Accurate Object Detection with Location Relaxation and Regionlets Re-localization

Chengjiang Long[1], Xiaoyu Wang[2], Gang Hua[1], Ming Yang[3], and Yuanqing Lin[2]

[1] Stevens Institute of Technology, Hoboken, NJ 07030
[2] NEC Laboratories America, Cupertino, CA 95014
[3] Facebook, Menlo Park, CA 94026

Abstract. Standard sliding window based object detection requires dense classifier evaluation on densely sampled locations in scale space in order to achieve an accurate localization. To avoid such dense evaluation, selective search based algorithms only evaluate the classifier on a small subset of object proposals. Notwithstanding the demonstrated success, object proposals do not guarantee perfect overlap with the object, leading to a suboptimal detection accuracy. To address this issue, we propose to first relax the dense sampling of the scale space with coarse object proposals generated from bottom-up segmentations. Based on detection results on these proposals, we then conduct a top-down search to more precisely localize the object using supervised descent. This two-stage detection strategy, dubbed *location relaxation*, is able to localize the object in the continuous parameter space. Furthermore, there is a conflict between accurate object detection and robust object detection. That is because the achievement of the later requires the accommodation of inaccurate and perturbed object locations in the training phase. To address this conflict, we leverage the rich spatial information learned from the Regionlets detection framework to determine where the object is precisely localized. Our proposed approaches are extensively validated on the PASCAL VOC 2007 dataset and a self-collected large scale car dataset. Our method boosts the mean average precision of the current state-of-the-art (41.7%) to 44.1% on PASCAL VOC 2007 dataset. To our best knowledge, it is the best performance reported without using outside data [4].

1 Introduction

An object may appear in any locations and scales in an image defined by the continuous parameter space spanned by (x, y, s, a), where (x, y) is the object center point, and s and a are the scale and aspect ratio of the object. In particular, different aspect ratios generally correspond to different viewpoints, leaving a difficult open question for robust object detection.

In order to accurately localize the object in the image, sliding window based detector [1–5] requires densely sampling a fixed size candidate object window (*i.e.*, a base window) from the continuous parameter space at each scale of a scale-space image pyramid. Then, a binary decision is made for each specific window to predict whether

[4] Convolutional neural network based approaches are commonly pre-trained on a large scale *outside* dataset and fine-tuned on the VOC dataset.

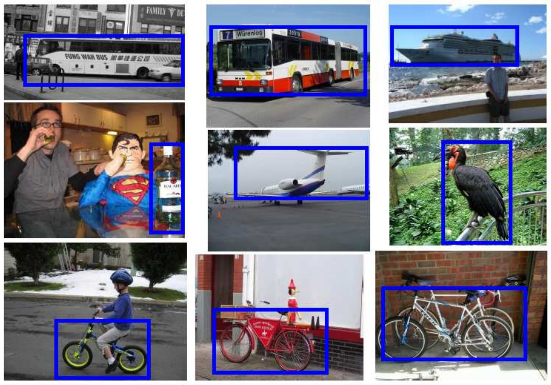

Figure 1: Sample detection results applying our detection framework to the PASCAL VOC 2007 dataset. First row: bus and boat detection. Second row: bottle, aeroplane and bird detection. Third row: bicycle detection.

it contains the object or not. To deal with different viewpoints of the object, one often discretizes the space of aspect ratio to define different base windows, and one classifier needs to be trained for each base window to detect the same object with different viewpoints.

Obviously, sliding window based approaches could be computationally prohibitive to obtain precise localization of the object, as it may potentially involve evaluating the classifier on millions or even billions of candidate windows. To reduce the computational cost, as suggested by the seminal Viola-Jones detector [6], a cascade classifier allows to early reject obvious non-object window, and hence achieves real-time performance. This strategy has been widely adopted in the literature. However, unless the weak classifier in the cascade can be efficiently evaluated, *e.g.*, by leveraging Haar features with integral images, the computational cost even with early rejection may still be very high.

Beyond cascade classifiers, the computational cost could be further reduced either from top-down or bottom-up approaches. Top-down methods, such as branch-and-bound [7], divide and conquer [8], and crosstalk [9] *etc.*, take advantage of observations from already evaluated windows to prune the windows which are not likely to have the object. While bottom-up methods guide their search by firstly identifying category independent candidate object locations before applying category specific detectors. This can be achieved either through low-level segmentations [10, 11] or through some "objectness" [12] measurement of a candidate window. Since the number of classifier evaluation is drastically pruned in such bottom-up methods, even computational intensive spatial pyramid matching [13], which is very successful in image classification, can be adopted for object detection.

Notwithstanding the great success of these methods for reducing the computational cost for object detection, none of these methods searched for the object in the full continuous parameter space, *i.e.*, the center point, scale, and aspect ratio of the object.

