Object localization in ImageNet by looking out of the window

Alexander Vezhnevets
vezhnick@gmail.com
Vittorio Ferrari
vferrari@gmail.com
University of Edinburgh
Edinburgh, Scotland, UK

Abstract

We propose a method for annotating the location of objects in ImageNet. Traditionally, this is cast as an image window classification problem, where each window is considered independently and scored based on its appearance alone. Instead, we propose a method which scores each candidate window in the context of all other windows in the image, taking into account their similarity in appearance space as well as their spatial relations in the image plane. We devise a fast and exact procedure to optimize our scoring function over all candidate windows in an image, and we learn its parameters using structured output regression. We demonstrate on 92000 images from ImageNet that this significantly improves localization over recent techniques that score windows in isolation [32, 35].

1 Introduction

The ImageNet database [9] contains over 14 million images annotated by the class label of the main object they contain. However, only a fraction of them have bounding-box annotations (10%). Automatically annotating object locations in ImageNet is a challenging problem, which has recently drawn attention [15, 16, 35]. These annotations could be used as training data for problems such as object class detection [8], tracking [21] and pose estimation [4]. Traditionally, object localization is cast as an image window scoring problem, where a scoring function is trained on images with bounding-boxes and applied to ones without. The image is first decomposed into candidate windows, typically by object proposal generation [3, 12, 23, 37]. Each window is then scored by a classifier trained to discriminate instances of the class from other windows [8, 14, 15, 17, 32, 33, 36] or a regressor trained to predict their overlap with the object [6, 20, 34, 35]. Highly scored windows are finally deemed to contain the object. In this paradigm, the classifier looks at one window at a time, making a decision based only on that window's appearance.

We believe there is more information in the collection of windows in an image. By taking into account the appearance of all windows *at the same time* and connecting it to their spatial relations in the image plane, we could go beyond what can be done by looking at one window at a time. Consider the baseball in fig. 1(a). For a traditional method to succeed, the appearance classifier needs to score the window on the baseball higher than the windows containing it. The container windows cannot help except by scoring lower and be discarded. By considering one window at a time with a classifier that only tries to predict whether it

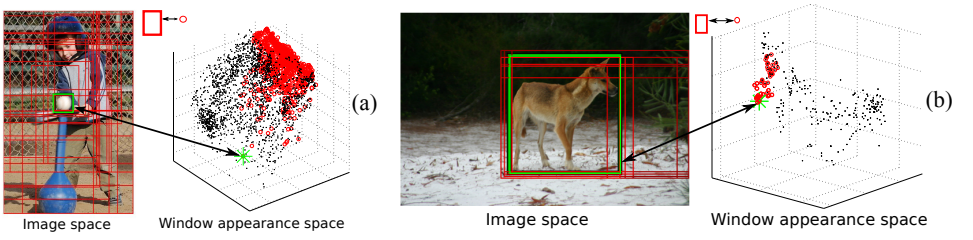

Figure 1: **Connecting the appearance and window position spaces.** *(a) a window tight on the baseball (green star in the appearance space plot) and some larger windows containing it (red circles in the appearance space). Black points in appearance space represent all other candidate windows. (b) all windows with high overlap with the wolf are shown in red, both in the image and in appearance space. The ground-truth bounding-box to be found is shown in green. The appearance space plots are actual datapoints, representing windows in 3-dimensional Associative Embedding of SURF bag-of-words. Please see main text for discussion.*

covers the object tightly, one cannot do much more than that. The first key element of our work is to predict richer spatial relations between each candidate window and the object to be detected, including part and container relations. The second key element is to employ these predictions to reason about relations between different windows. In this example, the container windows are predicted to contain a smaller target object somewhere inside them, and thereby actively help by *reinforcing* the score of the baseball window. Fig. 1(b) illustrates another example of the benefits of analyzing all windows jointly. Several windows which have high overlap with each other and with the wolf form a dense cluster in appearance space, making it hard to discriminate the precise bounding-box by its appearance alone. However, the precise bounding-box is positioned at an extreme point of the cluster — on the tip. By considering the configuration of all the windows in appearance space together we can reinforce its score.

