Regionlets for Generic Object Detection

Xiaoyu Wang Ming Yang Shenghuo Zhu Yuanqing Lin
NEC Laboratories America, Inc.
{xwang,myang,zsh,ylin}@nec-labs.com

Abstract

*Generic object detection is confronted by dealing with different degrees of variations in distinct object classes with tractable computations, which demands for descriptive and flexible object representations that are also efficient to evaluate for many locations. In view of this, we propose to model an object class by a cascaded boosting classifier which integrates various types of features from competing local regions, named as regionlets. A regionlet is a base feature extraction region defined proportionally to a detection window at an arbitrary resolution (*i.e. *size and aspect ratio). These regionlets are organized in small groups with stable relative positions to delineate fine-grained spatial layouts inside objects. Their features are aggregated to a one-dimensional feature within one group so as to tolerate deformations. Then we evaluate the object bounding box proposal in selective search from segmentation cues, limiting the evaluation locations to thousands. Our approach significantly outperforms the state-of-the-art on popular multi-class detection benchmark datasets with a single method, without any contexts. It achieves the detection mean average precision of 41.7% on the PASCAL VOC 2007 dataset and 39.7% on the VOC 2010 for 20 object categories. It achieves 14.7% mean average precision on the ImageNet dataset for 200 object categories, outperforming the latest deformable part-based model (DPM) by 4.7%.*

1. Introduction

Despite the success of face detection where the target objects are roughly rigid, generic object detection remains an open problem mainly due to the challenge of handling all possible variations with tractable computations. In particular, different object classes demonstrate a variable degree of deformation in images, either due to their nature, *e.g.*, living creatures like cats are generally more deformable than man-made objects like vehicles, or due to viewing distances or angles, *e.g.*, deformable objects may appear somehow rigid at a distance and even rigid objects may show larger variations in different view angles. These pose a fundamental dilemma to object class representations: on one hand, a delicate model describing rigid object appearances may hardly handle deformable objects; on the other hand, a high tolerance of deformation may result in imprecise localization or false positives for rigid objects.

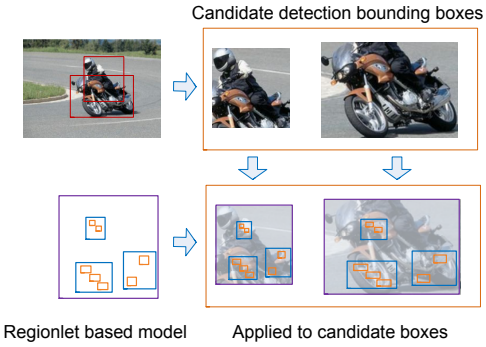

Figure 1: Illustration of the regionlet representation. Regionlet representation can be applied to candidate bounding boxes that have different sizes and aspect ratios. A regionlet-based model is composed of a number of regions (denoted by blue rectangles), and then each region is represented by a group of regionlets (denoted by the small orange rectangles inside each region).

Prior arts in object detection cope with object deformation efficiently with primarily three typical strategies. First, if spatial layouts of object appearances are roughly rigid such as faces or pedestrians at a distance, the classical Adaboost detection [26] mainly tackles local variations with an ensemble classifier of efficient features. Then a sliding window search with cascaded classifiers is an effective way to achieve precise and efficient localization. Second, the deformable part model (DPM) method [12] inherits the HOG window template matching [6] but explicitly models deformations by latent variables, where an exhaustive search of possible locations, scales, and aspect ratios are critical to localize objects. Later on, the DPM has been accelerated by coarse-to-fine search [19], branch and bound [16], and cross-talk approaches [9]. Third, object recognition

methods using spatial pyramid matching (SPM) of bag-of-words (BoW) models [17] are adopted for detection [25], and they inherently can tolerate large deformations. These sophisticated detectors are applied to thousands of object-independent candidate regions [25, 2], instead of millions of sliding windows. In return, little modeling of local spatial appearances leaves these recognition classifiers unable to localize rigid objects precisely, *e.g.*, bottles. These successful detection approaches inspire us to investigate a descriptive and flexible object representation, which delivers the modeling capacity for both rigid and deformable objects in a unified framework.

