Deep Regionlets for Object Detection

Hongyu Xu[1]*, Xutao Lv[2], Xiaoyu Wang[3], Zhou Ren[2], Navaneeth Bodla[1] and Rama Chellappa[1]
[1]Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS
University of Maryland, College Park, MD, USA
[2]Snap Inc. [3]Intellifusion
[1]hyxu@umiacs.umd.edu [2]lvxutao@gmail.com [3]fanghuaxue@gmail.com
[2]zhou.ren@snap.com [1]nbodla@umiacs.umd.edu [1]rama@umiacs.umd.edu

Abstract

In this paper, we propose a novel object detection framework named "Deep Regionlets" by establishing a bridge between deep neural networks and conventional detection schema for accurate generic object detection. Motivated by the advantages of regionlets on modeling object deformation and multiple aspect ratios, we incorporate regionlet into an end-to-end trainable deep learning framework. The deep regionlets framework consists of a region selection network and a deep regionlet learning module. Specifically, given a detection bounding box proposal, the region selection network serves as a guidance on where to select regions to learn the features from. The regionlet learning module focuses on local feature selection and transformation to alleviate local variations. To this end, we ***first*** *realize* ***non-rectangular*** *region selection within the detection framework to accommodate variations in object appearance. Moreover, we further design a "gating network" within the regionlet leaning module to enable soft regionlet selection and pooling. The Deep Regionlets framework is trained end-to-end without additional efforts. We perform ablation studies on its behavior and conduct extensive experiments on the PASCAL VOC and Microsoft COCO dataset. The proposed framework outperforms state-of-the-art algorithms, such as RetinaNet and Mask R-CNN, even without additional segmentation labels.*

1. Introduction

Generic object detection has been extensively studied in computer vision community over the decades [20, 3, 42, 14, 15, 36, 5, 26, 41, 40, 7, 10, 44, 12, 45] due to its appeal to both academic research explorations as well as commercial applications. Given an image of interest, the goal of object detection is to predict the locations of objects and classify them at the same time. The key challenge of the object detection task is to handle variations in object scale, pose, viewpoint and even part deformations when generating the bounding boxes for specified object categories.

Numerous methods have been proposed based on handcrafted features (*i.e.* HOG [7], LBP [1], SIFT [30]). These approaches usually involve an exhaustive search for possible locations, scales and aspect ratios of the object, by using the sliding window approach. However, Wang *et al*.'s [41] regionlet-based detection framework has gained a lot of attention as it provides the flexibility to deal with different scales and aspect ratios without performing an exhaustive search. It first proposed the concept of ***regionlet*** by defining a three-level structural relationship: candidate bounding boxes (sliding windows), regions inside the bounding box and groups of regionlets (sub-regions inside each region). It operates by directly extracting features from regionlets in several selected regions within an arbitrary detection bounding box and performs (max) pooling among the regionlets. Such a feature extraction hierarchy is capable of dealing with variable aspect ratios and flexible feature sets, which leads to improved learning of robust feature representation of the object for region-based object detection.

Putting this work in context, recently, deep learning has achieved significant success on computer vision tasks in many aspects such as image classification [23, 18], semantic segmentation [29] and object detection [14] using the deep convolutional neural network (DCNN) architecture. Despite the excellent performance of deep learning-based detection framework, most network architectures [36, 5, 28] do not take advantage of successful conventional ideas such as deformable part-based model (DPM) or *regionlets*. Those methods have been effective for modeling object deformation, sub-categories and multiple aspect ratios. Recent advances [33, 6, 32] have achieved promising results by extending the conventional DPM-based detection methodology with the deep neural network architectures.

These observations motivate us to establish a bridge between deep convolutional neural network and conventional