In a nutshell, we propose to localize objects in ImageNet by scoring each candidate window in the context of all other windows in the image, taking into account their similarity in appearance space as well as their spatial relations in the image plane. To represent spatial relations of windows we propose a descriptor indicative of the part/container relationship of the two windows and of how well aligned they are (sec. 2). We learn a windows appearance similarity kernel using the recent Associative Embedding technique [35] (sec. 3). We describe each window with a set of hyper-features connecting the appearance similarity and spatial relations of that window to all other windows in the same image. These hyper-features are indicative of the object's presence when the appearance of a window alone is not enough (e.g. fig 1). These hyper-features are then linearly combined into an overall scoring function (sec. 4). We devise a fast and exact procedure to optimize our scoring function over all candidate windows in a test image (sec. 4.1), and we learn its parameters using structured output regression [31] (sec. 4.2).

We evaluate our method on a subset of ImageNet containing 219 classes with more than 92000 images [15, 16, 35]. The experiments show that our method outperforms a recent approach for this task [35], an MKL-SVM baseline [33] based on the same features, and the popular UVA object detector [32]. The remainder of the paper is organized as follows. Sec. 2 and 3 introduce the spatial relation descriptors which we use in sec. 4 to define our new localization model. In sec. 5 we review related work. Experiments and conclusions are presented in sec. 6.

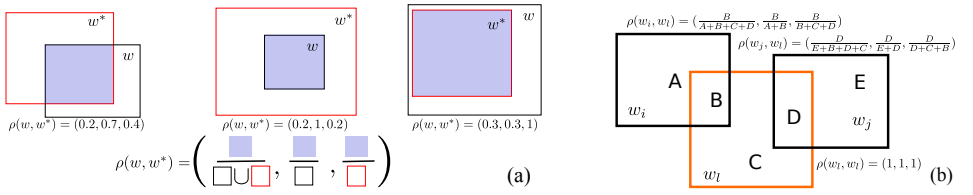

Figure 2: **Spatial relations $\rho(w,w')$ between two windows.** *(a) The first element indicate how much w and w' overlap, in the traditional PASCAL VOC sense [13]. The second element indicates whether window w is a part of w'. The third element measures whether w is a container of w'. (b) Computing the spatial relations $\rho(w_i,w_l)$ and $\rho(w_j,w_l)$ for hyper-features ϕ_G (3) and ϕ_L (4) .*

2 Describing spatial relations between windows

Candidate windows. Recently, object class detectors are moving away from the sliding-window paradigm and operate instead on a relatively small collection of *candidate windows* [3, 12, 14, 15, 23, 32, 35, 36] (also called 'object proposals'). The candidate window generators are designed to respond to objects of any class, and typically just $1000-2000$ candidates are sufficient to cover all objects in a cluttered image [3, 12, 23, 32]. Given a test image, the object localization task is then to select a candidate window covering an instance of a particular class (e.g. cars). Following this trend, we generate about 1000 candidate windows $W = \{w\}_{i=1}^{N}$ using the recent method [23].

Spatial relation descriptor. We introduce here a representation of the spatial relations between two windows w and w', which we later use in our localization model (sec. 4). We summarize the spatial relation between windows w and w' using the following *spatial relation descriptor* (fig. 2(a))

$$\rho(w,w') = \left(\frac{w \cap w'}{w \cup w'}, \frac{w \cap w'}{w}, \frac{w \cap w'}{w'} \right) \tag{1}$$

where the $\cap$ operator indicates the area of the intersection between the windows, and $\cup$ the area of their union. The descriptor captures three different kinds of spatial relations. The first is the familiar intersection-over-union (*overlap*), which is often used to quantify the accuracy of an object detector [1, 13]. It is 1 when $w = w'$, and decays rapidly with the misalignment between the two windows. The second relation measures how much of w is contained inside w'. It is high when w is a *part* of w', e.g. when w' is a car and w is a wheel. The third relation measures how much of w' is contained inside w. It is high when w *contains* w', e.g. w' is a snooker ball and w is a snooker table. All three relations are 0 if w and w' are disjoint and are 1 if w and w' match perfectly. Hence the descriptor is indicative for part/container relationships of the two windows and of how well aligned they are.

