image subsets corresponding to a subclass should be easier than directly on the target class. We could exploit this observation to localize objects recursively in a bottom-up fashion.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[2] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, 2010.

[3] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. SURF: Speeded up robust features. *CVIU*, 2008.

[5] N. Dalal and B. Triggs. Histogram of Oriented Gradients for human detection. In *CVPR*, 2005.

[6] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, 2010.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. `http://image-net.org/`.

[8] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010.

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

[10] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *CIVR*, 2009.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[13] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop of Generative Model Based Vision*, 2004.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on PAMI*, 32(9), 2010.

[15] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010.

[16] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[17] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007.

[18] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[19] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, and C. Cao, L. Huang. Large-scale image classification: fast feature extraction and SVM training. In *CVPR*, 2011.

[20] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[21] E. Parzen. On the estimation of a probability density function. *Annals of Mathematical Statistics*, 33(3), 1962.

[22] M. Pawan Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *ICCV*, 2011.

[23] F. Porikli. Computationally efficient histogram extraction for rectangular image regions. In *Proceedings of Real-Time Image Processing*, 2005.

[24] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*, 2008.

[25] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010.

[26] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 23(3):309–314, 2004.

[27] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.

[28] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *ICCV*, 2009.

[29] S. Thrun. Is learning the n-th thing any easier than learning the first? In *NIPS*, 1996.

[30] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010.

[31] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008.

[32] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.

[33] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from Abbey to Zoo. In *CVPR*, 2010.

[34] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 2007.

[3] `http://www.vision.ee.ethz.ch/~calvin/`