Associative embeddings for large-scale knowledge transfer with self-assessment

Alexander Vezhnevets Vittorio Ferrari
The University of Edinburgh
Edinburgh, Scotland, UK

Abstract

We propose a method for knowledge transfer between semantically related classes in ImageNet. By transferring knowledge from the images that have bounding-box annotations to the others, our method is capable of automatically populating ImageNet with many more bounding-boxes.The underlying assumption that objects from semantically related classes look alike is formalized in our novel Associative Embedding (AE) representation. AE recovers the latent low-dimensional space of appearance variations among image windows. The dimensions of AE space tend to correspond to aspects of window appearance (e.g. side view, close up, background). We model the overlap of a window with an object using Gaussian Processes (GP) regression, which spreads annotation smoothly through AE space. The probabilistic nature of GP allows our method to perform self-assessment, i.e. assigning a quality estimate to its own output. It enables trading off the amount of returned annotations for their quality. A large scale experiment on 219 classes and 0.5 million images demonstrates that our method outperforms state-of-the-art methods and baselines for object localization. Using self-assessment we can automatically return bounding-box annotations for 51% of all images with high localization accuracy (i.e. 71% average overlap with ground-truth).

1. Introduction

The large, hierarchical ImageNet dataset [1] presents new challenges and opportunities for computer vision. Although the dataset contains over 14 million images, only a fraction of them has bounding-box annotations (10%) and none have segmentations (object outlines). Automatically populating ImageNet with bounding-boxes or segmentations is a challenging problem, which has recently drawn attention [2, 3]. These annotations could be used as training data for problems such as object class detection [4], tracking [5] and pose estimation [6]. This use case makes it important for these annotations to be of high quality, otherwise they will lead to models fit to their errors. This requires auto-annotation methods to be capable of *self assessment*, i.e. estimating the quality of their own outputs. Self-assessment would allow to return only accurate annotations, automatically discarding the rest.

In this paper we propose a method for transferring knowledge of object appearance from *source* images manually annotated with bounding-boxes to *target* images without them. Source and target images may come from different, but semantically related classes (sec. 2). We model the overlap of an image window with an object using Gaussian Processes (GP) [7] regression. GP are able to model highly non-linear dependencies in a non-parametric way. Thanks to probabilistic nature of GP, we are also able to infer the probability distribution of the overlap value for a given window. This enables taking into account the uncertainty of the prediction. We then tackle the localization problem in a target image by picking a window that has high overlap with an object *with high probability*. The same principle is used for self-assessment.

Doing GP inference directly in a high-dimensional feature space such as HOG [4] is computationally infeasible on a large scale of ImageNet. Moreover, we would have to estimate thousands of hyper-parameters, risking overfitting to the source. Instead, we devise a new representation, coined *Associative Embedding (AE)*, which embeds image windows in a low dimensional space, where dimensions tend to correspond to aspects of object appearance (e.g. front view, close up, background). The AE representation is very compact, and embeds windows originally described by HOG [4] or Bag-of-Words [8] histograms into *just 3 dimensions*. This enables very efficient learning of GP hyper-parameters and inference. It also facilitates knowledge transfer by allowing the source annotations to spread directly along aspects of appearance.

A large scale experiment on a subset of ImageNet, containing 219 classes and 0.5 million images, shows that our method outperforms state-of-the-art competitors [2, 3] and various baselines for object localization.

The remainder of the paper is organized as follows. The next sec. 2 describes the configurations of source and target sets we consider. Sec. 3 formally introduces the setup and gives an overview of our method. Sec. 4 presents Associative Embedding and sec. 5 describes how we estimate

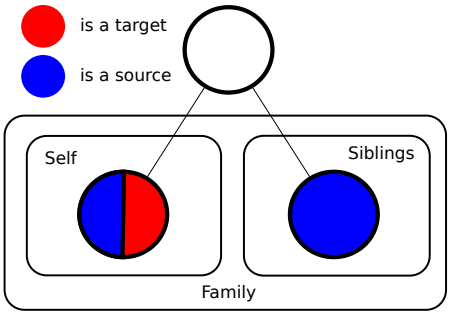


Figure 1. *The considered configurations of* source *and* target *sets. Nodes represent classes in ImageNet, connections correspond to "is a" relationships. Plates correspond to different configurations of possible* sources*. Note that the node of a* target *class on the left is split, as in certain configurations (self and family) it also contributes some annotated images to the* source *set.*

window overlap with Gaussian Processes. Sec. 7 reports some implementation details. Related work is reviewed in sec. 8, followed by experimental results and conclusion in sec. 9.