*Work started during an internship at Snap Research

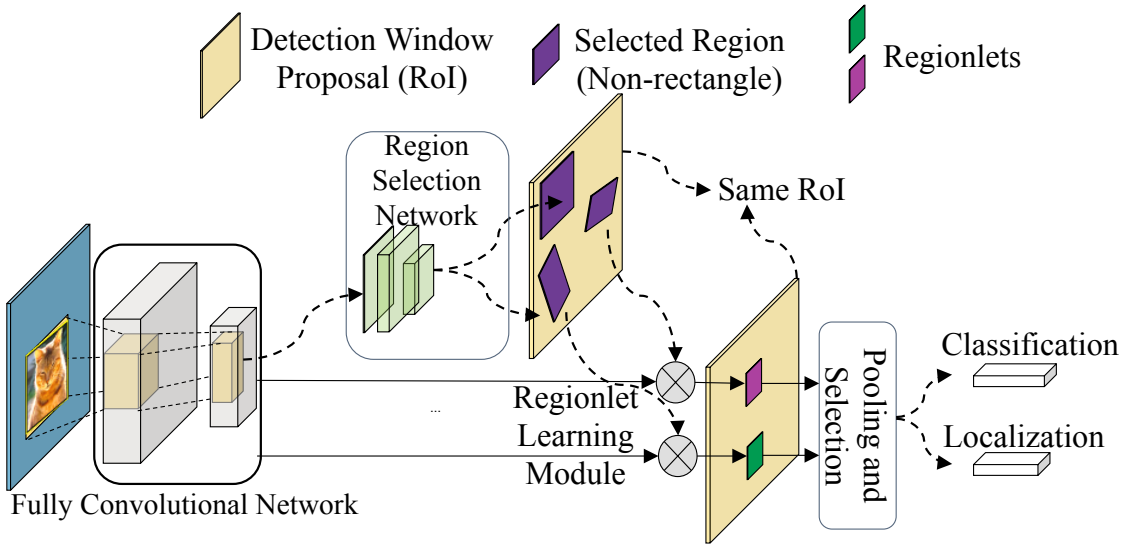

Figure 1: Architecture of the Deep Regionlets detection framework. It consists of a region selection network (RSN) and a deep regionlet learning module. RSN performs *non-rectangular* region selection from the detection window proposal generated by the region proposal network. Deep regionlet learning module learns the regionlets through a spatial transformation and a gating network. The entire pipeline is end-to-end trainable. For better visualization, region proposal network is not displayed here.

object detection schema. In this paper, we incorporate the conventional Regionlet method into an end-to-end trainable deep learning framework. Despite being able to handle the arbitrary bounding boxes, several drawbacks arise when directly integrating the regionlet methodology into the deep learning framework. First, in [41], Wang *et al*. proposed to learn cascade object classifiers after hand-crafted feature extraction in each regionlet. However, end-to-end learning is not feasible in such framework. Second, regions in regionlet-based detection have to be rectangular, which does not effectively model the deformation in the object with variable shapes. Moreover, both regions and regionlets are fixed after training is completed.

To this end, we propose a novel object detection framework named "Deep Regionlets" to introduce deep learning framework to the traditional regionlet method [41]. The overall design of the proposed detection system is illustrated in Figure 1. It consists of a region selection network and a deep regionlet learning module. The region selection network (RSN) performs *non-rectangular* region selection from the detection window proposal[1] (RoI) to address limitations of the traditional regionlet approach. We further design a deep regionlet learning module to learn the regionlets through a spatial transformation and a gating network. By using the proposed gating network, which is a soft regionlet selector, the final feature representation is more suitable for detection. The entire pipeline is end-to-end trainable using only the input images and ground truth bounding boxes as supervision.

We conduct a detailed analysis of our approach to understand its merits and properties. Extensive experiments on two detection benchmark datasets, PASCAL VOC [8] and Microsoft COCO [27] show that the proposed deep regionlet approach outperforms several competitors [36, 5, 47, 6, 32]. Even without segmentation labels and feature pyramid, we outperform state-of-the-art algorithms Mask R-CNN [16] and RetinaNet [26]. To summarize, we make the following contributions:

- We propose a novel deep regionlet approach for object detection. Our work extends the traditional regionlet method to the deep learning framework. The system could be trained in a fully end-to-end manner.
- We design a region selection network (RSN), which ***first*** performs ***non-rectangular*** regions selection within the detection bounding box generated from a detection window proposal. It provides more flexibility in modeling objects with variable shapes and deformable parts.
- We propose a deep regionlet learning module, including feature transformation and a gating network. The gating network serves as a soft regionlet selector and let the network focus on features that benefit detection performance.
- We present empirical results on object detection benchmark datasets, which demonstrates the superior performance over state-of-the-art.

[1]The detection window proposal is generated by a region proposal network (RPN) [36, 5, 15]. It is also called region of interest (ROI)

2. Related Work

Object detection has gained a lot of popularity over decades. Many approaches have been proposed including both traditional ones [10, 41, 40] and deep learning-based approaches [15, 36, 28, 34, 5, 14, 17, 6, 32, 11, 47, 19, 4, 46, 44, 12]. Traditional approaches mainly used hand-crafted features (*i.e.* HOG [7], LBP [1]) to train the object detectors using sliding window paradigm. One of the earliest works [40] used boosted cascaded detectors for face detection, which led to its wide adoption. Deformable Part Model based detection (DPM) [9] further extended the cascaded detectors to more general object categories. It proposed the concept of deformable part models to handle object deformations. Due to the rapid development of deep learning techniques [23, 18, 39], the deep learning-based detectors have become dominant object detectors.