In this paper, we propose a new object representation strategy for generic object detection, which incorporates adaptive deformation handling into both object classifier learning and basic feature extraction. Each object bounding box is classified by a cascaded boosting classifier, where each weak classifier takes the feature response of a region inside the bounding box as its input and then the region is in turn represented by a group of small sub-regions, named as *regionlets*. The sets of regionlets are selected from a huge pool of candidate regionlet groups by boosting. On one hand, the relative spatial positions of the regionlets within a region and the region within an object bounding box are stable. Therefore, the proposed regionlet representation can model fine-grained spatial appearance layouts. On the other hand, the feature responses of regionlets within one group are aggregated to a one dimensional feature, and the resulting feature is generally robust to local deformation. Also, our regionlet model is designed to be flexible to take bounding boxes with different sizes and aspect ratios. Therefore our approach is ready to utilizes the selective search strategy [25] to evaluate on merely thousands of candidate bounding boxes rather than hundreds of thousands (if not millions) of sliding windows as in the exhaustive search.

An illustration of the regionlet representation is shown in Figure 1, where the regionlets drawn as orange boxes are grouped within blue rectangular regions. The regionlets and their groups for one object class are learned in boosting with stable relative positions to each other. When they are applied to two candidate bounding boxes, the feature responses of regionlets are obtained at the their respective scales and aspect ratios without enumerating all possible spatial configurations.

The major contribution of this paper lies in two-fold. 1) It introduces the regionlet concept which is flexible to extract features from arbitrary bounding boxes. 2) The regionlet-based representation for an object class, which not only models relative spatial layouts inside an object but also accommodates variations especially deformations by the regionlet group selection in boosting and the aggregation of feature responses in a regionlet group. As validated in the experiment, the proposed representation adaptively models a varying degree of deformation in diverse object classes.

2. Related Work

Object detection is arguably an indispensable component for most of vision tasks, and it has achieved prominent successes for some specific targets such as faces [26, 14] and pedestrians [6, 24, 28, 4, 8]. Complete survey of object detection is certainly beyond the scope of this paper. Instead, we briefly review related generic object detection methods that do not focus on a particular type of object.

One of the most influential methods in generic object detection is the deformable part model (DPM) [12] and its extensions [12, 11, 31, 19]. The DPM object detector consists of a root filter and several part filters. Deformations among parts are inferred with latent variables. Since the resolutions of the object templates are fixed, an exhaustive sliding window search [12] is required to find objects at different scales and different aspect ratios. The exhaustive search can be accelerated by more efficient search [16, 15, 19, 9, 11, 4, 30]. In contrast, our regionlet-based detection handles object deformation directly in feature extraction, and it is flexible to deal with different scaling and aspect ratios without the need of an exhaustive search.

Recently, a new detection strategy [2, 25, 20] is to use multi-scale image segmentation to propose a couple thousands of candidate bounding boxes for each image and then the object categories of the bounding boxes are determined by strong object classifiers, *e.g.*, using bag-of-words (BoW) model with spatial pyramid match (SPM) [25]. Because the BoW models ignore spatial relations among local features, they are able to tolerate large deformations. However, because of the lack of local spatial relations, they may not localize rigid object precisely. Our method borrows the candidate window proposing procedure in [25] to speed up the detection, however, it is fundamentally different from [25] in both feature extraction and classifier learning.

Contexts from local or global appearance have been explored to improve object detection [23, 27, 8, 5, 18, 29]. We do not use any context cues in this paper and leave it as a future work.

3. Regionlets for Detection

Object detection is composed of two key components: determing where the candidate locations are in images and discerning whether they are the objects of interests. Beyond the straightforward exhaustive search of all locations, our regionlet detection approach screens the candidate windows derived from the selective search [25]. For selective search, given an image, it first over-segments the image into superpixels, and then those superpixels are grouped in a bottom-up manner to propose candidate bounding boxes. The work in [25] shows that such proposing bounding boxes, about

1,000∼2,000 each image, achieve very high recall. After this, the task of detection boils down to extraction of an appropriate object representation on each proposed box and learning of a scoring function to rank the boxes. To that end, we introduce regionlet features for each candidate bounding box. In our proposed method, we construct a largely over-complete regionlet feature pool and then design a cascaded boosting learning process to select the most discriminative regionlets for detection.

In the following, Section 3.1 describes what the regionlets are and explains how they are designed to handle deformation. Section 3.2 presents how to construct a largely over-complete regionlet pool and learn a cascaded boosting classifier for an object category by selecting the most discriminative regionlets.

