Looking out of the window: object localization by joint analysis of all windows in the image

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

Abstract

Traditionally, object localization 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 score 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 some of the best recent techniques that score windows in isolation.

1. Introduction

Traditionally, object localization is cast as an image window classification problem. The image is first decomposed into candidate windows, usually obtained through sliding window or object proposal generation. Each window is then scored by a classifier trained to discriminate instances of the class from other windows or a regressor trained to rank windows according to their overlap with the object. 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 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 analysing all the 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, with regards to the rest of the cluster, the precise bounding-box is positioned at the 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 an object localization method that 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. To represent spatial relations of windows we propose a descriptor, indicative for part/container relationships of the two windows and of how well aligned they are. We learn a windows appearance similarity kernel using the recent Associative Embedding technique, which enables us to meaningfully combine heterogeneous features like HOG and bag-of-words. We describe each window in an image with a set of hyper-features, which connect the appearance similarity and spatial relations of that particular window and all other image windows together. 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 a window scoring function. We devise a fast and exact procedure to optimize our score function over all candidate windows in a test image, and we learn its parameters using structured output regression.
We summarize the spatial relation between windows and conclude that are presented in sec. 7.

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 object localization model. Sec. 5 provides details on how we efficiently perform regression from appearance space to the space of spatial relations. In sec. 6 we review related work, including prior work on context [17, 18, 19, 20, 21, 4, 22]. Experiments and conclusions are presented in sec. 7.

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 [9, 1, 7, 23, 2, 10] (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 [9, 1, 19]. 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, state-of-the-art method [10].

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 method (sec. 4). We summarize the spatial relation between windows \( w \) and \( w' \) using the following spatial relation descriptor (fig. 2(a))

\[
\rho(w, w') = \left( \frac{w \cap w'}{w \cup w'}, \frac{w \cap w'}{w}, \frac{w \cap w'}{w'} \right)
\]

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 [24, 25]. 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 \( w_i \), we can compute the spatial relation descriptor to any window \( w \). This induces a vector field \( \rho(\cdot, w_i) \) 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 object 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, which we denote \( w^* \). For the
images in the training set, we know the spatial relations \( \rho(w, w') \) between all candidate windows \( w \) and the ground-truth 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.

We propose to use Gaussian Processes (GP) regression to learn to predict a probability distribution \( P(\rho(w, w^*)|x) \sim GP(m(x), K(x, x')) \) for each spatial relation \( r \in \{\text{overlap, part, cont}\} \) given window appearance features \( x \). We use zero mean \( m(x) = 0 \) and, for now, assume that a kernel (covariance function) \( K(x, x') \) is given. 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. Sec. 5 describes how we learn \( K \) for the high dimensional appearance features we use.

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 \( \mu(x_i) = (\mu_{\text{overlap}}(x_i), \mu_{\text{part}}(x_i), \mu_{\text{cont}}(x_i)) \), and their standard deviation as \( \sigma(x_i) \). The standard deviation is the same for all elements of the relation descriptor, as we use the same kernel and inducing points.

4. Object localization with spatial relations

Most object localization methods score candidate windows independently, i.e. each window is scored based only on that window’s appearance features \( E = \{B, E\} \). In contrast, we introduce a model that scores a candidate window by connecting together the information contained in the spatial relations the candidate has with all other candidate windows, their appearance similarity, and the GP prediction of their spatial relations with the object’s bounding-box. The appearance of \( w \) is of course also taken into account, by including the GP prediction of its spatial relations to the object bounding-box. All these elements are tied together in an energy function (sec. 4.1) which we minimize to select the window deemed to contain the object (sec. 4.2).

We consider a single-instance detection for the sake of clarity and show how to generalize our method to multi-instance detection in sec. 7. The parameters of this energy function are learned using structured output regression (sec. 4.3).

4.1. Object localization model

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 \( K(x_i, x_j) \).

Let \( w_l \in W \) be a candidate window to be scored. We proceed by defining a set of hyper-features \( \Phi(X, W, M, l) \), that characterize \( w_l \), and then define our energy function through them. Fig. 3 illustrates which information source each hyper-feature is using.