Deep learning-based detectors could be further categorized into two classes, single-stage detectors and two-stage detectors, based on whether the detectors have proposal-driven mechanism or not. The single-stage detectors [37, 34, 28, 11, 25, 26] apply regular, dense sampling windows over object locations, scales and aspect ratios. By exploiting multiple layers within a deep CNN network directly, the single-stage detectors achieved high speed but their accuracy was low compared to two-stage detectors.

Two-stage detectors [15, 36, 5, 6, 32, 16] involve two steps. It first generates a sparse set of candidate proposals of detection bounding boxes by the region proposal network (RPN). After filtering out the majority of negative background boxes by RPN, the second stage classifies the proposals of detection bounding boxes and performs the bounding box regression to predict the object categories and their corresponding locations. The two-stage detectors consistently achieve higher accuracy than single-stage detectors and numerous extensions have been proposed [15, 36, 5, 6, 32, 16]. Our method follows the two-stage detector architecture by taking advantage of the region proposal network without the need of dense sampling of object locations, scales and aspect ratios.

3. Our Approach

In this section, We first review the traditional regionlet-based detection methods and then present the overall design of the proposed deep regionlet approach with end-to-end training. Finally, we discuss in detail each module in the proposed end-to-end deep regionlet approach.

3.1. Traditional Regionlet-based Approach

A *regionlet* is a base feature extraction region defined proportionally to a window (*i.e.* a sliding window or a detection bounding box) at arbitrary resolution (*i.e.* size and aspect ratio). Wang *et al.* [41] first introduced the concept of regionlet, as illustrated in Figure 2. It defines a three-level structure among a detecting bounding box, number of regions inside the bounding box and a group of regionlets (sub-regions inside each region). In Figure 2, the yellow box is a detection bounding box. R is a *rectangular* feature extraction region inside the bounding box. Furthermore, small sub-regions $r_{i\{i=1...N\}}$(*e.g.* r_1, r_2) are chosen within region R, where we define them as a set of *regionlets*.

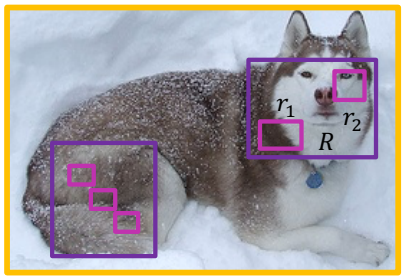

Figure 2: Illustration of structural relationships among the detection bounding box, feature extraction regions and regionlets. The yellow box is a detection bounding box and R is a feature extraction region shown as a purple rectangle with filled dots inside the bounding box. Inside R, two small sub-regions denoted as r_1 and r_2 are the *regionlets*.

The difficulty of the arbitrary detection bounding box has been well addressed by using the *relative* positions and sizes of regionlets and regions. However, in the traditional approach, the initialization of regionlets possess randomness and both regions (R) and regionlets (*i.e.* r_1, r_2) are fixed after the training. Moreover, it is based on hand-crafted feature (*i.e.* HOG [7] or LBP descriptors [1]) in each regionlet respectively and hence not end-to-end trainable. To this end, we propose the following deep regionlet-based approach to address such limitations.

3.2. System Architecture

Generally speaking, an object detection network performs a sequence of convolutional operations on an image of interest using a deep convolutional neural network. At some layer, the network bifurcates into two branches. One branch, RPN generates a set of candidate bounding boxes[2] while the other branch performs classification and regression by pooling the convolutional features inside the proposed bounding box generated by the region proposal network [36, 5]. Taking advantage of this detection network, we introduce the overall design of the proposed object detection framework, named "Deep Regionlets", as illustrated in Figure 1.

The general architecture consists of a region selection network (RSN) and a deep regionlet learning module. In particular, our region selection network is used to predict the

[2] [36, 5, 15] also called the detection bounding box as detection window proposal

transformation parameters to select regions given a candidate bounding box, which is generated by the region proposal network. The regionlets are further learned within each selected region defined by the region selection network. The system is designed to be trained in a fully end-to-end manner using only the input images and ground truth bounding box. The region selection network as well as the regionlet learning module can be simultaneously learned over each selected region given the detection window proposal.