3.1. Regionlets

3.1.1 Regionlet definition

In object detection, an object category is essentially defined by a classifier where both object appearance and the spatial layout inside an object shall be taken into account. For simplicity, appearance features are mostly extracted from some *rectangular* sub-regions within an object, which we refer as feature extraction regions in the paper. Features extracted from a small region often provide a good localization ability, but are vulnerable to variations; a big region tends to tolerate more variations but may not be sensitive enough for accurate localization. When large variations especially deformations occur, a large rectangle region may not be appropriate for extracting descriptive features of an object. Because some parts or the regions may not be informative or even distractive. This motivates us to define sub-parts of a region, *i.e.*, the regionlets, as the basic units to extract appearance features, and organize them into small groups which are more flexible to describe distinct object categories with different degrees of deformation.

We would like to introduce the regionlets with an example illustrated in Figure 2. The first column in Figure 2 shows three samples of a person that are the target object to detect and they are cropped by black bounding boxes in the second column. A rectangle feature extraction region inside the bounding box is denoted as R, which will contribute a weak classifier to the boosting classifier. Within this region R, we further spot some small sub-regions (*e.g.*, r_1,r_2 and r_3) and define them as a group of regionlets. We employ the term *regionlet*, because the features of these sub-regions will be aggregated to a single feature for R, and they are below the level of a standalone feature extraction region in an object classifier. In short, in the proposed method, a detection bounding box is represented by a number of regions, each of which is composed of a small set of regionlets.

This example also illustrates how regionlets are designed to handle deformation. Hand, as a supposingly informative

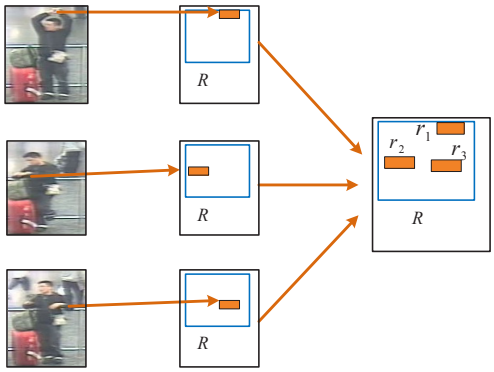

Figure 2: Illustration of the relationship among a detection bounding box, a feature extraction region and regionlets. A feature extraction region R, shown as a light blue rectangle, is cropped from a fixed position from 3 samples of a person. Inside R, several small sub-regions denoted as r_1, r_2 and r_3 (in orange small rectangles) are the *regionlets* to capture the possible locations of the hand for person detection.

part for a person, may appear at different locations within the bounding box of a person. If we extract the feature for a hand from the whole region R which roughly covers the possible locations of the hand, the appearance of some non-hand regions on the torso or background clearly are also included in the feature. An ideal deformation handling strategy is to extract features only from the hand region in all three cases. To that end, we introduce three regionlets inside R (In general, a region can contain many regionlets. Here "three" is mainly for illustration purpose). Each regionlet r covers a possible location of hand. Then only features from the regionlets are extracted and aggregated to generate a compact representation for R. Irrelevant appearance from backgrounds are largely discarded. More regionlets in R will increase the capacity to model deformations, *e.g.*, hand surely may appear in more positions than three. On the other hand, rigid objects may only require one regionlet from a feature extraction region.

3.1.2 Region feature extraction

Feature extraction from R takes two steps: 1) extracting appearance features, *e.g.*, the HOG [6] and LBP descriptors [1] from each regionlet respectively; and 2) generating the representation of R based on regionlets' features. The first step is straightforward. For the second step, we define a permutation invariant feature operation on features extracted from regionlets, and such an operation also assumes an exclusive relation among regionlets. Let's denote $T(R)$ as the feature representation for region R, $T(r_j)$ as the feature extracted from the j^{th} regionlet r_j in R, then the

operation is defined as following:

$$T(R) = \sum_{j=1}^{N_R} \alpha_j T(r_j), \quad (1)$$

$$\text{subject to } \alpha_j \in \{0,1\}, \sum_{j=1}^{N_R} \alpha_j = 1,$$

where N_R is the total number of regionlets in region R, α_j is a binary variable, either 0 or 1. This operation is permutation invariant, namely, the occurrence of the appearance cues in any of regionlets is equivalent, which allows deformations among these regionlet locations. The operation also assumes the exclusiveness within a group of regionlets, namely, one and only one regionlet will contribute to the region feature representation. The exclusive assumption is that when deformation occurs, the discriminative sub-region appears at only one position in a specific training/testing sample.