In other words, for top-down approaches, the detection accuracy is still bounded by the level of quantization these algorithms operating on. For bottom-up approaches, the recall of the detector is bounded by the recall of the category independent object proposal.

Moreover, most of the above approaches still rely on classification models to localize the object. While a classifier could be robust due to large scale training, it is not necessarily optimized for accurate object localization. What worsens the situation is that many detectors such as DPM [4] are not trained on the exact ground truth positive samples. These detectors allow samples with sufficient overlap with the ground truth being positive training samples, for either data augmentation purpose or a more comprehensive modeling of visual appearance among different positive samples. Thus in contrast to aiming at precise localization as much as possible, the visual classification models are learned to accommodate inaccurate localizations.

These observations motivate us to develop a detection framework which is capable of precisely searching for the object in a full parameter space with favorable efficiency. To achieve this goal, we first relax dense sampling of the object location and scale, dubbed the name *location relaxation*, and only evaluate the detector at a much coarser set of locations and scales. For coarse detection windows which have relatively high response, we apply supervised descent search [14] to find potential object hypothesis by simultaneously optimizing their center point, scale, and aspect ratio. The resulting detections are much more improved with supervised descent search but still not sufficient in terms of accurate localization. Thus we introduce Regionlets Re-localization, which is naturally built based on the quantized Regionlets features, to directly predict the true object location based on results from supervised descent search.

Figure 2 takes person detection as an example to illustrate our object detection framework. By applying an object detector to bottom-up object proposals, we obtain coarse detections, *i.e.*, the bounding boxes shown in Figure 2(b). Among them, the red box is relatively confident detection compared to others. Through the supervised descent search starting from the red bounding box, a better detection is obtained as the dash box in Figure 2 (c). Finally we apply Regionlets Re-localization to determine the object location as shown in Figure 2 (d). We show some sample detection results on the PASCAL VOC 2007 dataset in Figure 1.

The contribution of this paper lies on three aspects. Firstly, it proposed coarse detection plus supervised descent search in a fully parameterized location space for generic object detection which shows promising performance. Secondly, it proposed a novel Regionlets Re-localization method which complements the suboptimal object localization performance given by object detectors. Finally, our detection framework achieves the best performance on the PASCAL VOC 2007 dataset without using any outside data. It also demonstrates superior performance on our self-collected car dataset.

2 Our approach

Our object detection framework is composed of three key components: bottom-up object hypotheses generation, top-down object search with supervised descent and object re-localization with a localization model.

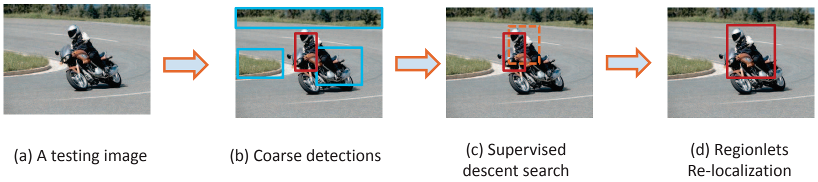

Figure 2: Illustration of the proposed object detection framework. (a) A testing image on which we want to detect all persons. (b) Coarse detection results obtained from object detectors applied to bottom-up proposals. The red bounding box indicates a relatively confident detection. (c) More confident detections obtained through supervised descent search. (d) The Regionlets Re-localization is employed to produce better localization. A non-max suppression procedure is followed to generate the final detection result.

There are several alternatives to obtain object hypotheses. For example, through the objectness measurement [12], the saliency analysis or their combinations [15], or using segmentation cues [10]. Because our top-down search algorithm is applied locally, we expect the bottom-up object hypotheses to split the object location space evenly, to avoid the search algorithm converging to the same local minimum. To this end, we employ low-level segmentation to propose the object hypotheses. The superpixel segmentation merges similar pixels locally into disjoint sets which perfectly matches our need. However, over-segments only provide small object candidates. To obtain object hypotheses for large objects, the over segmented superpixels are gradually merged to produce larger candidates.

The detection with location relaxation takes coarse detection results from a detector applied on the bottom up object proposals. Then it searches the object location guided by discriminatively learned descent model inspired by Xiong and De la Torre [14]. The learned supervised descent model is used to predict the next more accurate object location to explore based on observations from the current location. Although our method is applicable with any black box object detector, we use the Regionlets detector [16] due to its outstanding performance and flexibility to detect objects in any viewpoints.

All the detection results, including the original coarse detections as well as detections generated by supervised descent search, are fed to our Regionlets Re-localization process to more accurately locate the target objects.