2. Knowledge sources

The goal of this paper is to localize objects in a target set of images, which do not have manual bounding-box annotations. We tackles this by transferring knowledge from a source set of images that do have them. The target set is composed of all images of a certain target class without bounding-boxes. For some classes, ImageNet offers a rather large subset of images with annotations, so both the source and target sets can come from the same class. In the case when a target class has little or no manual bounding-boxes, we can construct the source set from images of semantically related classes. Different combinations of source and target sets leads to different learning problems. This paper explores the following setups (fig. 1):

1. **Self:** source and target sets of images come from the same class, e.g. from koalas to koalas. This setup is close to a classic object detection problem [9]. Here we have the additional knowledge that every target image contains an object of the class and we are given all target images at training time (*transductive learning*).
2. **Siblings:** the source set consists of images from classes semantically related to the target, e.g. from kangaroos to koalas. This is the setup considered in [3], which can be used to produce annotations even for a target class without any initial manual annotations (*transfer learning*).
3. **Family = self + siblings:** source and target sets consist of images coming from the same mix of semantically related classes, e.g. from kangaroos *and* koalas to kangaroos and koalas. This setup aims at improving performance on related classes when both have some manual annotations by processing them simultaneously (*multitask learning*).

These source/target setups cover a broad range of knowledge transfer scenarios. Below we devise a model that covers all these scenarios in unified manner. We expect an adequate model to have an increasing performance as more supervision is being provided (with **siblings** having the worst and **family** having the best performance).

3. Overview of our method

In this section we define the notation and introduce our method on a high level, providing the details for each part later (Sec. 4,5,6). Every image I, in both source and target sets, is decomposed into a set of windows $\{w_i\}$. To avoid considering every possible image window, we use the *objectness* technique [10] to sample 1000 windows per image. As shown in [10], these are enough to cover the vast majority of objects. Source windows are annotated with the overlap $Y(w)$, defined as their intersection-over-union [9] (IoU) with the ground-truth bounding-box of an object.

The key idea of our method is to use GP regression to transfer overlap annotations from source windows to target ones. GP infers a probability distribution over the overlap of a target window with the object: $P(Y(w^t) = y|w^t)$ [1]. Having a full distribution enables to measure and control uncertainty. Consider the score function[2]:

$$\eta(w, \lambda) \equiv \max_{\xi \in [0,1]} \xi \quad \texttt{s.t. } P(Y(w) \geq \xi | w) \geq \lambda. \qquad (1)$$

This score is the largest overlap $Y(w)$ that a window w is predicted to have with at least λ probability. The higher the λ, the more certain we need to be before assigning a window a high score.

Performing GP inference in the high-dimensional space typical for common visual descriptors (e.g. HOG, bag-of-words) is computationally infeasible at the large scale of ImageNet. Moreover, it would require estimating thousands of hyper-parameters, risking overfitting. To reduce the dimensionality of the feature space without losing descriptive power, we introduce Associative Embedding (AE) ψ, a low-dimensional representation $\dim(\psi(x)) << \dim(x)$ of windows (sec. 4). Let x_o be the representation of an object *exemplar* (ground-truth bounding-box) from the source set in some feature space (e.g. HOG). Let x_i the representation of a window w_i in the same feature space. First we describe w_i in terms of its similarity to the exemplars $\{x_o\}_{o=1}^{N_o}$ in the source set. We quantify how similar the window x_i is to the exemplar x_o by the output of an Exemplar-SVM (E-SVM) [11] with parameters θ_o trained on x_o. Now every window is described by a vector of E-SVM outputs $[x_i\theta_1^T, ..., x_i\theta_{N_o}^T]$, one for each exemplar in the source set. Intuitively, this intermediate representation describes what a window *looks like*, rather than *how it looks*. The advantage of using E-SVMs over a standard similarity measure

[1] We slightly abuse notation here, dropping the conditioning on source data: $P(Y(w^t) = y|w^t, \{Y(w_s) = y_s\}_{s \in \texttt{source}})$