In our framework, we simply apply max-pooling over regionlet features. So Eq. 1 is instantiated as:

$$T(R) = \max_j T(r_j). \quad (2)$$

The max-pooling happens for each feature dimension independently. For each regionlet r_j, we first extract low-level feature vectors, such as HOG or LBP histograms. Then, we pick a 1D feature from the same dimension of these feature vectors in each regionlet and apply Eq. 2 to form the feature for region R. We have millions of such 1D features in a detection window and the most discriminative ones are determined through a boosting type learning process (to be described in Section 3.2.2).

Figure 3 illustrates the process to extract $T(R)$, the 1-D feature for a region R. Here we again use the example in Figure 2, where the blue region R is the one covering the variation of hand locations. Assuming the first dimension of the concatenated low-level features is the most distinctive feature dimension learned for hand, we collect this dimension from all the three regionlets and represent $T(R)$ by the strongest feature response from the top regionlet.

3.1.3 Regionlets normalized by detection windows

In this work, the proposed regionlet representations are evaluated on the candidate bounding boxes derived from selective search approach [25]. In principle, they are also applicable for sliding windows. The selective search approach first over-segments an images into superpixels, and then the superpixel are grouped in a bottom-up manner to propose some candidate bounding boxes. This approach typically produces 1000 to 2000 candidate bounding boxes for an object detector to evaluate on, compared to millions of windows in an exhaustive sliding window search.

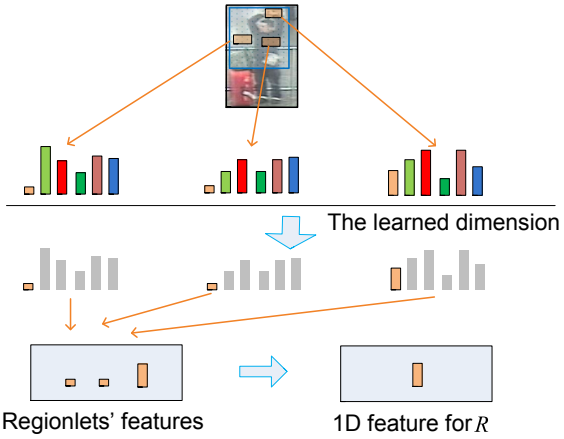

Figure 3: Example of regionlet-based feature extraction.

However, these proposed bounding boxes have arbitrary sizes and aspect ratios. As a result, it is not feasible to use template regions (or template regionlets) with fixed absolute sizes that are widely used in sliding window search. We address this difficulty by using the *relative* positions and sizes of the regionlets and their groups to an object bounding box. Figure 4 shows our way of defining regionlets in contrast to fixed regions with absolute sizes. When using a sliding window search, a feature extraction region is often defined by the top-left (l,t) and the bottom-right corner (r,b) *w.r.t.* the anchor position of the candidate bounding box. In contrast, our approach normalizes the coordinates by the width w and height h of the box and records the relative position of a region $(l',t',r',b') = (\frac{l}{w}, \frac{t}{h}, \frac{r}{w}, \frac{b}{h}) = R'$. These relative region definitions allow us to directly evaluate the regionlets-based representation on candidate windows at different sizes and aspect ratios without scaling images into multiple resolutions or using multiples components for enumerating possible aspect ratios.

3.2. Learning the object detection model

We follow the boosting framework to learn the discriminative regionlet groups and their configurations from a huge pool of candidate regions and regionlets.

3.2.1 Regions/regionlets pool construction

Deformation may occur at different scales. For instance, in person detection, deformation can be caused by a moving finger or a waving hand. A set of small regionlets that is effective to capture finger-level deformation may hardly handle deformation caused by hand movements. In order to deal with diverse variations, we build a largely over-complete pool for regions and regionlets with various positions, aspect ratios, and sizes. Before regionlet learning,

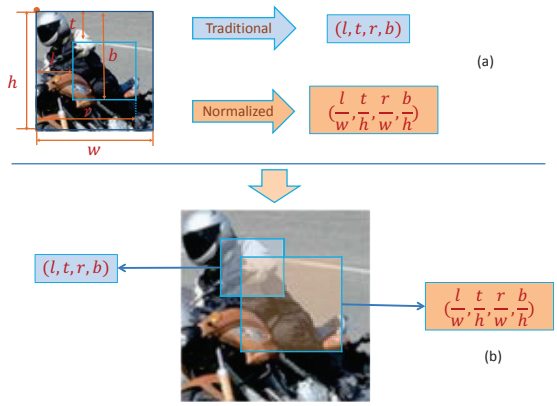

Figure 4: Relative regions normalized by a candidate window that are robust to scale and aspect ratio changes.

a region R' or a regionlet r' are not applied to a detection window yet, so we call R' a feature region prototype and r' a regionlet prototype.