2.1 Bottom-up object proposal

To complement our top-down searching strategy, we employ a segmentation based bottom-up scheme to generate our initial set of candidate searching locations. Similar to [10], we start with over-segments (*i.e.*, superpixels) of an image and then hierarchically group these small regions to generate object hypotheses. We use [17] to generate superpixel segments. A segmented region r_i is described by several characteristics , *i.e.*, the size of the region (total number of pixels), color histograms, and the texture information (gradient orientation histograms). Four neighbor region similarities are defined based on these characteristics as shown in the following equations:

$$S_c(r_i, r_j) = \sum_{k=1}^{n} \min(c_i^k, c_j^k), \tag{1}$$

$$S_s(r_i, r_j) = 1 - \frac{sz(r_i) + sz(r_j)}{sz(im)}, \tag{2}$$

$$S_t(r_i, r_j) = \sum_{k=1}^{n} \min(t_i^k, t_j^k), \tag{3}$$

$$S_f(r_i, r_j) = 1 - \frac{sz(bb_{ij}) - sz(r_i) - sz(r_j)}{sz(im)}. \tag{4}$$

where c_i^k is the kth dimension of the color histogram, $sz(r_i)$ is the number of pixels in image region r_i, im stands for the whole image, t_i^k is the kth dimension of the texture histogram, bb_{ij} is the rectangular region which tightly bound region r_i and r_j. S_c, S_s and S_t are the color similarity, size similarity, texture similarities, respectively. S_f measures how the combined two regions will occupy the rectangular bounding box which tightly bounds them. The similarity of two adjacent regions can be determined by any combination of the four similarities.

The two regions with the highest similarity *w.r.t* the similarity measurement are merged first and this greedy process is repeated following an agglomerative style clustering scheme. Each merging step produces a bounding box which bounds the merged two regions. In principle, we want regions from the same object to be merged together. Each low level cue contributes from its aspect. For example, the color similarity measures the color intensity correlation between neighbor regions which encourage regions similar in color to be merged together. The size similarity encourages small regions to merge first. The fill similarity encourages the bounding box to tightly bound the merged region. The texture similarity measures the similarity of appearance in gradient, which is complementary to color similarity. The usage of similarity measures and segmentation parameters are detailed in the experiment section.

2.2 Top-down Supervised Object Search

Once the coarse object hypotheses are obtained, we apply an object detector to determine relatively confident detections. The top-down supervised descent search is only applied to these confident detections.

Supervised descent is a general approach to optimize an objective function which is neither analytically differentiable nor practical to be numerically approximated. It is very suitable for vision problems when visual feature is involved in optimizing the objective function, because most visual features such as SIFT, HOG, and LBP histogram are not differentiable with respect to locations. Instead of computing the descent direction from the gradient, supervised descent uses a large number of examples to train a regression model to predict the descent direction. The training process requires features, which serves as the regressor, to be a fixed length vector, while bottom up segmentations naturally produces arbitrary size proposals. To deal with this issue, we normalize the bounding boxes to a fixed size. In the following, we explain how the supervised descent is adopted to find objects in a full parameter space.

Given an initial object hypothesis location $\mathbf{o}_0 = [x_0, y_0, s_0, a_0]^T$, which may not accurately bound the object, our objective is to use supervised descent to greedily adjust the bounding box by a local movement $\varDelta\mathbf{o} = [\varDelta x, \varDelta y, \varDelta s, \varDelta a]^T$, leading to a more accurate localization of the object. The goal of the supervised descent training process is hence to learn a sequence of K models to predict the optimal descent direction of the bounding box for each step of the supervised descent, where the needed supervised descent step K is also automatically identified from the training process.

More specifically, denote $\varPhi(\mathbf{o}_{k-1})$ to be the n dimensional feature vector extracted from the bounding box defined by $\mathbf{o}_{k-1}$ in the $k-1$ step of the supervised descent process, we learn an $n \times 4$ linear projection matrix $\mathbf{R}_{k-1} = [\mathbf{r}^x_{k-1}, \mathbf{r}^y_{k-1}, \mathbf{r}^s_{k-1}, \mathbf{r}^a_{k-1}]^T$ and a four dimensional bias vector $\mathbf{b}_{k-1} = [b^x_{k-1}, b^y_{k-1}, b^s_{k-1}, b^a_{k-1}]^T$ so that the bounding box movement can be predicted as $\varDelta\mathbf{o}_k = \mathbf{R}^T_{k-1}\varPhi(\mathbf{o}_{k-1}) + \mathbf{b}_{k-1}$ based on the location from the $k-1$ step. $\varPhi(\cdot)$ indicates the feature extracted which is HOG and LBP histogram in our experiments.