3.3. Region Selection Network

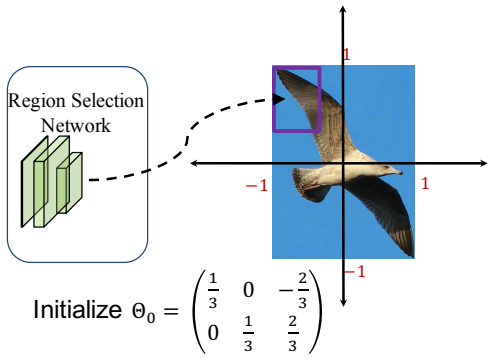

Figure 3: Example of initialization of one affine transformation parameter. Normalized affine transformation parameters $\Theta_0 = [\frac{1}{3}, 0, -\frac{2}{3}; 0, \frac{1}{3}, \frac{2}{3}]$ ($\theta_i \in [-1, 1]$) selects the top-left region in the 3×3 evenly divided detection bounding box, shown as the purple rectangle.

We design the region selection network (RSN) to have the following properties: 1) End-to-end trainable; 2) Simple structure; 3) Generate regions with arbitrary shape. Keeping these in mind, we design the RSN to predict a set of *affine* transformation parameters. By using these affine transformation parameters, as well as not requiring the regions to be rectangular, we have more flexibility in modeling object with arbitrary shape and deformable parts.

Specifically, we design the RSN using a small neural network with three fully connected layers. The first two fully connected layers have output size of 256, with ReLU activation. The last fully connected layer has the output size of six, which is used to predict the set of affine transformation parameters $\Theta = [\theta_1, \theta_2, \theta_3; \theta_4, \theta_5, \theta_6]$.

Note that the candidate detection bounding boxes proposed by RSN have arbitrary sizes and aspect ratios. In order to address this difficulty, we use *relative* positions and sizes of the selected region within a detection bounding box. The candidate bounding box generated by the region proposal network is defined by the top-left point (w_0, h_0), width w and height h of the box. We normalize the coordinates by the width w and height h of the box. As a result, we could use the normalized affine transformation parameters $\Theta = [\theta_1, \theta_2, \theta_3; \theta_4, \theta_5, \theta_6]$ ($\theta_i \in [-1, 1]$) to evaluate one selected region within one candidate detection window at different sizes and aspect ratios without scaling images into multiple resolutions or using multiple-components to enumerate possible aspect ratios, like anchors [36, 28, 11].

Initialization of Region Selection Network: Taking advantage of the *relative* and *normalized* coordinates, we initialize the RSN by equally dividing the whole detecting bounding box to several sub-regions, named as *cell*s, without any overlap among them. Figure 3 shows an example of initialization from one affine transformation (*i.e.* 3×3). The first cell, which is the top-left bin in the whole region (detection bounding box) could be defined by initializing the corresponding affine transformation parameter $\Theta_0 = [\frac{1}{3}, 0, -\frac{2}{3}; 0, \frac{1}{3}, \frac{2}{3}]$. The other eight of 3×3 cells are initialized in a similar way.

3.4. Deep Regionlet Learning

After regions are selected by the region selection network, regionlets are further learned from the selected region defined by the normalized affine transformation parameters. Note that our motivation is to design the network to be trained in a fully end-to-end manner using only the input images and the ground truth bounding boxes. Therefore, both the selected regions and regionlet learning should be able to be trained by CNN networks. Moreover, we would like the regionlets extracted from the selected regions to better represent objects with variable shapes and deformable parts.

Inspired by the spatial transform network [21], any parameterizable transformation including translation, scaling, rotation, affine or even projective transformation can be learned by spatial transformer. In this section, we introduce our deep regionlet learning module to learn the regionlets in the selected region, which is defined by the affine transformation parameters.