The Consistency of predicted & induced spatial relations \( \phi_C \)

\[ \phi_C(X, W, l) = \max_i |\rho^*(w_i, w_l) - \mu^*(x_i)| \] (2)

Assume for a moment that \( w_l \) correctly localizes an instance of the object class. Selecting \( w_l \) would induce spatial relations \( \rho^*(w_i, w_l) \) to all other windows \( w_i \). The hyper-feature \( \phi_C \) checks whether these induced spatial relations are consistent with those predicted by GP based on the appearance of the other windows \( \mu^*(x_i) \). 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 \( \rho^*(w_i, w_l) \) and predicted \( \mu^*(x_i) \) on the window \( w_i \) with the largest disagreement. Fig. 4 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 \( \{\mu^*(x_i)\}_{i=1}^N \) and observed \( \{\rho^*(w_i, w_l)\}_{i=1}^N \).
measurements of the vector field $\rho^r(\cdot, w_l)$, which is more indicative of $w_l$ being a correct localization.

**Global spatial relations & appearance $\phi_G$**

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

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 $\phi_G$ 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 test image. By measuring this pattern, $\phi_G$ 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 $\phi_G^{\text{part}}$, because other windows containing it often look similar to windows not containing it. In this case, a high value of $\phi_G$ 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 $\phi_G^{\text{overlap}}$, 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 energy, which is separate for each object class.

**Local spatial relations & appearance $\phi_L$**

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

This hyper-feature is analogue to $\phi_G$, 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 $\phi_S$.** The last hyper-features 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 [3, 4, 5, 11, 16, 17]. In principle, we could use any such method as the score $\phi_S$ here.

In practice, we reuse the GP prediction of the overlap of a window with the object bounding-box as $\phi_S$. One possibility would be to simply use the mean predicted overlap $\mu^{\text{overlap}}(x_l)$. However, as shown in [2], 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

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

Using this hyper-features alone would correspond to the method of [2].

**Complete energy.** Let

$$
\Phi(X, W, l) = [\phi^G(X, W, l), \phi^L(X, W, l), \phi^C(X, W, l), \phi_S(X, l)]
$$

be a concatenation of all hyper-features defined above for a particular candidate window $w_l$, over all three possible relations: $r \in \{\text{overlap, part, cont}\}$. This amounts to 11
The energy function \( E(\alpha, X, W, l) \) is parametrized by the weight vector \( \alpha \). We learn an \( \alpha \) from the training data of each class using a structured output regression formulation [11, 22, 12, 13]. Ideally, we look for \( \alpha \) so that, for each training image \( I \), the candidate window \( l'_i \) that best overlaps with the ground-truth bounding-box has the lowest energy. It is also good to encourage the energy 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 \( \Delta(l, l') = 1 - \rho_{\text{overlap}}(w_l, w_{l'}) \) proposed by [11], which formalizes this intuition. We can find \( \alpha \) by solving the following optimization problem

\[
\min_{\alpha, \xi} \frac{1}{2} ||\alpha||^2 + \gamma \sum_i (\xi_i) \\
\text{s.t.} \quad \xi_i \geq 0, \forall I \\
(\alpha, \phi(X, l)) - (\alpha, \phi(X, l'_i)) \leq \Delta(l, l'_i) - \xi_i, \forall I, \forall l \in L_i \setminus l'_i
\]

where \( I \) indexes over all training images. This is a convex optimization problem, but it has hundreds 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 [28]. This involves finding the most violated constraint for a current \( \alpha \). We do this exactly as we can solve the inference problem [6] and the loss \( \Delta \) 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) [28].

5. Exemplar-driven kernel learning

In this section we learn the kernel \( K \), used in hyper-features \( \phi_G \) and \( \phi_L \) (sec. 4.1) and in the GP (sec. 3) that predicts the spatial relations of candidate windows to the object. We define \( K \) as

\[
K(x_i, x_j) = \lambda_0 \exp(-(x_i - x_j)^T C(x_i - x_j)^T)
\]
where $C$ is a positive-definite matrix and $\lambda_0$ a scaling parameter. Directly learning $C$ in a high dimensional space requires estimating $(D^2 - D)/2$ parameters. In our case $D$ is in the order of $10^5$, which renders it impossible. Learning a diagonal $C$ (squared exponential kernel [26]) would involve much fewer parameters, but it would also ignore the correlation between different feature dimensions. Moreover, it would be prohibitively expensive in terms of computation and memory. We have about 1000 inducing points, corresponding to candidate windows $W$, for each training image. We must keep them all in memory to run GP prediction, and the computation of the kernel scales linearly with dimensionality.