We first explain the training process for the first supervised descent model, followed by details to train models sequentially after. Given a set of labeled ground truth object locations $\{\mathbf{o}^i_* = (x^i_*, y^i_*, s^i_*, a^i_*)\}$, we construct the starting locations $\{\mathbf{o}^i_0 = (x^i_0, y^i_0, s^i_0, a^i_0)\}$ of the object by applying a random perturbation from the ground truth but assure that they are overlapped. The training of the projection matrix $\mathbf{R}_0$ and the bias $\mathbf{b}_0$ is to solve the following optimization problem:

$$\arg\min_{\mathbf{R}_0, \mathbf{b}_0} \sum_i ||\varDelta\mathbf{o}^i_{0*} - \varDelta\mathbf{o}^i_0||^2, \tag{5}$$

where $\varDelta\mathbf{o}^i_{0*} = \mathbf{o}^i_* - \mathbf{o}^i_0$ is the true movement and $\varDelta\mathbf{o}^i_0 = \mathbf{R}^T_0\varPhi(\mathbf{o}^i_0) + \mathbf{b}_0$ is the predicted displacements of the state vector. The optimal $\mathbf{R}_0$ and $\mathbf{b}_0$ are computed in a closed-form by a linear least square method.

The subsequent $\mathbf{R}_k$ and $\mathbf{b}_k$ for $k = 1, 2, \ldots$, can be learned iteratively. At each iteration, we update the new locations determined by the previous model $\mathbf{R}_{k-1}$ and $\mathbf{b}_{k-1}$,

$$\mathbf{o}^i_k = \mathbf{o}^i_{k-1} + \mathbf{R}^T_{k-1}\varPhi(\mathbf{o}^i_{k-1}) + \mathbf{b}_{k-1}. \tag{6}$$

By updating $\varDelta\mathbf{o}^i_{k*} = \mathbf{o}^i_* - \mathbf{o}^i_k$ and $\varDelta\mathbf{o}^i_k = \mathbf{R}^T_k\varPhi(\mathbf{o}^i_{k-1}) + \mathbf{b}_{k-1}$ the optimal $\mathbf{R}_k$ and $\mathbf{b}_k$ can be learned from a new linear regression problem by minimizing

$$\arg\min_{\mathbf{R}_k, \mathbf{b}_k} \sum_i ||\varDelta\mathbf{o}^i_{k*} - \varDelta\mathbf{o}^i_k||^2. \tag{7}$$

The error empirically decreases as more iterations are added [14]. In our experiments, this training of supervised descent models often converged in 20-30 steps.

Given a testing image, we firstly apply the cascade regionlets detector [16] to the coarse bottom-up object candidates. Object hypotheses which produces high detection scores are fed to the iterative supervised descent search process to perform local search.

New locations output by supervised descent search are re-evaluated by the object detector to obtain the detection score. By ranking all the detection scores from searched locations, we keep the most confident detections.

2.3 Regionlets Object Re-localization

The supervised descent search introduced in the previous subsection significantly improve the detection rate by scanning more predicted object candidates. In this section, we assume the object has already been detected, but with non-perfect localization. To further improve the object detection system, we train a model specific for object localization taking advantage of features extracted from the Regionlets detection model.

The Regionlets detector [16] is composed of thousands of weak classifiers learned with RealBoost. These weak classifiers are formed as several cascades for early rejection, yielding fast object detection. The cascade structure is not related to our re-localization approach and would not be included in the following presentation without any misunderstanding. The input of each weak classifier in the Regionlets model is a 1-D feature extracted from a rectangular region in the detection window. In the trainging process, these 1-D features are greedily chosen to minimize the logistic loss over all training samples, which is based on classification errors. More details about the Regionlets learning and testing are beyond the scope of this paper and can be found from [16].

Not only does the Regionlets training process greedily select discriminative visual appearances, but also it determines the spatial regions to extract the 1-D feature. Thus the resulting weak features extracted from regionlets implicitly encode thousands of spatial locations, which could be used to further predict the precise location of an object. It is worth noting that the detector learning only targets on minimizing the classification error which does not necessarily guarantee that the localization error is also minimized at the same time.

To leverage the rich spatial information encoded in the Regionlets model, we let each Regionlet vote the object's position. Given the object location (l, t, r, b) detected by the object detector ((l, t, r, b) represents the object's left, top, right and bottom coordinates, respectively), the problem is equivalent to predict the localization error $(\Delta l_n, \Delta l_t, \Delta l_r, \Delta l_b)$ of the current detection so that the true object location is computed as:

$$\begin{aligned} l^* &= l + w\Delta l_n, & t^* &= t + h\Delta t_n, \\ r^* &= r + w\Delta r_n, & b^* &= b + h\Delta b_n. \end{aligned} \tag{8}$$

Here (l^*, t^*, r^*, b^*) is the ground truth object location. (l, t, r, b) is the bounding box detected with the Regionlets model. $w = r - l + 1$, $h = b - t + 1$ are the detected bounding box width and height respectively. $(\Delta l_n, \Delta t_n, \Delta r_n, \Delta b_n)$ are the relative localization error between the ground truth and the current detection. It is normalized by the width and height of the detected objects[5]. Detections from Regionlets model have

[5] We empirically found that using the four coordinates for our localization model produces better performance than using (x, y, s, a). Thus we choose (l, t, r, b) in our Regionlets Re-localization approach.

various sizes, we observe that normalizing displacement errors is critical to stabilize the training and prediction.