Vector field of window relations. Relative to a particular candidate window w_l, we can compute the spatial relation descriptor to any window w. This induces a vector field $\rho(\cdot,w_l)$ over the continuous space of all possible window positions. We observe the field only at the discrete set of candidate windows W. A key element of our work is to connect this field of spatial relations to measurements of appearance similarity between windows. This connection between position and appearance spaces drives the new components in our localization model (sec. 4).

3 Predicting spatial relations with the object

A particularly interesting case is when w' is the true bounding-box of an object w^*. For the images in the training set, we know the spatial relations $\rho(w,w^*)$ between all candidate windows w and the bounding-box w^*. We can use them to learn to predict the spatial relation between candidate windows and the object from window appearance features x in a test image, where ground-truth bounding-boxes are not given.

Following [35], we use Gaussian Processes regression (GP) [27] to learn to predict a probability distribution $P(\rho^r(w,w^*)|x) \sim \mathcal{GP}(m(x),K(x,x'))$ for each spatial relation $r \in \{\texttt{overlap},\texttt{part},\texttt{cont}\}$ given window appearance features x. We use zero mean $m(x)=0$ and learn the kernel (covariance function) $K(x,x')$ as in [35]. This kernel plays the role of an appearance similarity measure between two windows. The GP learns kernel parameters so that the resulting appearance similarity correlates with the spatial relation to be predicted, i.e. so that two windows which have high kernel value also have a similar spatial relation to the ground-truth. We will use the learnt $K(x,x')$ later in sec. 4.

For a window w_i in a test image, the GP predicts a Gaussian distribution for each relation descriptors. We denote the means of these predictive distributions as $\mu(x_i) = (\mu^{\texttt{overlap}}(x_i), \mu^{\texttt{part}}(x_i), \mu^{\texttt{cont}}(x_i))$, and their standard deviation as $\sigma(x_i)$. The standard deviation is the same for all relations, as we use the same kernel and inducing points.

4 Object localization with spatial relations

We are given a test image with (a) set of candidate windows $W = \{w_i\}_{i=1}^N$; (b) their appearance features $X = \{x_i\}_{i=1}^N$; (c) the mean $M = \{\mu(x_i)\}_{i=1}^N$ and standard deviation $\Sigma = \{\sigma(x_i)\}_{i=1}^N$ of their spatial relations with the object bounding-box, as predicted by the GP; (d) the appearance similarity kernel $K(x_i,x_j)$ (sec. 3).

Let $w_l \in W$ be a candidate window to be scored. We proceed by defining a set of hyper-features $\Phi(X,W,M,l)$ characterizing w_l, and then define our scoring function through them.

Consistency of predicted & induced spatial relations ϕ_C

$$\phi_C^r(X,W,l) = \max_i |\rho^r(w_i,w_l) - \mu^r(x_i)| \tag{2}$$

Assume for a moment that w_l correctly localizes an instance of the object class. Selecting w_l would induce spatial relations $\rho^r(w_i,w_l)$ to all other windows w_i. The hyper-feature ϕ_C checks whether these induced spatial relations are consistent with those predicted by GP based on the appearance of the other windows ($\mu^r(x_i)$). If so, that is a good sign that w_l is indeed the correct location of the object. More precisely, the hyper-feature measures the disagreement between the induced $\rho^r(w_i,w_l)$ and predicted $\mu^r(x_i)$ on the window w_i with the largest disagreement. Fig. 3 illustrates it on a toy example. The maximum disagreement, instead of a more intuitive mean, is less influenced by disagreement over background windows, which are usually predicted by GP to have small, but non-zero relations to the object. It focuses better on the alignment of the peaks of the predicted $\{\mu^r(x_i)\}_{i=1}^N$ and observed $\{\rho^r(w_i,w_l)\}_{i=1}^N$ measurements of the vector field $\rho^r(\cdot,w_l)$, which is more indicative of w_l being a correct localization.