Following the idea of [2], we first learn a low dimensional linear embedding $\Psi$ of the original feature space, and then learn a diagonal covariance in the embedded space by maximizing the GP marginal likelihood on the training data. The resulting covariance in the embedded space corresponds to a non-diagonal covariance in the original space and so captures correlations between different dimensions. Moreover, the GP can operate exclusively in the low dimensional space, relieving the memory and computational burden, both at training and test time.

**Learning the embedding $\Psi$.** We adopt the exemplar-driven Associate Embedding [2] (AE) $\Psi$, which greatly reduces the dimensionality of the appearance feature space while preserving its descriptive power. The method proceeds by first training Exemplar-SVMs (E-SVM) [3] from each ground-truth object bounding-box. Then all candidate windows are represented as a vector of E-SVM outputs. This representation is rich, since each entry is a similarity to a ground-truth bounding-box. It is also redundant, as outputs of E-SVMs trained on similar examples are very correlated and background windows receive similar negative scores from the majority of E-SVMs. This redundancy makes it suitable for heavy compression. We apply SVD-based dimensionality reduction to this representation, obtaining a linear embedding of the original feature space. The AE space was shown [2] to be informative even when reducing dimensionality to as little as 10 dimensions.

**Implementation details.** We use 5 feature types: (i) three ultra-dense SIFT bag-of-words histograms on different color spaces [1] (each 36000 dimensions); (ii) a SURF bag-of-word from [2] (17000 dimensions); (iii) HOG [6] (2048 dimensions). We use linear E-SVMs for HOG, and use a $\chi^2$ kernel for all other feature types. For computational efficiency, we approximate the $\chi^2$ kernel with the expansion technique of [29]. We embed each feature type separately in a 10-dimensional AE space. Next, we concatenate them together and add location and scale features as in [23][2]. 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 [9].

**6. Related work**

Traditionally, object localization is cast as a window classification problem [3][5][1][6][7][8], where each window is scored individually based only on its own appearance. Our work goes beyond by evaluating windows based on richer cues measured outside the window and is related to previous work on context [17][18][19][20][21][4][22] as well as to works that use structured output regression formulation [11][27][12][13]. We review both of these areas below.

**Context.** The seminal work of Torralba [22] 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 [17][5][4][1][30] have employed such global context to re-score their detections, thereby removing out-of-context false-positives.

Other works [21][18][19][20] model context as the interactions between multiple object classes in the same image. Rabinovich et al. [21] 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 [19][20] 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.

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 [11]. They devised a training strategy that specifically optimizes localization accuracy, by taking into account the overlap of training image windows with the ground-truth. The structured output regression tries to learn a window scoring function, which scores windows with high overlap with ground-truth higher than those with low overlap. The approach was further extended by [12] to include latent variables for handling multiple aspects of appearance and truncated training 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 [27][13].
Importantly, the scoring function in [11, 27, 12, 13] 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. 4.1). We as well use structured output regression [14] for learning the parameters of the scoring function (sec. 4.3). Due to interaction between all windows in the test image, our maximization problem is more complex than in [11 27], making their branch-and-bound method inapplicable. Instead, we devise an early-rejection method (sec. 4.2) that uses the particular structure of our scoring function to reduce the number of evaluations of its most expensive terms.

**Associative Embedding.** We build on the Associative Embedding of [2], which we use to learn the kernel between windows appearance features. The model of [2] is equivalent to using only the $\phi_S$ hyper-feature and still scores each candidate window based only on its own appearance.