Training the localization model is to learn a vector V, so that we can predict the localization error :$\Delta L = V^T R$, where ΔL is either Δl_n, Δt_n, Δr_n, or Δb_n, R is the feature extracted for from regionlets. We minimize the squared localization error in the model training phase. More specifically, we solve a support vector regression problem for each of the four coordinates respectively:

$$\min_V \left\{ \frac{\|V\|}{2} + C \sum_{m=1}^{M} \max(0, |\Delta L_m - V^T R_m| - \epsilon)^2 \right\}, \tag{9}$$

where V is the coefficient vector to be learned, ΔL_m is the normalized localization error of training sample m, R_m is the feature extracted from all the Regionlets in the object detection model for the mth sample as explained in the following, M is the total number of training examples. The first term in the Equation (9) is the regularization term, while C is a trade-off factor between the regularization and the sum of squared error, ϵ is the tolerance factor. The problem can be effectively solved using the publicly available liblinear package [18].

The feature R is extracted from the discriminatively learned Regionlets detection model. However, directly applying Regionlets features produces poor performance. Based on the weak classifier learned on each Regionlets feature, we transfer the 1-D Regionlet feature into a sparse binary vector. Each Regionlets weak classifier is a piece-wise linear function implemented using a lookup table:

$$h_i = \sum_{j=1}^{8} w_{i,j} \delta(Q(f_i) - j), \tag{10}$$

where f_i is the 1-D feature extracted from a group of regionlets, $Q(f_i)$ quantize the feature f_i into an integer from 1 to 8. $\delta(x) = 1$ when $x = 0$ otherwise 0. $\{w_{i,j}\}_{j=1}^{8}$ is the classifier weights learned in the boosting training process. We transfer $Q(f_i)$ into an 8-dimensional binary vector r, where the jth dimension is computed as $r(j) = \mathbb{1}(Q(f_i) = j)$, and $\mathbb{1}(\cdot)$ is the indicator function. Apparently, there is one and only one nonzero dimension in r. Note that the Regionlets object detector is a combination of N weak classifiers:

$$H = \sum_{i=1}^{N} h_i. \tag{11}$$

Thus by concatenating these binary vectors from all weak classifiers, the detection model naturally produces $8N$ dimensional sparse vectors, denoted as $R = (r_1^T, r_2^T, \ldots, r_N^T)^T$. It serves as the feature vector R_m in Equation (9). Intuitively, each Regionlets feature f_i has 8 options to vote for the actual object location depending on the binarized feature vector r_i. Learning the weight vector V in Equation (9) is to jointly determine the votes (regression coefficients) in 8 different scenarios for all Regionlets features.

The sparse binary features extracted from regionlets are very high dimensional. We observed significant over-fitting problem if there are not enough training samples. To avoid over-fitting during training, we randomly sample 80k bounding boxes around ground truth objects to train the localization model.

Discussion The supervised descent search is designed to search more object candidates in a principled way to increase the detection rate, and a following discriminative visual model (Regionlets detector) is mandatory to determine the detection scores of new locations. Regionlets Re-localization is only used to predict the accurate object location. There is no detector followed to evaluate the new location as in the supervised search. Thus it adjusts the detection to a more precise location without changing the detection score. In contrast, using the object detector to re-evaluate the detection score decreases the performance. Because the newly predicted location usually gives lower detection score which causes the predicted location being eliminated in the post non-max suppression process. To summarize, the role of supervised descent search is to find objects based on detections with coarse locations. Regionlets Re-localization is conducted on fine detections from supervised descent search. It aims at further improvement in accurate localization based on reasonable good localizations from supervised descent search. Leaving out any of these two schemes would significantly hurt the detection performance according to our observation.

3 Experiments

We evaluate the proposed detection framework with the Regionlets detector [16] on the PASCAL VOC2007 dataset and a self-collected car dataset. Our collected car dataset contains 5559 images (17501 cars) for training and 3893 images (12546 cars) for testing. We use the average precision (AP) and mean average precision (mAP) as performance measurement. We first analyze the performance of location relaxation search detection, followed with quantitative results of Regionlets Re-localization.