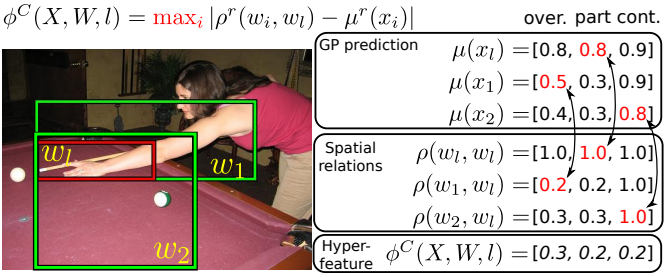

Figure 3: **Hyper-features** ϕ_C *for a window* w_l *are computed by finding a maximum disagreement between spatial relations that are predicted by GP* $\mu^r(x_i)$ *and induced by spatial relations* $\rho(w_i,w_l)$ *for all other windows* w_i *in the image. The figure illustrates this on a toy example with three windows* w_l, w_1, w_2. *For each* $r \in \{\texttt{overlap}, \texttt{part}, \texttt{cont}\}$ *the pairs of* $\mu^r(x_i)$ *and* $\rho(w_i,w_l)$ *with maximum disagreement are highlight in red.*

Global spatial relations & appearance ϕ_G

$$\phi^r_G(X,W,l) = \frac{2}{N^2-N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |\rho^r(w_i,w_l) - \rho^r(w_j,w_l)| \cdot K(x_i,x_j). \tag{3}$$

This hyper-feature reacts to pairs of candidate windows (w_i,w_j) with similar appearance (high $K(x_i,x_j)$) but different spatial relations to w_l. Two windows w_i,w_j contribute significantly to the sum if they look similar (high $K(x_i,x_j)$) *and* their spatial relations to w_l are different (high $|\rho^r(w_i,w_l) - \rho^r(w_j,w_l)|$).

A large value of ϕ_G indicates that the vector field of the spatial relations to w_l is not smooth with respect to appearance similarity. This indicates that w_l has a special role in the structure of spatial and appearance relations within that image. By measuring this pattern, ϕ_G helps the localization algorithm to select a better window, when the information contained in appearance features of w_l alone is not enough. For example, a window w_l tightly covering a small object such as the baseball in fig. 1(a) has high $\phi^{\texttt{cont}}_G$, because other windows containing it often look similar to windows not containing it. In this case, a high value of ϕ_G is a positive indication for w_l being a correct localization. On the other hand, a window w_l tightly covering the wolf in fig. 1(b) has low $\phi^{\texttt{overlap}}_G$, because windows that overlap with it are all similar to each other in appearance space. In this case, this low value is a positive indication for w_l being correct. In which direction to use this hyper-feature is left to the learning of its weight in the full scoring function, which is separate for each object class (sec 4.2).

Local spatial relations & appearance ϕ_L

$$\phi^r_L(X,W,l) = \frac{1}{N} \sum_{i=1}^{N} |1 - \rho^r(w_i,w_l)| \cdot K(x_i,x_l). \tag{4}$$

This hyper-feature is analogue to ϕ_G, but focuses around w_l in appearance space. It is indicative of whether windows that look similar to w_l (high $K(x_i,x_l)$) are also similar in position in the image, i.e. their spatial relation $\rho^r(w_i,w_l)$ to w_l is close to 1.

Window classifier score ϕ_S. The last hyper-feature is the score of a classifier which predicts whether a window w_l covers an instance of the class, based only on its appearance features x_l. Standard approaches to object localization typically consider only this cue [8,

[14, 15, 17, 32, 33, 36]. In principle, we could use any such method as the score ϕ_S here. In practice, we reuse the GP prediction of the overlap of a window with the object bounding-box as ϕ_s. One possibility would be to simply use the mean predicted overlap $\mu^{\texttt{overlap}}(x_l)$. However, as shown in [35], it is beneficial to take into account the uncertainty of the predicted overlap, which is also provided by the GP as the standard deviation $\sigma(x_l)$ of the estimate

$$\phi_S(X,l) = [\mu^{\texttt{overlap}}(x_l),\ \ \sigma(x_l)] \tag{5}$$

Using this hyper-features alone would correspond to the method of [35].