We first explain how the pool of region feature prototypes is constructed. Using the definition in Section 3.1.3, we denote the 1D feature of a region relative to a bounding box as $R' = (l', t', r', b', k)$ where k denotes the kth element of the low-level feature vector of the region. R' represents a feature prototype. The region pool is spanned by $\mathcal{X} \times \mathcal{Y} \times \mathcal{W} \times \mathcal{H} \times \mathcal{F}$, where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the space of horizontal and vertical anchor position of R in the detection window, $\mathcal{W}$ and $\mathcal{H}$ are the width and height of the feature extraction region R', and $\mathcal{F}$ is the space of low-level feature vector (*e.g.*, the concatenation of HOG and LBP). Enumerating all possible regions is impractical and not necessary. We employ a sampling process to reduce the pool size. Algorithm 1 describes how we sample multiple region feature prototypes. In our implementation, we generate about 100 million feature prototypes.

Afterwards, we propose a set of regionlets with random positions inside each region. Although the sizes of regionlets in a region could be arbitrary in general, we restrict regionlets in a group to have the identical size because our regionlets are designed to capture the same appearance in different possible locations due to deformation. The sizes of regionlets in different groups could be different. A region may contain up to 5 regionlets in our implementation.

So the final feature space used as the feature pool for boosting is spanned by $\mathcal{R} \times \mathcal{C}$, where $\mathcal{R}$ is the region feature prototype space, $\mathcal{C}$ is the configuration space of regionlets. Therefore, we augment a region feature prototype $R' = (l', t', r', b', k, c)$ with a regionlet configuration c.

Algorithm 1: Generation of region feature prototypes

Input: Region width step s_w and height step s_h; maximum width W and height H of region prototypes; horizontal step p_x and vertical step p_y for the region anchor position; minimum width w_{min} and height h_{min} of region prototypes; the number of features N to extract from one region

```
begin
    w ← w_min, h ← h_min, i ← 0
    for w < W do
        h ← h_min
        for h < H do
            h ← h + s_h
            l ← 0, t ← 0
            for l < W − w do
                t ← 0
                for t < H − h do
                    for k=1,...N do
                        r ← l + w, b ← t + h
                        R' = (l/w, t/h, r/w, b/h, k)
                        R ← R ∪ {R'}
                    t ← t + p_y, i ← i + 1
                l ← l + p_x
            h ← h + s_h
        w ← w + s_w
```

Output: Region feature prototype pool $\mathcal{R}$

3.2.2 Training with boosting regionlet features

We use RealBoost [22] to train cascaded classifiers for our object detector. One boosting classifer consists of a set of selected weak classifiers. Similar to [14], we define the weak classifier using a lookup table:

$$h(x) = \sum_{o=1}^{n-1} v^o \mathbb{1}(B(x) = o), \tag{3}$$

where $h(x)$ is a piece-wise linear function defined by a lookup table, v^o is the table value for the oth entry, $B(x)$ quantizes the feature value x into a table entry, and $\mathbb{1}(\cdot)$ is an indicator function. In each round of the training, v^o is computed based on the sample weight distribution as $v^o = \frac{1}{2}\ln(\frac{U_+^o}{U_-^o})$, where U_+^o is the summation of the weights of the positive examples whose feature values fall into the oth entry of the table. The U_-^o is defined in a similar manner for the weights of negative examples.

Let's denote Q as a candidate bounding box, $R'(Q)$ as a rectangular region in Q, and $T(R'(Q))$ as the one-dimensional feature computed on $R'(Q)$ (similar notation as in Eq. 1). Substituting x in Eq. 3 with the extracted feature, we can get the weak classifier in the tth round of train-

ing for the bounding box Q:

$$h_t(T(R'(Q))) = \sum_{o=1}^{n-1} v_t^o \mathbb{1}(B_t(T(R'(Q))) = o), \quad (4)$$

where v_t^o is the table value of the oth entry at the tth round of training. Then, for each boosting classifier, the learning process obtains a set of weak classifiers H for separating the positive samples from negative ones:

$$\begin{aligned} H(Q) &= \sum_{t=1}^{T} \beta_t h_t(T(R'_{i_t}(Q))) \\ &= \sum_{t=1}^{T} \beta_t h_t \left(\sum_{j=1}^{N_{i_t}} \alpha_{i_t,j} T(r'_{i_t,j}(Q)) \right), \end{aligned} \quad (5)$$

where i_t is the index of the region selected in the tth round of training, N_{i_t} is the total number of regionlets in R_{i_t}, and β_t is the weight of the selected weak classifier. The classification result of the candidate bounding box Q is determined by the final round of cascade if it passes all previous ones, and it is expressed as $f(Q) = sign(H^*(Q))$ where H^* denotes the last stage of cascade.

In each cascade training, we generate 100 millions of candidate weak classifiers. To feed into memory, a reduced set of 20K weak classifiers are sampled uniformly. The training terminates once the error rates (37.5% for negative and 1% for positive samples) are achieved except the last cascade. The last round stops until it collects 5K weak classifiers. The training results in 6-7 cascades and 5K-6K weak classifiers.

Given a test image, we first propose a number of candidate bounding boxes using the selective search [25]. Then, each candidate bounding box is passed along the cascaded classifiers learned in the boosting process. Because of early rejections, only a small number of candidate bounding boxes reach the last stage of the cascade. Therefore, except the time spent on proposing bounding boxes, our method yields a very fast testing speed.

4. Experiments

Following most of recent work on generic object detection, we evaluate our object detection performance on the challenging PASCAL VOC 2007 and VOC 2010 datasets. The PASCAL VOC datasets contain 20 categories of objects including rigid objects such as cars and deformable objects like cats. They are popular benchmark datasets for generic multi-class object detection. Detection performance of each category is measured by the average precision. The overall performance is reported by the mean of average precisions (mAP) over all classes. The proposed method is further validated on the much larger ImageNet object detection dataset (ILSVRC2013) [21], including 200 object categories. We also use mAP as the metric for performance evaluation.

4.1. Experiment on PASCAL VOC datasets

In our implementation of regionlet-based detection, we utilize the selective search bounding boxes from [25] to train our detector. The HOG [6], LBP [1] and covariance features [24] are adopted as candidate features for the regionlets. To validate the advantages of the proposed approach, we compare it with three baselines: deformable part-based model [12] which is one of the most effective sliding window based detectors, and two recent approaches based on selective search [3, 25].

Accuracy: Table 1 presents the performance on the PASCAL VOC 2007 dataset, from which we learn some interesting insights from the comparison between the first two baselines. In [12] features from each small 8×8 patches are aligned inside the detection window for the root filter, allowing local deformation for parts with misplacement costs. While [25] builds spatial pyramids over detection windows. So [12] enforces a strict spatial constraint in the root filter but allows for small local deformations, while [25] ignores spatial configuration to a large extent. From Table 1, we observe that [12] is excellent at detecting rigid objects (such as bus and car) and objects with well defined contours but abundant local deformations (such as horse and person). In contrast, [25] performs better for objects with significant global deformations such as cat, cow and sheep, as expected. It is very interesting that [25] significantly outperforms [12] in the aeroplane and tvmonitor categories, both of which seem to be rigid objects. After taking a close look at the data we figure out it is because these categories have very diverse viewpoints or rich sub-categories.

Our method significantly outperforms all three baselines, as we won 16 out of 20 categories. Our approach performs decently for both rigid objects and those objects with local or global deformations. Compared to [25], our edge comes from our regionlet representation encoding object's spatial configuration. Compared to [12], our improvement on accuracy is led by the joint deformation and misalignment handling powered by the regionlets representation with multiple resolution features. If we limit the number of regionlets in a region to be 1, our method obtained a mean average precision of 36.8%. Allowing multiple regionlets consistently improves the object detection accuracy for each class and pushes the number to 41.7%, which to our best knowledge, is the best performance reported on the VOC 2007. The results of [3] are based on the old version of DPM. Direct comparsion to other methods may not be fair. It combines the output of [12] and objectness measurement which gives 0.6% improvement. It suggests selective search itself does not benefit detection too much in terms of accuracy. A similar conclusion has been drawn in [25].