3.1 Location Relaxation Search

In the training phase of supervised descent, our starting points include the one from the Regionlets [16] confident detection and a set of random perturbations from the groundtruth. We found adding such starting points with perturbation samples to be necessary for a stable training. In testing phase, it always starts from Regionlets coarse detections. In this subsection, Regionlets [16] is used as a baseline for performance comparison to better understand the location relaxation search. We first study the performance of the location relaxation search with different bottom-up object proposals. Then we choose the best bottom-up setting for a thorough performance evaluation.

Effects of bottom-up object proposal The top-down search strategy is evaluated on bottom-up object hypotheses using several different settings based on 1) the color space used for over-segmentation. 2) the algorithm parameter used for over-segmentation, 3) the similarity functions (defined in Section 2.1) used for generating object proposals.

We use the graph-based image segmentation proposed by Felzenszwalb *et al.* [17](denoted as F-g) with the scale parameter k = 50 or k=100 to capture both small and large regions. Two color space are investigated in our experiments, *i.e.*, the RGB color space and the Lab color space. Following Section 2.1, the four different similarity measures used are color similarity S_c, size similarity S_s, texture similarity S_t, and fill similarity

Table 1: Cues used to generate object hypotheses. The last column shows average number of object hypotheses generated per image based on these cues.

#cues	Color space	Segmentation	Similarity	#object hypotheses
1	RGB	F-g (k=50)	(S_c, S_t, S_s, S_f)	955
2	RGB	F-g (k=50)	(S_c, S_t, S_s, S_f), (S_t, S_s, S_f)	1454
4	RGB	F-g (k=50, k=100)	(S_c, S_t, S_s, S_f), (S_t, S_s, S_f)	2045
8	RGB, Lab	F-g (k=50, k=100)	(S_c, S_t, S_s, S_f), (S_t, S_s, S_f)	3367

S_f. There are two levels of combination of these four similarity measurements. 1) Similarity level: combining these similarities as the final similarity measurement for merging neighbor regions. For example, (S_t, S_f) means the final similarity is the weighted summation of texture similarity and fill similarity. 2) Object hypotheses level: object proposals generated using different similarity combinations are collected together for coarse detection. The first combination does not increase the number of object proposals but it affects the neighbor merging activity. The second combination increases the total number of object proposals.

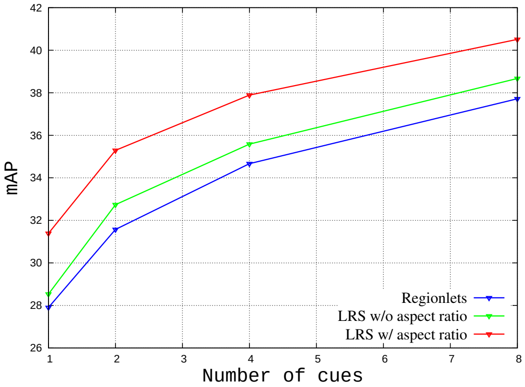

Figure 3: The detection mean average precision vs number of cues used on the PASCAL VOC 2007 dataset. **Regionlets:** the performance of regionlets without local search. **LRS w/o aspect ratio:** Location relaxation search without searching optimal object aspect ratio. **LRS w/ aspect ratio:** Location relaxation search with aspect ratio optimization.

We call one bottom-up object hypotheses generation setup as one cue. The number of object hypotheses is increased by applying different cues independently and collecting all the resulting object hypotheses. Obviously, employing more cues increases the chance of covering the target object. Figure 3 shows the detection performance of our top-down supervised search. We evaluated four detection settings which gradually

Table 2: Performance comparison with the baselines on the PASCAL VOC 2007 dataset (average precision %). LRS w/o aspect ratio: location relaxation search without optimizing aspect ratio. LRS w/ aspect ratio: location relaxation search with optimizing aspect ratio. **mAP** is the mean average precision over all the 20 categories.

AP **%**	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**table**
Regionlets [16]	53.1	49.5	16.7	25.9	16.3	49.8	64.2	37.9	16.7	39.3	44.7
LRS w/o aspect ratio	53.3	49.1	17.0	25.9	17.9	50.6	64.5	41.5	17.2	40.1	**46.8**
LRS w/ aspect ratio	**54.2**	**52.4**	**18.0**	**27.3**	**22.5**	**53.8**	**68.6**	**43.1**	**20.6**	**42.8**	45.6

	dog	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**	**mAP**
Regionlets [16]	23.2	50.4	52.7	35.6	11.7	29.5	31.3	56.1	50.0	37.7
LRS w/o aspect ratio	25.0	51.6	53.3	36.6	13.0	29.6	34.4	55.6	50.5	38.7
LRS w/ aspect ratio	**26.2**	**56.2**	**57.2**	**42.7**	**16.0**	**37.0**	**38.7**	**57.1**	**51.7**	**41.6**

increase the number of cues to get object hypotheses. The configurations are summarized in Table 1. Figure 3 presents the result including the performance of the original coarse Regiolets detection, the performance of our top-down search without optimizing object aspect ratio (a setup close to branch-and-bound, cross-talk, divide and conquer search) and the performance of our top-down search with optimizing the object aspect ratio. Although achieving promising improvement by searching only for the correct object center and scale, ignoring the aspect ratio during supervised descent search substantially suppresses the best detection accuracy we can obtain. Augmented with aspect ratio search, our top-down supervised search consistently improve the detection performance with a large margin. The more cues we used, the better performance we have. That is because our supervised descent search is targeted to find a local maximum which cannot save missing objects which are far away from the coarse detection.