Complete score. Let

$$\Phi(X,W,l) = [\{\phi_G^r(X,W,l)\}_r, \{\phi_L^r(X,W,l)\}_r, \{\phi_C^r(X,W,l)\}_r, \phi_S(X,l)]$$

be a concatenation of all hyper-features defined above for a particular candidate window w_l, over all three possible relations: $r \in \{\texttt{overlap}, \texttt{part}, \texttt{cont}\}$. This amounts to 11 features in total. We formulate the following score function

$$E(\boldsymbol{\alpha}, X, W, l) = \langle \boldsymbol{\alpha}, \Phi(X,W,l) \rangle \tag{6}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product. Object localization translates to solving $\hat{l} = \arg\max_l E(\boldsymbol{\alpha}, X, W, l)$ over all candidate windows in the test image. The vector of scalars $\boldsymbol{\alpha}$ parametrizes the score function by weighting the hyper-features (possibly with a negative weight). We show how to efficiently maximize E over l in sec. 4.1 and how to learn $\boldsymbol{\alpha}$ in sec. 4.2.

4.1 Fast inference

We can maximize the complete score (6) over l simply by evaluating it for all possible l and picking the best one. The most computationally expensive part is the hyper-feature ϕ_G (3). For a given l, it sums over all pairs of candidate windows, which requires $O(N^2)$ subtractions and multiplications. Thus, a naive maximization over all l costs $O(N^3)$.

To simplify the notation, here we write the score function with only one argument $E(l)$, as the other arguments $\boldsymbol{\alpha}, X, W$ are fixed during maximization. Note that $0 \leq |\rho^r(w_i,w_l) - \rho^r(w_j,w_l)| \leq 1$, therefore

$$\alpha_G^r \cdot |\rho^r(w_i,w_l) - \rho^r(w_j,w_l)| \cdot K(x_i,x_j) \leq \left\{ \begin{array}{ll} 0 & ,\alpha_G^r \leq 0 \\ \alpha_G^r K(x_i,x_j) & ,\alpha_G^r > 0 \end{array} \right\} \tag{7}$$

Where α_G^r is the weight of hyper-feature ϕ_G^r. By substituting the elements in the sum over pairs in (3) with the bound (7), we obtain an upper bound on the ϕ_G^r term of the score (in fact three bounds, one for each r). We can then obtain an upper bound $\tilde{E}(l) \geq E(l)$ on the full score by computing all other hyper-features and adding them to the bounds on ϕ_G^r. This upper bound $\tilde{E}(l)$ is fast to compute, as (7) only depends on appearance features X, not on l, and computing the other hyper-features is linear in l.

We use the bound $\tilde{E}$ in an early rejection algorithm. We form a queue by sorting windows in descending order of $\tilde{E}(l)$. We then evaluate the full score $E(l)$ of the first l in the queue and store it as the current maximum. We then go to the next l in the queue. If its upper bound $\tilde{E}(l)$ is smaller than the current maximum, then we discard it without computing its full score. Otherwise, we compute $E(l)$ and set it as the current maximum if it is better than the previous best one. We iteratively go through the queue and at the end return the current maximum (which is now the best over all l). Notice, that the proposed fast inference method is exact: it outputs the same solution as brute force evaluation of all possible l.

4.2 Learning α with structured output regression

The scoring function (6) is parametrized by the weight vector $\boldsymbol{\alpha}$. We learn an $\boldsymbol{\alpha}$ from the training data of each class using a structured output regression formulation [6, 19, 20, 34]. Ideally, we look for $\boldsymbol{\alpha}$ so that, for each training image I, the candidate window l_I^* that best overlaps with the ground-truth bounding-box has the highest score. It is also good to encourage the score difference between the best window l_I^* and any other window w_l to be proportional to their overlap. This makes the learning problem smoother and better behaved than when using a naive $0/1$ loss which equally penalizes all windows other than l_I^*. Hence, we use the loss $\Delta(l,l') = 1 - \rho^{\texttt{overlap}}(w_l, w_{l'})$ proposed by [6], which formalizes this intuition. We can find $\boldsymbol{\alpha}$ by solving the following optimization problem

$$\min_{\boldsymbol{\alpha},\xi} \frac{1}{2}||\boldsymbol{\alpha}||^2 + \gamma \sum_I (\xi_I)$$

$$\text{s.t. } \xi_I \geq 0, \forall I : \langle \boldsymbol{\alpha}, \phi(X_I, l_I^*) \rangle - \langle \boldsymbol{\alpha}, \phi(X_I, l) \rangle \geq \Delta(l, l_I^*) - \xi_I, \forall I, \forall l \in L_I \backslash l_I^* \tag{8}$$