[2] Corrected thanks to O. Russakovsky

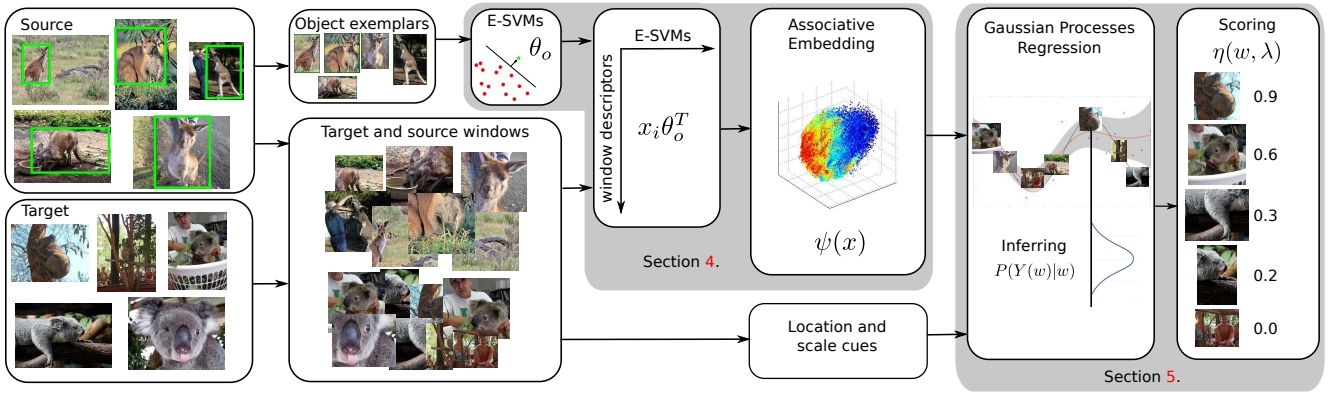


Figure 2. *Illustration of our pipeline. First, E-SVMs are trained for each object exemplar in the source set. Next, these E-SVMs are used to represent all* windows *in both source and target sets, which are then embedded into a low dimensional space (coined Associative Embedding, AE). After adding location and scale cues to the AE representation of windows, GP regression is used to transfer overlap annotation from source windows to target ones. The final score assigned to a target window integrates the mean predicted overlap and the uncertainty of the prediction.*

is its ability to suppress the influence of the background present in the window and to focus on exemplar-specific features [11].

Next, we embed all windows and exemplars into a low dimensional space. We optimize the embedding to minimize the difference between scalar products in the original and embedded spaces. We call the resulting space *Associative Embedding* (AE). The dimensions in EA space tend to correspond to aspects of the appearance of classes (e.g. close up, side view, partially occluded). The AE space enables the GP to transfer knowledge directly along the aspects which greatly facilitates the process. For HOG and Bag-of-Words descriptors, an AE space with *just 3 dimensions* is sufficient to support this process.

Fig. 2 depicts our method procedural flow. Having embedded source and target windows into AE space, we use GP to infer probability distributions of target windows overlap $P(Y(w^t) = y|w^t)$ and to compute the score $\eta(w^t, \lambda)$ (eq. (1)). We employ this score to tackle localization task in sec. 6. For localization we simply return the highest scored window in each target image. We also reuse the score for self-assessment.

4. Associative embedding

This section details the proposed AE representation (fig. 4). The method proceeds by first training E-SVMs for object exemplars in the source set, then representing all windows from both source and target sets as the output of E-SVMs applied to their feature descriptors, and finally embedding these representations in a lower dimensional space.

Training E-SVMs. For each object exemplar x_o in a given feature space, we train an E-SVM [11] hyperplane θ_o to minimize the following loss

$$\theta_o = \arg\min_{\theta} ||\theta||^2 + C_1 h(x_o\theta^T) - C_2 \sum_{x_i \in \mathcal{N}} h(x_i\theta^T), \tag{2}$$

where h is the hinge loss $h(x) = \max(0, 1 - x)$ and $\mathcal{N}$ is a set of negative windows from the source pool. It contains a few thousand windows with overlap smaller than 0.5 with the object's bounding-box. The weights C_1 and C_2 regulate the relative importance of the terms.

Encoding windows as E-SVM outputs. We encode every window w_i in both the source and target sets as a vector a_i where each element $a_{io} = x_i\theta_o^T$ is the response of an E-SVM θ_o. This representation is much richer than the original feature space, as each entry in a_i encodes how much this window looks like one of the exemplars.

Embedding. Let A be a matrix with a row a_i for each window in both the source and target sets. Let u_i and v_o be the embedded representations of x_i and θ_o, which we are trying to produce. Let matrices U and V have u_i and v_o as rows, respectively. We seek the embedding that minimizes the reconstruction error:

$$\min_{U,V} ||UV^T - A||_F^2 \tag{3}$$

where $||\cdot||_F$ is the Frobenius norm. This can be done by a truncated SVD decomposition [12]: $A = U\Sigma\hat{V}^T$. We encapsulate singular values into the E-SVM representation: $V = \Sigma\hat{V}^T$. Elements of a common visual descriptor (e.g. HOG) correspond to local statistics of an image window. However, each element of a_i correspond to the output of a battery of E-SVMs on the same image window. These are individually much more informative, yet their collection is redundant, enabling compression to a few dimensions. For instance, the representation a_i of a background window will contain similar negative values across all the E-SVMs. Moreover, E-SVMs that are learnt from similar exemplars coming from the same aspect of object's appearance (e.g., gorilla's face, close up) will produce correlated outputs uniformly for all windows. Altogether, this dramatically collapses a_i variability, which allows SVD to achieve