Table 3 presents the effectiveness of different feature configurations. Figure 5 shows the average number of regionlets used per region for each class. Deformable objects

Table 1: Performance comparison with the baselines on the PASCAL VOC 2007 dataset (average precision %). **DPM:** deformable part based model. **SS_SPM:** selective search with spatial pyramid features. **Regionlets-S:** our regionlets approach with a single regionlet per region. **Regionlets-M:** our regionlets approach allowing for multiple regionlets per region. **mAP** is the mean average precision over all the 20 categories.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM [12][1]	33.2	**60.3**	10.2	16.1	**27.3**	54.3	58.2	23.0	**20.0**	24.1	26.7	12.7	**58.1**	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
SS_SPM [25][2]	43.5	46.5	10.4	12.0	9.3	49.4	53.7	39.4	12.5	36.9	42.2	26.4	47.0	52.4	23.5	12.1	29.9	36.3	42.2	48.8	33.8
Objectness [3]	28.6	54.5	1.1	14.2	26.7	42.0	50.2	18.2	16.5	17.5	26.2	7.7	46.8	39.6	36.2	11.6	14.1	23.1	34.8	39.2	27.4
Regionlets-S	50.8	44.6	17.0	23.5	16.7	48.9	67.6	39.1	16.5	32.4	44.0	18.9	52.1	46.6	36.6	13.8	33.8	27.6	55.5	50.4	36.8
Regionlets-M	**54.2**	52.0	**20.3**	**24.0**	20.1	**55.5**	**68.7**	**42.6**	19.2	**44.2**	**49.1**	**26.6**	57.0	**54.5**	**43.4**	**16.4**	**36.6**	**37.7**	**59.4**	**52.3**	**41.7**

Table 2: Performance comparison with the baselines on the PASCAL VOC 2010 dataset (average precision %).

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM [12]	45.6	**49.0**	11.0	11.6	**27.2**	50.5	43.1	23.6	**17.2**	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	29.6
SS_SPM [25]	58.2	41.3	19.2	14.0	14.3	44.8	36.7	48.8	12.9	28.1	28.7	**39.4**	**44.1**	52.5	25.8	14.1	38.8	**34.2**	43.1	42.6	34.1
Regionlets-M	**65.0**	48.9	**25.9**	**24.6**	24.5	**56.1**	**54.5**	**51.2**	17.0	**28.9**	**30.2**	35.8	40.2	**55.7**	**43.5**	**14.3**	**43.9**	32.6	**54.0**	**45.9**	**39.7**

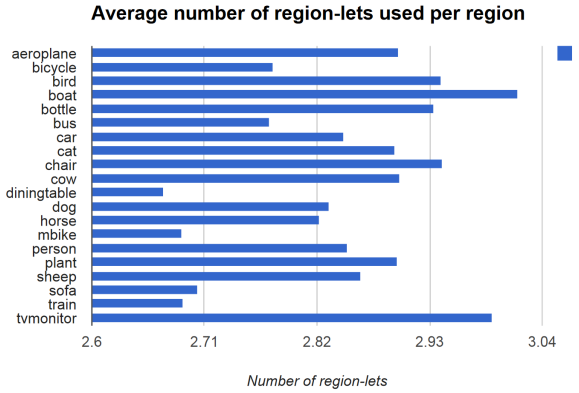

Figure 5: Statistics of number of regionlets used for each class. Deformable objects generally prefer more regionlets.

such as bird, cat, sheep, person and dog, *etc*. prefer more regionlets than rigid objects like bicycle, bus, diningtable, motorbike, sofa and train. An interesting yet consistent phenomenon has been observed for rigid objects like aeroplane and tvmonitor, as in the comparison of [12] and [25]: our algorithm selects even more regionlets than those for other deformable objects. We speculate the regionlets in these two cases may help to handle misalignment due to multiple viewpoints and sub-categories. Table 2 shows our detection performance compared with baselines on the VOC 2010 dataset. Our regionlet approach again achieves the best mean average precision. Table 4 compares our approach with other state-of-the-arts methods on both VOC 2007 and VOC 2010 datasets in terms of mean average precision over the 20 categories. While the accuracies of [10] and [23] are close to ours, both of them are based on the DPM [12] with different ways to explore the context information. In contrast, our approach does not utilize any context cues and the context information will likely further benefit our detection approach.

Table 3: Performance of different features on the PASCAL VOC 2007 dataset.