Overall performance Table 2 shows the detailed performance for each object category using 8 cues for coarse detection. Without aspect ratio search, our method only improves the detection mean average precision by 1%. Adding aspect ratio to the supervised descent procedure significantly boost the performance by 3% . Note that the detection results of the Regionlets [16] detector reported here is the average precision without conducting the exhaustive local grid search (*i.e.*, only a coarse detection is applied in order to validate the effectiveness of our supervised descent search). If such exhaustive local grid search is conducted, it bumps the mAP to be 41.7% as reported in [16]. Table 2 suggests that our principled supervised descent search achieves comparable results with exhaustive dense local search.

Understanding supervised descent As aforementioned, training the supervised descent models in our experiments takes about 20 to 30 iterations to converge. Hence in testing, the supervised descent would run up to 20 to 30 steps. To better understand the supervised descent steps, we use an example to visualize how the bounding box would be evolving with the progress of the supervised descent, as illustrated in Figure 4

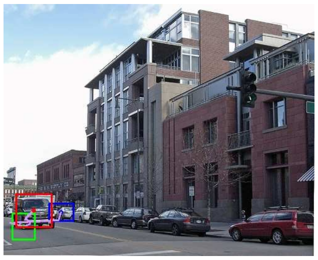

Figure 4: The trace of the searched bounding box center in supervised descent.

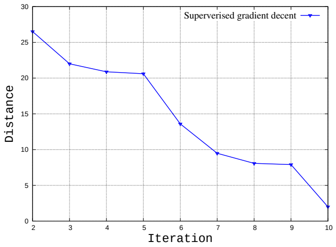

Figure 5: The distance between the searched bounding box center and the true object center in supervised descent.

and Figure 5. In Figure 4, it shows the trace of the object center (the pink curve) when supervised descent is gradually applied. The blue box is the initial coarse detection based on the bottom-up segmentation and the red box is where the search converged. We plot the distance between the searched bounding box center and the ground truth object center in Figure 5. The distance is gradually reduced in the search process. Note that this is just an illustration for understanding the process. In practice, the algorithm does not necessarily always converge to a true detection. An initialization with a false detection which is far away from any ground truth objects may result in a higher false positive during the local search. We rely on the object detector to eliminate false positives.

3.2 Regionlets Re-localization

Table 3 shows the performance of our Regionlets Re-localization approach built upon the location relaxation search on the PASCAL VOC 2007 dataset. Our localization model improves 19 out of 20 object categories. For the person category which usually has many articulated poses, our approach dramatically boosts the average precision by 6.3%. It suggests that even a dense search with the classification model does not solve the precise localization problem. This can be explained by the fact that the classification model is targeted for robust detection which accommodates inaccurate object locations, while a localization model largely complements the effort for accurate object localization.

Table 4 shows the comparison between our Regionlets Re-localization and the location prediction approach used in DPM (DPM-BB). In contrast to DPM-BB for which the improvements are within 0.5% for most of the object categories, our method yields a larger improvement, in average 2.5%. Combined with location relaxation search, our detection approach produces 44.1% mean average precision on the PASCAL VOC 2007 dataset, which to our best knowledge, is the best performance reported on this dataset without using outside data. Table 5 presents the performance comparison of our detector with recent state-of-the-art detection systems.

Table 3: Effectiveness of Regionlets Re-localization. LRS: Regionlets with location relaxation search with aspect ratio search. LRS-RR: Location relaxation search and Regionlets Re-localization

AP **%**	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mAP**
LRS	54.2	52.4	18.0	27.3	22.5	53.8	68.6	43.1	20.6	42.8	45.6	26.2	56.2	**57.2**	42.7	16.0	37.0	38.7	57.1	51.7	41.6
LRS-RR	**55.8**	**53.5**	**22.1**	**28.8**	**25.1**	**54.1**	**71.5**	**45.9**	**22.3**	**45.7**	**50.6**	**29.6**	**58.4**	55.6	**49.0**	**17.6**	**41.1**	**42.4**	**59.5**	**54.2**	**44.1**

Table 4: Performance comparison between our Regionlets Re-localization and the bounding box prediction used in deformable part base model. DPM base: base DPM performance in [19]; DPM with BB: DPM with bounding box prediction in [19]. LRS base: our base location relaxation search with aspect ratio search. LRS with RR: LRS with Regionlets Re-localization.