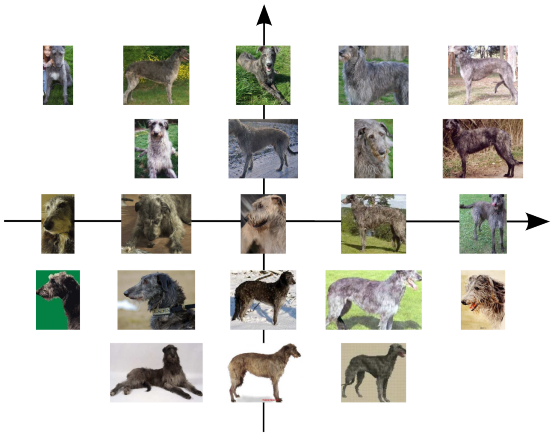


Figure 3. *A visualization of AE for the class 'Scottish Deerhound'. We show some source windows with high overlap with the ground-truth bounding-box. The position corresponds to the 2nd and 3rd coordinates in their AE representation. Note how windows with similar appearance cluster together: the top right corner corresponds to facing-left side views, the center to face close ups, and the bottom to facing-right side views. This shows how AE manages to recover the underlying appearance variation dimensions.*

low reconstruction error with just a few dimensions, which tend to correspond to object/background discrimination and the aspects of object appearance (fig. 3).

Solving the optimization problem for all windows in the source and target sets at once is computationally very expensive. Therefore, in a first step we use only a sub-sample of the windows, but all exemplars, to minimize (3). This results in an embedded representation of all exemplars V. In a second step we keep V fixed and embed any remaining window x_i with

$$\psi(x_i) = \arg\min_{u} \sum_{o \in \text{source}} ||x_i \theta_o^T - u v_o^T||^2 \qquad (4)$$

We solve the above optimization using least-squares. Fig. 5(a) depicts the embedding of Koala class SURF Bag-of-Words window descriptors in AE space.

5. Estimating window overlap with Gaussian Processes

This section describes how to infer a probability distribution $P(Y(w^t)|w^t)$ of the overlap of a target window w^t with an object and calculate the final score (eq. 1). We first construct an extended representation of a window using AE, its objectness score [13] and its position and scale in the image. Then we define a GP over that representation and estimate its hyper-parameters. We also show how to speed up inference at prediction time in sec. 5.2.

5.1. GP construction

Let x^{SURF} and x^{HOG} be the SURF bag-of-words and HOG appearance descriptors of a window w, and

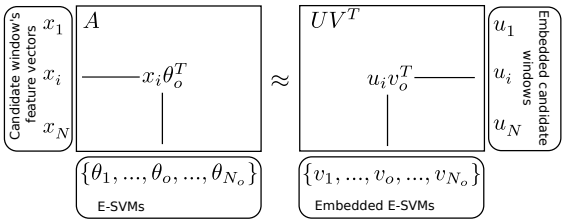


Figure 4. *Illustration of Associative Embedding (AE). A matrix of E-SVM outputs A (left) on windows from both source and target sets is factorised into UV^T (right).*

$\psi(x^{\text{SURF}}), \psi(x^{\text{HOG}})$ their AE. The final representation $\phi(w)$ of w is

$$\phi(w) = [c(w), \text{Obj}(w), \psi(x^{\text{SURF}}), \psi(x^{\text{HOG}})], \qquad (5)$$

where $c(w)$ is a position and scale descriptor of a window. Following [3] we define it as $c(w) = [c_1, c_2, \log(WH), \log(W/H)]$. Here c_1 and c_2 are the coordinates of the center of the window, W and H are width and height of the window (all normalized by the image size). $\text{Obj}(w)$ is the objectness score of w, which estimates how likely it is to contain an object rather than background [10].

We propose to model the overlap $Y(w)$ of a window w with a true object bounding-box as a Gaussian Process (GP)

$$Y(w) \sim \mathcal{GP}\left(m(\phi(w)), k(\phi(w), \phi(w'))\right) \qquad (6)$$

where $m(\phi(w))$ is a mean function and $k(\phi(w), \phi(w'))$ is a covariance function. Here the mean function is zero $m(\phi(w)) = 0$ and the covariance (kernel) $k(\phi(w), \phi(w'))$ is the squared exponential

$$\exp\left(-\frac{1}{2}(\phi(w) - \phi(w'))^T \text{diag}(\gamma)^{-2}(\phi(w) - \phi(w'))\right) \qquad (7)$$

where γ is a vector of hyper-parameters regulating the influence of each element of $\phi(w)$ on the output of the kernel.