More specifically, we aim to learn regionlets from one selected region defined by one affine transformation Θ to better match the shapes of objects. This is done with a selected region R from region selection network, transformation parameters $\Theta = [\theta_1, \theta_2, \theta_3; \theta_4, \theta_5, \theta_6]$ and a set of feature maps $Z = \{Z_i, i = 1, \ldots, n\}$. Without loss of generality, let Z_i be one of the feature map out of the n feature maps. A selected region R is of size $w \times h$ with the top-left corner (w_0, h_0). Inside the Z_i feature maps, we propose the regionlet learning module discussed below:

Let (x_p^s, y_p^s) define the spatial location in feature map Z_i. U_{nm}^c is the value at location (n, m) in channel c of the input feature. The total output feature map V is of size $H \times W$. Let $V(x_p^t, y_p^t, c|\Theta, R)$ be the output feature value at location (x_p^t, y_p^t) ($x_p^t \in [0, H]$, $y_p^t \in [0, W]$) in channel c, which is computed as

$$V(x_p^s, y_p^s, c|\Theta, R) = \sum_n^H \sum_m^M U_{nm}^c \max(0, 1 - |x_p^s - m|) \max(0, 1 - |y_p^s - n|) \quad (1)$$

Back Propagation through Spatial Transform To allow back propagation of the loss through the regionlet learning module, we can define the gradients with respect to both the feature maps and the region selection network. In this layer's `backward` function, we have partial derivative of the loss function with respect to both feature map variable U_{mn}^c and affine transform parameter $\Theta = [\theta_1, \theta_2, \theta_3; \theta_4, \theta_5, \theta_6]$. Motivated by [21], the partial derivative of the loss function with respect to the feature map is:

$$\frac{\partial V(x_p^s, y_p^s, c|\Theta, R)}{\partial U_{nm}^c} = \sum_n^H \sum_m^M \max(0, 1 - |x_p^s - m|) \times \max(0, 1 - |y_p^s - n|) \quad (2)$$

Moreover, during the back propagation, we need to compute the gradient with respect to each affine transformation parameter $\Theta = [\theta_1, \theta_2, \theta_3; \theta_4, \theta_5, \theta_6]$. In this way, the region selection network could also be updated to adjust the selected region. We take θ_1 as an example due to space limitations and similar derivative can be computed for other parameters $\theta_i (i = 2, \ldots, 6)$ respectively.

$$\frac{\partial V(x_p^s, y_p^s, c|\Theta, R)}{\partial \theta_1} = \frac{\partial V(x_p^s, y_p^s, c|\Theta, R)}{\partial x_p^s} \frac{\partial x_p^s}{\partial \theta_1}$$
$$= x_p^t \sum_n^H \sum_m^M U_{nm}^c \max(0, 1 - |y_p^s - n|) \times \begin{cases} 0 \text{ if } |m - x_p^s| \geq 1 \\ 1 \text{ if } m \geq x_p^s \\ -1 \text{ if } m \leq x_p^s \end{cases} \quad (3)$$

It is worth noting that (x_p^t, y_p^t) are normalized coordinates in range $[-1, 1]$ so that it can to be scaled with respect to w and h with start position (w_0, h_0).

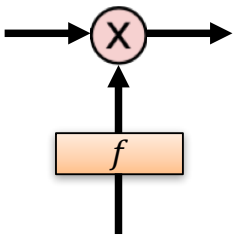

Figure 4: Design of the gating network. f denotes the non-negative gate function (*i.e.* sigmoid)

Gating Network: The gating network, which serves as a soft regionlet selector, is used to assgin regionlets with different weight and generate regionlet feature representation. We design a simple gating network using fully connected layer with `sigmoid` activation. The output values of the gating network are within range of $[0, 1]$. Given the output feature maps $V(x_p^s, y_p^s, c|\Theta, R)$ described above, we use a fully connected layer to generate the same number of output as feature maps $V(x_p^s, y_p^s, c|\Theta, R)$, which is followed by an activation layer `sigmoid` to generate the corresponding weight respectively. The final feature representation is generated by the product of feature maps $V(x_p^s, y_p^s, c|\Theta, R)$ and their corresponding weights.

Regionlet Pool Construction Object deformations may occur at different scales. For instance, deformation could be caused by different body parts in person detection. Same number of regionlets (size $H \times W$) learned from small selected region have higher extraction density, which may lead to non-compact regionlet representation. In order to learn a *compact*, *efficient* regionlet representation, we further perform the pooling (*i.e.* max/ave) operation over the feature maps $V(x_p^s, y_p^s, c|\Theta, R)$ of size $(H \times W)$. We reap two benefits from the pool construction: (1) Regionlet representation is compact (small size). (2) Regionlets learned from different size of selected regions are able to represent such regions in the same efficient way, thus to handle the object deformations at different scales.

3.5. Links to Recent Works

Our deep regionlet approach is related to some recent works in different aspects. In this section, we discuss both similarities and differences in detail.