DPM base	DPM with BB	Improvement
26.3%	26.8%	0.5%

LRS base	LRS with RR	Improvement
41.6%	44.1%	2.5%

The detection performance of Regionlets Re-localization on the car dataset is evaluated with two different criteria. The first criterion treats a detection as true detection if it has more than 50% overlap (intersection/union) with the ground truth. The second criterion set the threshold to be 70%, which requires much better localization. As shown in Table 6, with the 0.5 overlap criterion, our Regionlets Re-localization improves the performance by 2.6%. With the 0.7 overlap criterion, it largely improves the average precision by 9.1%. This experiment strongly demonstrates that the detections are much more accurate after Regionlets Re-localization.

3.3 Run-time Speed

Our detection system runs at 4 frames per second if the over-segments are ready. The over segmentation took 1 seconds per image. However, recent approaches [25] show it is possible to obtain real-time over segmentation.

4 Conclusions

In this paper, we proposed an object detection strategy which is a combination of bottom-up object hypotheses generation and top-down local object search for generic

Table 5: Comparison with state of the arts using mAP over 20 classes. “WC” means the method utilizes context cues. We do not use any context information in our method.

	VOC 2007	Results year
DPM(WC) [4]	35.4	2008
UCI_2009 [20]	27.1	2009
INRIA_2009 [21]	28.9	2009
MIT_2010 [2]	29.6	2010
Song *etal*(WC) [22]	37.7	2011
Li *etal*(WC) [23]	35.2	2011
SS_SPM [10]	33.8	2011
Cinbis *etal*(WC) [24]	35.0	2012
Regionlets [16]	41.7	2013
Ours(LRS + RR)	**44.1**	2014

Table 6: Performance of Regionlets Re-localization on the car dataset. **0.5 ov:** A true detection must have more than 50% overlap with the ground truth. **0.7 ov:** A true detection much have more than 70% overlap with ground the truth.

	0.5 ov	**0.7 ov**
LRS	62.7%	34.8%
LRS-RR	65.3%	43.9%
Improvement	2.6%	9.1%

object detection. Our framework optimizes the object location in a full parameter space which can also search the aspect ratio of the object. The Regionlets Re-localization model complement existing classification models and can produce more precise localization, pushing even more accurate object detection.

Acknowledgements

The main part of the work was carried out when the first author was a summer intern at NEC Laboratories America in Cupertino, CA. Research reported in this publication was also partly supported by the National Institute Of Nursing Research of the National Institutes of Health under Award Number R01NR015371. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This work is also partly supported by US National Science Foundation Grant IIS 1350763, China National Natural Science Foundation Grant 61228303, GH’s start-up funds form Stevens Institute of Technology, a Google Research Faculty Award, a gift grant from Microsoft Research, and a gift grant from NEC Labs America.

References

1. Chen, G., Ding, Y., Xiao, J., Han, T.X.: Detection evolution with multi-order contextual co-occurrence. In: CVPR. (2013)
2. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: CVPR. (2010)
3. Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: ICCV. (2009)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
6. Viola, P., Jones, M.: Robust real-time object detection. IJCV (2001)
7. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: object localization by efficient subwindow search. In: CVPR. (2008)
8. Lampert, C.H.: An efficient divide-and-conquer cascade for nonlinear object detection. In: CVPR. (2010)
9. Dollar, P., Appel, R., Kienzle, W.: Crosstalk cascades for frame-rate pedestrian detection. In: ECCV. (2012)
10. Van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: ICCV. (2011)
11. Cinbis, R.G., Verbeek, J., Schmid, C.: Segmentation Driven Object Detection with Fisher Vectors. In: ICCV. (2013)
12. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE T-PAMI (2012)
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
14. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: CVPR. (2013)
15. Chang, K.Y., Liu, T.L., Chen, H.T., Lai, S.H.: Fusing generic objectness and visual saliency for salient object detection. In: ICCV. (2011)
16. Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: ICCV. (2013)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)
18. Fan, R., Chang, K., Hsieh, C., Wang, X., Jin, C.: Liblinear: A library for large linear classification. In: JMLR. (2008)
19. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE T-PAMI (2010)
20. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV. (2009)
21. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV. (2009)
22. Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. In: CVPR. (2011)
23. Li, C., Parikh, D., Chen, T.: Extracting adaptive contextual cues from unlabeled regions. In: ICCV. (2011)
24. Cinbis, R.G., Sclaroff, S.: Contextual object detection using set-based classification. In: ECCV. (2012)
25. Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. In: ECCV. (2012)