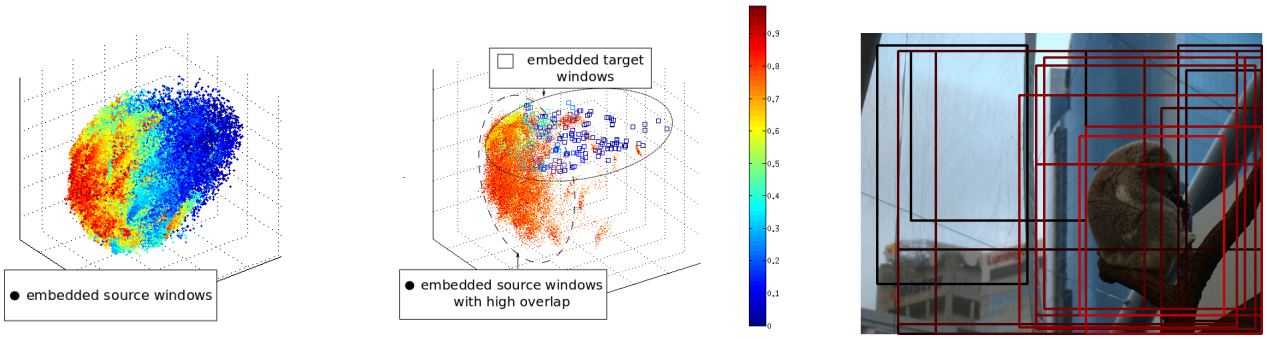
Let $\{w_s, y_s\}_s^{N_s}$ be a set of windows from the source set and their respective ground-truth overlaps $y_s = Y(w_s)$ (*inducing points*). Let K be a kernel matrix $K_{s,s'} = k(\phi(w_s), \phi(w_{s'}))$. Consider now a target window w^t, for which $Y(w^t)$ is unknown. Let $\boldsymbol{k} = [k(w^t, w_1), k(w^t, w_2), ..., k(w^t, w_{N_s})]$. Then a predictive distribution for w^t is

$$P(Y(w^t)|w^t) = \mathcal{N}\left(\mu(w^t), \sigma^2(w^t)\right) \qquad (8)$$

where

$$\mu(w^t) = \boldsymbol{k}^T K^{-1} \boldsymbol{y}_{1:N}$$
$$\sigma^2(w^t) = k(w^t, w^t) - \boldsymbol{k}^T K^{-1} \boldsymbol{k} \qquad (9)$$

In practice, inference involves the inversion of kernel matrix K and a series of scalar product. Matrix inversion for a large set of inducing points poses a computational problem, with which we deal with in sec. 5.2.



(a) (b) (c)

Figure 5. *Illustration of AE of windows of the Koala class, using the SURF Bag-of-Words descriptor. (a) the embedded source windows. Points correspond to windows and their colour to the overlap with the ground-truth objects in their respective images. Note how source windows form a comet shape with background windows in the center and the tail of the comet. (b) the embedded target windows of an image (squares) together with source windows (dots) with high overlap with the ground-truth. (c) a target image with a few randomly sampled windows. Colour intensity corresponds to the overal score $\eta(w, 0.5)$ output by our method.*

Overlap estimation. We can now compute the score $\eta(w, \lambda)$ from eq. (1).

$$\eta(w, \lambda) = \mu(w) + \beta(\lambda)\sigma(w) \tag{10}$$

where β is a constant which depends on λ and can be analytically computed by integrating the Gaussian density. The score $\eta(w, \lambda)$ equals to the maximum overlap that window w may have with at least λ probability.

Estimating hyper-parameters. We estimate the hyper-parameters γ by minimizing the regularized negative log-likelihood of the overlaps of the windows in the source set

$$-\log P\left(\mathcal{D}(w)|\phi(w), \gamma\right) + \alpha||\gamma||^2, \tag{11}$$

where α is the regularization strength. While it is not common practice to put a regularizer on γ, we experimentally found that it significantly improves the result.

5.2. Fast inference for large-scale datasets

The source set can contain millions of windows, which poses a computational problem for standard GP inference methods [7]. Instead of exact inference we use an approximation [3] during both training and test. For training hyper-parameters we also sub-sample the windows from the source set. Since the dimensionality of γ is very low (11 in our case), it can be reliably estimated from modest amounts of data.

To speed up prediction on windows in a target image, as inducing points of GP we only use windows from source images that have a similar global appearance. The idea is that globally similar images are more likely to contain objects and backgrounds related to those in the target image, and therefore are the most relevant source for transfer. This is related to ideas proposed before for scene parsing [15, 16]. However, while those works used a simple monolithic global image descriptor (bag-of-words or GIST) for this task, we directly use the set of window descriptors $\phi(w)$ to construct the global similarity measure. In the feature space spanned by ϕ, every image has an empirical distribution, i.e. a cloud of points corresponding to the windows it contains (fig. 5(b)). We use per-coordinate kernel density estimation to represent the cloud of an image. The global image similarity between a source and a target image is then defined as the average per-coordinate KL-divergence between their distributions.