where I indexes over all training images. This is a convex optimization problem, but it has hundred of thousands of constraints (i.e. about 1000 candidate windows for each training image, times about 500 training images per class in our experiments). We solve it efficiently using quadratic programming with constraint generation [5]. This involves finding the most violated constraint for a current $\boldsymbol{\alpha}$. We do this exactly as we can solve the inference problem (6) and the loss Δ decomposes into a sum of terms which depend on a single window. Thanks to this, the constraint generation procedure will find the global optimum of (8) [5].

5 Related work

The first work to try to annotate object locations in ImageNet [15] addressed it as a window classification problem, where a classifier is trained on annotated images is then used to classify windows in images with no annotation. Later [35] proposed to regress the overlap of a window with the object using GP [27], which allowed for self-assessment thanks to GP probabilistic output. We build on [35], using Associative Embedding to learn the kernel between windows appearance features and GP to predict the spatial relations between a window and the object. Note how the model of [35] is equivalent to ours when using only the ϕ_S hyper-feature.

Importantly, both [15, 35], as well as many other object localization techniques [8, 14, 32, 33, 36], score each window individually based only on its own appearance. Our work goes beyond by evaluating windows based on richer cues measured outside the window. This is related to previous work on context [7, 10, 17, 18, 25, 26, 30] as well as to works that use structured output regression formulation [6, 19, 20, 34]. We review these areas below.

Context. The seminal work of Torralba [30] has shown that global image descriptors such as GIST give a valuable cue about which classes might be present in an image (e.g. indoor scenes are likely to have TVs, but unlikely to have cars). Since then, many object detectors [7, 17, 25, 28, 29, 32] have employed such global context to re-score their detections, thereby removing out-of-context false-positives. Some of these works also incorporate the region surrounding the object into the window classifier to leverage local context [8, 22, 24, 32].

Other works [7, 10, 18, 26] model context as the interactions between multiple object classes in the same image. Rabinovich et al. [26] use local detectors to first assign a class

label to each segment in the image and then adjusts these labels by taking into account co-occurrence between classes. Heitz and Koller [18] exploits context provided by "stuff" (background classes like road) to guide the localization of "things" (objects like cars). Several works [7, 10] model the co-occurrence and spatial relations between object classes in the training data and use them to post-process the output of individual object class detectors. An extensive empirical study of different context-based methods can found in [11].

The force driving those works is the semantic and spatial structure of scenes as arrangements of different object classes in particular positions. Instead, our technique works on a different level, improving object localization for a *single class* by integrating cues from the appearance and spatial relations of all windows in an image. It can be seen as a new, complementary form of context.

Localization with structured output regression was first proposed by [6]. They devised a training strategy that specifically optimizes localization accuracy, by taking into account the overlap of training image windows with the ground-truth. They try to learn a function which scores windows with high overlap with ground-truth higher than those with low overlap. The approach was extended by [34] to include latent variables for handling multiple aspects of appearance and truncated object instances. At test time an efficient branch-and-bound algorithm is used to find the window with the maximum score. Branch-and-bound methods for localization where further explored in [19, 20].

Importantly, the scoring function in [6, 19, 20, 34] still scores each window in the test image independently. In our work instead we score each window in the context of all other windows in the image, taking into account their similarity in appearance space as well as their spatial relations in the image plane (sec. 4). We also use structured output regression [31] for learning the parameters of our scoring function (sec. 4.2). However, due to interaction between all windows in the test image, our maximization problem is more complex than in [6, 19], making their branch-and-bound method inapplicable. Instead, we devise an early-rejection method that uses the particular structure of our scoring function to reduce the number of evaluations of its most expensive terms (sec. 4.1).

6 Experiments and conclusions

We perform experiments on the subset of ImageNet [9] defined by [15, 16, 35], which consists of 219 classes for a total of 92K images with ground-truth bounding-boxes. Following [35], we split them in two disjoint subsets of 60K and 32K for training and testing respectively. The classes are very diverse and include animals as well as man-made objects (fig. 5). The task is to localize the object of interest in images known to contain a given class [15, 16, 35]. We train a separate model for each class using the corresponding images from the training set.