Feature	**HOG**	**LBP**	**COV**	**HOG+LBP**	**HOG+COV**	**LBP+COV**
mAP	35.1	33.5	33.7	38.47	37.3	37.7

Table 4: 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	**VOC 2010**	Results year
DPM(WC) [12]	35.4	33.4	2008
UCI_2009 [7]	27.1	N/A	2009
INRIA_2009 [13]	28.9	N/A	2009
NLPR(WC) [10]	N/A	36.8	2010
MITUCLA(WC) [10]	N/A	36.0	2010
UVA [10]	N/A	32.9	2010
MIT_2010 [31]	29.6	N/A	2010
Song *et al.* (WC) [23]	37.7	36.8	2011
Li *et al.* (WC) [18]	35.2	N/A	2011
SS_SPM [25]	33.8	34.1	2011
Cinbis *et al.* (WC) [5]	35.0	N/A	2012
Ours (Regionlets)	**41.7**	**39.7**	2013

[1]We obtained the latest results from the author's website http://www.cs.berkeley.edu/~rbg/latent/index.html.

[2]We read the performance from the figure in [25].

Speed: We conducted our experiments on 12-core Xeon 2.1GHz blade servers. Multiple threading is utilized to speed up the training and testing procedure. Training for one category on a single machine takes 4 to 8 hours, depending on how difficult the category is. The detection runs at 50 frames per second on a server or 5 frames per second using a single core.

4.2. Experiment on ImageNet Object Detection

We demonstrate the scalability of the regionlet-based detection on the ImageNet object detection task, using the training set for training and validation set for testing. The DPM performance is obtained by applying the DPM v5 (version 5) code on the ImageNet[1]. The mAP over the 200 object categories is reported in Table 5. Our approach outperforms the latest DPM by 4.7%(mAP) across the 200 categories which is indeed a significant improvement. The objects in ImageNet have large variations in terms of deformation and sub-categories. Due to the regionlets representation and enforced spatial layout learning, our proposed approach performs perfectly in both cases.

Table 5: Comparison with DPM on the ImageNet dataset.

	ImageNet 2013 (mAP)	Results year
DPM v5 [12]	10.0	2013
Ours (Regionlets)	**14.7**	2013

5. Conclusion and Future Work

In this paper, we propose a regionlet-based approach for generic object detection. Regionlets provide a radically different way to model object deformation compared to existing BoW approaches with selective search and DPM approaches. Our regionlet model can well adapt itself for detecting rigid objects, objects with small local deformations as well as long-range deformations. Validated on the challenging PASCAL VOC datasets and ImageNet object detection dataset, the proposed regionlet approach demonstrates superior performance compared to the existing approaches.

As a future work, we plan to improve the way of proposing bounding boxes in term of recall and speed. Indeed, this step currently is the computation bottleneck in our method. Second, we will investigate how the context information can be integrated into the boosting learning process for further improving detection performance.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *ECCV*, 2004.

[1]We thank our intern Miao Sun from Univeristy of Missouri for evaluating the DPM performance on ImageNet.

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

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

[4] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012.

[5] R. G. Cinbis and S. Sclaroff. Contextual object detection using set-based classification. In *ECCV*, 2012.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

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

[8] Y. Ding and J. Xiao. Contextual boost for pedestrian detection. In *CVPR*, 2012.

[9] P. Dollar, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, 2012.

[10] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[11] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

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

[14] C. Huang, H. Ai, B. Wu, and S. Lao. Boosting nested cascade detector for multi-view face detection. In *ICPR*, 2004.

[15] C. H. Lampert. An efficient divide-and-conquer cascade for nonlinear object detection. In *CVPR*, 2010.

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

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[18] C. Li, D. Parikh, and T. Chen. Extracting adaptive contextual cues from unlabeled regions. In *ICCV*, 2011.

[19] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *CVPR*, 2011.

[20] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011.

[21] O. Russakovsky, J. Deng, J. Krause, A. Berg, and L. Fei-Fei. Large scale visual recognition challenge 2013 (ILSVRC2013). http://www.image-net.org/challenges/LSVRC/2013/, 2013.

[22] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, 1999.

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

[24] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, 2007.

[25] K. E. A. Van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011.

[26] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2001

[27] X. Wang, X. Bai, W. Liu, and L. J. Latecki. Feature context for image classification and object detection. In *CVPR*, 2011.

[28] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *ICCV*, 2009.

[29] J. Zhang, K. Huang, Y. Yu, and T. Tan. Boosted local structured HOG-LBP for object localization. In *CVPR*, 2011.

[30] Z. Zhang, J. Warrell, and P. H. S. Torr. Proposal generation for object detection using cascaded ranking SVMs. In *CVPR*, 2011.

[31] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.