All the modifications above reduces full inference for all windows in a test image to approximately 4 seconds on a standard desktop computer. Hence our method is suitable for large scale datasets like ImageNet.

6. Object localization

The technique presented above produces a score $\eta(w, \lambda)$ for each target window in a target image $w \in I^t$. This section explains how to use this score for object localization. We simply select the window with the highest score $w^* = \arg\max_{w \in I^t} \eta(w, \lambda)$, out of all windows in the target image. This window is the final output of the system, which returns one window for each target image.

The score can be also be used for self-assessment. For example, we can retrieve all bounding-boxes that have overlap higher than 60% with 0.8 probability by taking only the windows such that $\eta(w, 0.8) > 60$.

7. Implementation details

Associative Embedding. We use AE in SURF [17] bag-of-words [8] and HOG feature spaces. We build quantized SURF histograms with a codebook of 2000 visual words. For computational efficiency, we approximate the χ^2 kernel with the expansion technique of [18] and train linear E-SVMs. HOG descriptors [4] are computed over a 8×8 grid and the associated E-SVMs are linear, as in [11]. The soft margin parameters C_1 and C_2 are set to $C_1 = 1$ and

[3] infFITC from the toolbox [14]

$C_2 = 0.001$. We create AE of a dimensionality 3 (for each of HOG and SURF bag-of-words).

Gaussian Processes. To learn GP hyper-parameters (sec. 5) we sub-sample 15000 windows from the source set. Since the dimensionality of GP hyper-parameters is low ($\dim(\gamma) = 11$ in our experiments), this is sufficient. We set the regularizer to $\alpha = 100$ in eq. (10). For the scoring function $\eta(w, \lambda)$ we set $\lambda = 0.8$ for object localization, and $\lambda = 0.5$ for self-assessment.

For prediction on a target image, we retrieve the 300 most similar source images as described in sec. 5.2, and then infer the distribution of windows overlap using them.

8. Related work

Populating ImageNet. The two most related works are [2, 3]. Like us, they address the problem of populating ImageNet with automatic annotations (bounding-boxes [3] and segmentations [2]). Guillaumin and Ferrari (GF) [3] train a series of monolithic window classifiers, one for each source class, using different cues (HOG, colour, location, etc.). They are combined into a final window score by a discriminatively trained weighting function and applied to windows in the target set. As we show in sec. 9, GF is not suited for self assessment. Moreover, our method produces better localizations overall.

The work [2] populates ImageNet with segmentations. It propagates ground-truth segmentations from PASCAL VOC [9] onto ImageNet. They use a nearest neighbour technique [19] to transfer segmentations from a given source set to a target image. We compare to this method experimentally (sec. 9), by putting a bounding-box over their segmentations.

Transfer learning is used in computer vision to facilitate learning a new target class with the help of labelled examples from related source classes. Transfer is typically done through regularization of model parameters [20, 21], an intermediate attribute layer [22] (e.g. yellow, furry), or by sharing parts [23]. In GP [24] transfer learning is usually based on sharing hyper-parameters between tasks. In this work we not only share hyper-parameters, but the *inducing points* as well. Also, our GP kernel is defined over an augmented AE space $\phi(x)$, which is constructed specifically for a particular combination of source and target classes. In principle, one could view AE as kernel learning method for GP, which exploits the specifics of visual data.

Exemplar SVMs were first proposed by [11] and are rapidly gaining popularity as a better way to measure visual similarity [25, 26, 27]. Their main advantage is the ability to select features within a window that are relevant to the object, down-weighting background clutter. Some authors propose to use [26, 27] E-SVMs as a similarity measure for discovering different aspects of object appearance. They explicitly group training objects into clusters according to their aspects of appearance. They produce a set of clean

Figure 6. *Localization results for our AE-GP+ method.*

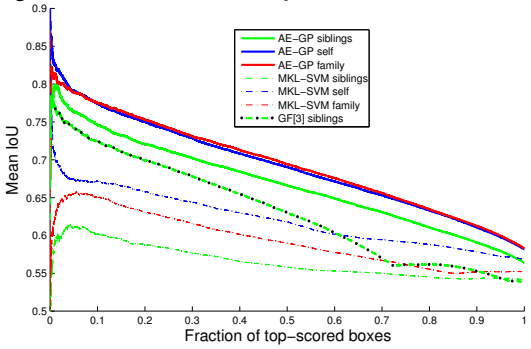


Figure 7. *Self-assessment performance curves for our method (AE-GP), GF [3], and MKL-SVM.*

aspect-specific object detectors, whose responses on a test image are merged together for the final result. Instead we embed image windows into a low dimensional space, where aspects of appearance are expressed smoothly in the space's dimensions.