Features. For our method and all the baselines we use the same features as AE-GP+ method from [35]: (i) three ultra-dense SIFT bag-of-words histograms on different color spaces [32] (each 36000 dimensions); (ii) a SURF bag-of-word from [35] (17000 dimensions); (iii) HOG [8] (2048 dimensions). We embed each feature type separately in a 10-dimensional AE [35] space. Next, we concatenate them together and add location and scale features as in [15, 35]. In total, this leads to a 54-dimensional space on which the GP operates, i.e. only 54 parameters to learn for the GP kernel.

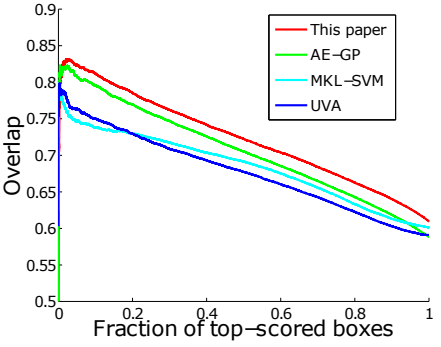

Figure 4: **Mean overlap curves.** *We retain the single top scoring candidate window in a test image and measure the mean overlap of the output windows with the ground-truth. We vary a threshold on the score of the output windows to generate performance curves. The higher the curve, the better.*

6.1 Baselines and competitors

MKL-SVM. This represents a standard, classifier driven approach to object localization, similar to [32, 33]. On 90% of the training set we train a separate SVM for each group of features described above. We combine these classifiers by training a linear SVM over their outputs on the remaining 10% of the data. We also include the location and scale features of a window in this second-level SVM. This baseline uses exactly the same candidate windows [23] and features as our method (sec. 6).

UVA [32]. The popular method [32] can be seen as a smaller version of the MKL-SVM baseline we have just defined. In order to make a more exact comparison to [32], we remove the additional features and use only their three SIFT bag-of-words. Moreover, instead of training a second level SVM, we simply combine their outputs by averaging. This corresponds to [32], but using the recent state-of-the-art object proposals [23] instead of selective search proposals. This method [32] is one of the best performing object detectors. It has won the ILSVRC 2011 [1] detection challenge and the PASCAL VOC 2012 detection challenge.

AE-GP [35]. Finally, we compare to the AE-GP+ model of [35]. It corresponds to a degenerate version of our model which uses only ϕ_S to score each window in isolation by looking at its own features (5). This uses the same candidate windows [23] and features we use. This technique was shown to outperform earlier work on location annotation in ImageNet [15].

6.2 Results

For each method we retain the single top scoring candidate window in a test image. We measure localization accuracy as the mean overlap of the output windows with the ground-truth [35] (fig. 4). We vary a threshold on the score of the output windows to generate performance curves. The higher the curve, the better.

As fig. 4 shows, the proposed method consistently outperforms the competitors and the baseline over the whole range of the curve. Our method achieves 0.75 mean overlap when returning annotations for 35% of all images. At the same accuracy level, AE-GP, MKL-SVM and UVA return 28%, 9% and 4% images respectively, i.e. we can return 7% more annotation at this high level of overlap than the closest competitor. Producing very accurate bounding-box annotations is important for their intended use as training data for various models and tasks. Improving over AE-GP validates the proposed idea of scoring candidate windows by