### 7. Experiments and conclusions

We perform experiments on the subset of ImageNet [15, 16] defined by [23, 2], which consists of 219 classes for a total of 92K images with ground-truth bounding-boxes. Following [2], 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. 6). The task is object localization [16, 23, 2], i.e. localize the object of interest in the test images of a given class.

#### 7.1. Baselines and competitors

**MKL-SVM.** This represents a standard, classifier driven approach to object localization, similar to [3, 1]. On 90% of the training set we train (i) three SVMs with ultra-dense SIFT bag-of-words defined on different color spaces from [11] (linear with $\chi^2$ expansion [29]); (ii) an SVM on SURF bag-of-words [2]; (iii) an SVM on HOG [6] (linear). 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 [10] and features as our method (sec. 5).

**UVA [11]** The popular method [11] 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 [11], 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 [11], but using the recent state-of-the-art object proposals [10] instead of selective search proposals. This method [11] is one of the best performing object detectors. It has won the ILSVRC 2011 [16] detection challenge and the PASCAL VOC 2012 detection challenge.

**AE-GP [2]**. Finally, we compare to the AE-GP+ model of [2]. It corresponds to a degenerate version of our model which uses only $\phi_S$ to score each window in isolation by looking at its own features [5]. This uses the same candidate windows [10] and features we use (sec. 5).

#### 7.2. Results

**Single-instance.** For single instance localization, each method outputs the single top scoring candidate window in a test image. We report two performance measures: (a) mean overlap of the returned windows with the ground-truth. This measures the spatial accuracy of the detections (fig. 5(a)); (b) detection-rate, i.e. the percentage of returned windows with overlap $> 0.5$ (i.e. the PASCAL VOC criterion [24]: fig. 5(b)). For both measures we vary a threshold on the score of the returned windows to generate performance curves.

As fig. 5 shows, the proposed method consistently outperforms the competitors and the baseline over nearly the...
Figure 6. Example results. Our method (red) vs the AE-GP method [2] that only uses $\phi_S$ (green). Results improve noticeably for difficult, small objects.

Multi-instance. To produce multi-instance localization we use the same non-maxima suppression method from [5] for all the baselines and competitors. For our method we first pick the window with the highest score, then apply the non-maxima suppressor [5] and iteratively re-run inference (sec. 4.2) and non-maxima suppression until no candidate window is left. This procedure incurs only a small additional computational cost over the initial localization, as we cache the values of the hyper-features and reuse them across iterations. We report precision/recall curves over the whole dataset in fig. 7. The proposed method moderately, but uniformly outperforms AE-GP (its reduced version) for all levels of recall. Both MKL-SVM and UVA methods perform very similarly. At fixed recall of 20% our system achieves precision of 70% against 60% and 64% by UVA and MKL-SVM respectively.

7.3. Conclusion

We have presented a new method for object localization, which goes beyond considering one candidate window at a time. Instead, it 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. As we have demonstrated on 92K images from ImageNet, our method improves over some of the best performing object detectors [1][2], including the one we build on [2].

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

[1] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[2] A. Vezhnevets and V. Ferrari, “Associative embeddings for large-scale knowledge transfer with self-assessment,” in *CVPR*, 2014.

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

[4] H. Harzallah, F. Jurie, and C. Schmid, “Combining efficient object localization and image classification,” in *ICCV*, 2009.

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE Trans. on PAMI*, vol. 32, no. 9, 2010.

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

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

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

[9] B. Alexe, T. Deselaers, and V. Ferrari, “Learning to localize objects with structured output regression,” in *ECCV*, 2008.

[10] A. Vedaldi and A. Zisserman, “Structured output regression for detection with partial truncation,” in *NIPS*, 2009.

[11] A. Lehmann, P. V. Gehler, and L. J. Van Gool, “Branch&rank: Non-linear object detection,” in *BMVC*, vol. 2, p. 1, 2011.

[12] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” *JMLR*, vol. 6, pp. 1453–1484, 2005.

[13] 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.

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

[15] 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.

[16] G. Heitz and D. Koller, “Learning spatial context: Using stuff to find things,” in *ECCV*, 2008.

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

[18] C. Desai, D. Ramanan, and C. Folkess, “Discriminative models for multi-class object layout,” in *ICCV*, 2009.