9. Experiments and conclusion

We perform experiments on the same subset of ImageNet as [2, 3], which allows direct comparison to their results. The subset consists of 219 classes (e.g. phalanger, beagle, etc.) spanning over 0.5 million images. Classes are selected such that each has less than 10.000 images and has siblings with some manual bounding-box annotations [3]. For a given target class we consider the following source sets: i) **self** — the class itself; ii) **siblings** of the class (as in [3]); iii) **family** — the class itself plus its siblings (see sec. 2 and fig. 1).

Ground-truth. We use 92K images with ground-truth bounding-boxes in total. We split them in two disjoint sets of 60K and 32K. We use the first set exclusively as source and the second one exclusively as target. This allows us to compare transfer from different source types (siblings, self, family) on exactly the same target images. The first batch of 60K images is the same as used in [3]. This ensures proper comparison to [3], as when using siblings as source, our method transfers knowledge from exactly the same images as [3].

Baselines and [3]. For localization, we compare against the following. **MidWindow:** A window in the center of the image, occupying 50% of its area. **TopObj:** The window with the highest objectness score [10]. **MKL-SVM:** This represents a standard, discriminative approach to object localization, similar to [28]. On the source set we train a SVM on HOG (linear) and a SVM with SURF bag-of-words (linear with χ^2 expansion [18]) using 90% of the data. To combine them, we train a linear SVM over their outputs using the 10% holdout data. We improve the baseline by adding the objectness score, location and scale cues (as in sec. 5) of a window as features for the second-level SVM. **GF:** We compare to [3], when using siblings as the source. We use the output of [3] as provided by the authors on their website. Notice how MKL-SVM baseline and the competitor GF [3] are defined on the same object proposals [10] and features as our AE-GP. This ensures that any improvement comes from better modelling.

Metrics. To measure the quality of localizations we use the *intersection-over-union* criterion (IoU) as defined in the PASCAL VOC [9]. We also measure *detection rate*: the percentage of images where the output has IoU > 0.5.

To measure the quality of self-assessment we evaluate how well the score (1) ranks the output bounding-boxes. We sort the outputs by their scores and measure the mean IoU of the top $p\%$ outputs. To compare to [3] we use their scores released along with their output bounding-boxes. For MKL-SVM we use the score output by the second-level SVM.

Comparison to baselines and [3]. Table 1 summarizes localization results, averaged over all 32K target images for which we have evaluation ground-truth. The results of our method steadily improve as the source set changes from siblings to self to family, unlike the MKL-SVM baseline, whose performance decreases from self to family. This behaviour shows that our method is applicable to a wide range of knowledge transfer scenarios. Using only siblings as source, we outperform GF by the same margin as GF outperforms the trivial baselines TopObj and MidWindow in terms of mean IoU. Overall, our method delivers the best results for all kinds of source sets and metrics, compared to both competitors and baselines. While GF [3] sometimes produces a bounding-box occupying most of an image, our localizations are typically more specific to the object.

Fig. 7 presents self-assessment curves. Our method nicely trades off the amount of returned localizations for their quality, as demonstrated by the visible slope of the solid curves. For all sources, our method outperforms MKL-SVM and GF over the entire range of the curve. The advantage over MKL-SVM is greater especially in the left part of the curves, where self-assessment plays a bigger role. Note how both MKL-SVM and GF have cusps in their curves. This means that many high quality localizations get a low score (GF, right half of the curve) or some high scoring localizations are poor (MKL-SVM, left half). Interestingly, our MKL-SVM baseline performs similarly to GF when evaluated on all images (tab. 1), but GF is better at self-assessment (fig. 7).

Method	Source	IoU%	IoU at 50%	Detection%
AE+GP	Siblings	**56.4**	**66.6**	**63.5**
GF [3]	Siblings	53.7	63	58.5
MKL-SVM	Siblings	54.1	55.8	59.7
AE+GP	Self	**58.2**	**69**	**66.5**
AE+GP+	Self	**59.9**	**71**	**68.3**
MKL-SVM	Self	56.7	61.8	63.8
AE+GP	Family	**58.3**	**69.4**	**66.7**
MKL-SVM	Family	55.1	58.9	60.8
TobObj	-	50.1	55.3	52.7
MidWindow	-	49.3	-	60.7
KGF [2]	-	59.9	-	66.9

Table 1. *Localization results: mean IoU, mean IoU of the top* 50% *ranked outputs, and detection rate.*

Analysis of AE. We validate here the importance of AE and whether it can reduce dimensionality without loss in representation power (fig. 8).

In **PCA-GP** we substitute AE with a standard PCA in each original feature space (HOG, SURF bag-of-words), keeping the rest of the method exactly the same. PCA-GP performs significantly worse than AE-GP, highlighting the importance and power of AE.