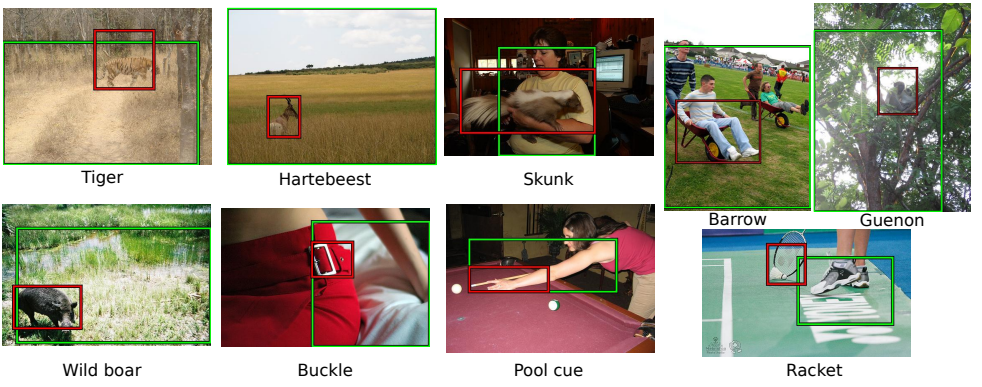

Figure 5: **Qualitative results.** *Results of our method (red) vs AE-GP [35] (green).* Notice, how our method is able to detect small, off-center objects despite occlusion (pool cue) or the object blending with its surroundings (tiger).

taking into account spatial relations to all other windows and their appearance similarity. The favourable comparison to the excellent method UVA [32] and to its extended MKL-SVM version demonstrates that our system offers competitive performance. Example objects localized by our method and by AE-GP are shown in fig. 5. Our method successfully operates in cluttered images (guenon, barrow, skunk). It can find camouflaged animals (tiger), small objects (buckle, racket), and deal with diverse classes and high intra-class variation (pool cue, buckle, racket).

To evaluate the impact of our fast inference algorithm (sec. 4.1) we compared it to brute force (i.e. evaluating the energy for all possible configurations) on the baseball class. On average brute force takes 17.6s per image, whereas our fast inference takes 0.14s. Since our inference method is exact, it produces the same solution as brute force, but 124× faster.

6.3 Conclusion

We have presented a new method for annotating the location of objects in ImageNet, which goes beyond considering one candidate window at a time. Instead, it scores each window in the context of all other windows in the image, taking into account their similarity in appearance space as well as their spatial relations in the image plane. As we have demonstrated on 92K images from ImageNet, our method improves over some of the best performing object localization techniques [32, 35], including the one we build on [35].

Acknowledgment This work was supported by the ERC Starting Grant VisCul. A. Vezhnevets was also supported by SNSF fellowship PBEZP-2142889.

References

[1] Imagenet large scale visual recognition challenge (ILSVRC). http://www.image-net.org/challenges/LSVRC/2011/index, 2011.

[2] University of amsterdam and euvision technologies at ilsvrc2013 (ILSVRC). http://www.image-net.org/challenges/LSVRC/2013/slides/ILSVRC2013-UvA-Euvision-web.pdf, 2013.

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

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

[5] G.H Bakir, Bernhard Schölkopf, Alexander J Smola, Ben Taskar, Vishwanathan, and S.V.N. Predicting structured data, 2007.

[6] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

[7] M. Choi, J. Lim, A. Torralba, and A. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

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

[9] 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.

[10] C Desai, D. Ramanan, and C. Folkess. Discriminative models for multi-class object layout. In *ICCV*, 2009.

[11] Santosh Kumar Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, pages 1271–1278, 2009.

[12] P. Dollar and C. Zitnick. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[15] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2012.

[16] M. Guillaumin, D. Küttel, and V Ferrari. ImageNet auto-annotation with segmentation propagation. *IJCV*, 2014.

[17] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

[18] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[19] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

[20] Alain Lehmann, Peter V Gehler, and Luc J Van Gool. Branch&rank: Non-linear object detection. In *BMVC*, volume 2, page 1, 2011.

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

[22] C. Li, D. Parikh, and T. Chen. Extracting adaptive contextual cues from unlabeled regions. In *ICCV*, pages 511–518. IEEE, 2011.

[23] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898. IEEE, 2014.

[25] K. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *NIPS*, 2003.

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

[27] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[28] B. Russell, A. Torralba, C. Liu, R. Ferugs, and W. Freeman. Object recognition by scene alignment. In *NIPS*, 2007.

[29] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011.

[30] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):153–167, 2003.

[31] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6: 1453–1484, 2005.

[32] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

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

[34] Andrea Vedaldi and Andrew Zisserman. Structured output regression for detection with partial truncation. In *NIPS*, 2009.

[35] A. Vezhnevets and V. Ferrari. Associative embeddings for large-scale knowledge transfer with self-assessment. In *CVPR*, 2014.

[36] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *ICCV*, pages 17–24. IEEE, 2013.