AE-GP-d100 increases the dimensionality of AE to 100 for each feature space. This drastic increase of dimensionality makes little difference in the results. This shows that AE does not lose much representation power even when reducing the dimensionality to just 3.

Better features and proposals. The experiments above demonstrate that our AE-GP outperforms baselines and [3], given the same features and object proposals. To push performance even further we introduced here a version of our method, coined **AE-GP+**, that uses state-of-the-art features [29] and object proposals [30]. To accommodate very high dimensional features [29] we slightly increase the dimensionality of AE to 10. We use the initial features (sec. 7) as well, improving SURF bag-of-words by adding spatial binning. Results in fig. 8 and tab. 1 show that AE-GP+ delivers the best results, improving over AE-GP by 1.8% in detection rate. Fig. 6 demonstrates localizations by AE-GP+.

Comparison to [2]. We compare here to the state-of-the-art segmentation method KGF [2] by putting a bounding-box around their output segmentation. AE-GP+ moderately outperforms KGF [2] by 1.4% in detection rate (tab. 1). Most importantly, as KGF [2] assigns no score to its output, self-assessment is impossible. The user can't *automatically* retrieve high quality localizations from KGF [2]. In contrast, AE-GP+ has the ability to select the localizations that have high IoU. Also, unlike the scoring schemes in GF and MKL-SVM, ours allows the user to retrieve localizations that are *predicted* to have an overlap higher than, say, 60% with a 0.5 probability. Using AE-GP+ with self as source,

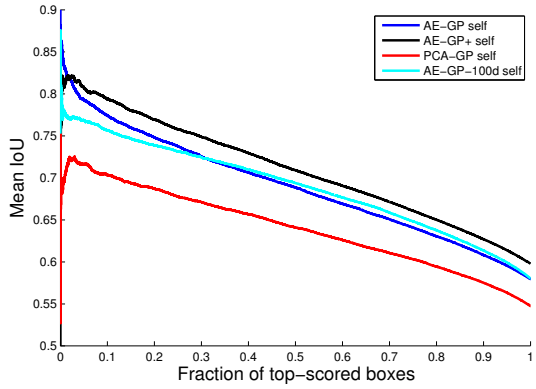


Figure 8. *Self-assessment performance curves for AE-GP versions.*

this fully automatically returns 51% of all localizations with a mean IoU of 71% (which is very accurate, c.f. the PASCAL detection criterion is IoU> 0.5). This means about 251K images in the dataset we processed! The results are released online [4].

Conclusion. Knowledge transfer in ImageNet is motivated by the fact that semantically related objects *look similar* (e.g. police car and taxi). Hence, in a latent space of appearance variation image windows that contain objects of related classes are close to each other, while background windows are far away from them. Our work formalizes this intuition with the AE+GP method. AE recovers the latent space of appearance variation and embeds windows into it. Next, we construct a GP over the AE space to transfer localization annotations from the source to target image set. Thanks to probabilistic nature of GP, our model is capable of self-assessment. Large-scale experiments demonstrate that our method outperforms state-of-the-art techniques [2, 3] for populating ImageNet with bounding-boxes and segmentations, as well as a strong MKL-SVM baseline defined on the same features.

Acknowledgements This work was supported by ERC VisCul starting grant. A. Vezhnevets is also supported by SNSF fellowship PBEZP-2142889.

References

[1] 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.

[2] M. Guillaumin, D. Kuettel, and V. Ferrari, "ImageNet Auto-annotation with Segmentation Propagation," *IJCV*, 2014, to appear.

[3] M. Guillaumin and V. Ferrari, "Large-scale knowledge transfer for object localization in ImageNet," in *CVPR*, Jun 2012.

[4] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for human detection," in *CVPR*, 2005.

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

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

[4] http://groups.inf.ed.ac.uk/calvin/proj-imagenet/page/

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[8] 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.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.

[10] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. on PAMI*, 2012.

[11] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, 2011.

[12] S. C. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, 1990.

[13] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *CVPR*, 2010.

[14] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *The Journal of Machine Learning Research*, 2010.

[15] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *CVPR*, 2009.

[16] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *ECCV*, 2010.

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

[18] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *CVPR*, 2010.

[19] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," in *CVPR*, 2012.

[20] 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," *CVIU*, 2007.

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

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

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

[24] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," 2008.

[25] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem, "Learning collections of part models for object recognition," in *CVPR*, 2013.

[26] O. Aghazadeh, H. Azizpour, J. Sullivan, and S. Carlsson, "Mixture component identification and learning for visual recognition," in *ECCV*, Springer, 2012.

[27] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan, "Subcategory-aware object classification," in *CVPR*, 2013.

[28] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV*, 2009.

[29] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, 2013.

[30] S. Manén, M. Guillaumin, and L. Van Gool, "Prime Object Proposals with Randomized Prim's Algorithm," in *ICCV*, 2013.