Spatial Transform Networks (STN) Jaderberg *et al*. [21] first proposed the spatial transformer module to provide spatial transformation capabilities into a deep neural network. It only learns *one global parametric transformation* (scaling, rotations as well as affine transformation). Such learning is known to be difficult to apply on semi-dense vision tasks (*e.g.*, object detection) and the transformation is on the entire feature map, which means the transformation is applied identically across all the regions in the feature map.

Our region selection network learns a set of affine transformation and each transformation can be considered as the localization network in [21]. However, our regionlet learning is different from image sampling [21] method as it adopts a region-based parameter transformation and feature wrapping. By learning the transformation locally in the detection bounding box, our method provide the flexibility of learning a compact, efficient feature representation of objects with variable shape and deformable parts.

Deformable Part Model (DPM) [9] and **its deep learning extensions** [32, 6]. Deformable Part Model (DPM) [9] explicitly models spatial deformations of object parts via latent variables. A root filter is learned to model the global appearance of the objects, while the part filters are designed to describe the local parts in the objects. However, DPM is

a shallow model and the training process involves heuristic choices to select components and part sizes, making end-to-end training inefficient.

Both works [6, 32] extend the DPM with end-to-end training in deep CNNs. Motivated by DPM [10] to allow parts to slightly move around their reference position (partition of the initial regions), they share the similar idea of learning part offsets[3] to model the local element and pool the features at their corresponding locations after the shift. While [6, 32] show promising improvement over other deep learning-based object detectors [15, 36], it still lacks the flexibility of modeling non-rectangular objects with sharp shapes and deformable parts.

It is noticeable that the regionlet learning on the selected region is a generalization of [6, 32]. First, we generalize the selected region to be non-rectangular by learning the affine transformation parameters. Such non-rectangular regions could provide the capabilities of *scaling*, *shifting* and *rotation* around the original reference region. If we only enforce the region selection network to learn the shift, our regionlet learning mechanism would degenerate to similar deformable RoI pooling as in [6, 32]

Spatial-based RoI pooling [24, 22, 17]. Traditional spatial pyramid pooling [24] performs pooling over hand crafted regions at different scales. With the help of deep CNNs, [17] proposes to use spatial pyramid pooling in deep learning-based object detection. However, as the pooling regions over image pyramid still need to be carefully designed to learn the spatial layout of the pooling regions, therefore the end-to-end training is not well facilitated. In contrast, Our deep regionlet learning approach learns pooling regions end-to-end in deep CNNs. Moreover, the region selection step for learning regionlets accommodates different sizes of the regions. Hence, we are able to handle object deformations at different scales without generating the feature pyramid.

4. Experiments

In this section, we present comprehensive experimental results of the proposed approach on two challenging benchmark datasets: PASCAL VOC [8] and MS-COCO [27]. There are in total 20 categories of objects in PASCAL VOC [8] dataset, which includes rigid objects such as cars and deformable objects like cats. We follow the common settings used in [36, 3, 5, 15] to draw compelete comparsions. More specifically, we train our deep model on (1) VOC2007 `trainval` and (2) union of VOC2007 `trainval` and VOC2012 `trainval` and evaluate on VOC2007 `test` set. We also report results on VOC2012 `test` set with the model trained on the VOC2007 `trainvaltest` and VOC2012 `trainval`. In addition, we report the results on the VOC2007 `test` split for ablation study.

[3] [6] uses term offset while [32] uses term displacement

MS-COCO [27] is a widely used challenging dataset, which contains 80 object categories. Following the official settings in COCO website[4], we use the COCO 2017 `trainval` split (union of 135k images from `train` split and 5k images from `val` split) for training. We report the COCO-style average precision (AP) on `test-dev` 2017 split, which requires evaluation from the MS-COCO server[5] for testing.

For the base network, We choose both VGG-16 [39] and ResNet-101 [18] to demonstrate the generalization of our approach regardless of which network backbone we use. The *á trous* algorithm [29, 31] is adopted in stage 5 of ResNet-101. Following the suggested settings in [5, 6], we also set the pooling size to 7 by changing the conv5 stage's effective stride from 32 to 16 to increase the feature map resolution. In addition, the first convolution layer with stride 2 in the conv5 stage is modified to 1. Both backbone networks are intialized with the pre-trained ImageNet [18, 23] model.

In the following sections, we report the results of a series of ablation experiments to understand the behavior of the proposed deep regionlet approach. Furthermore, we present comparisons with state-of-the-art detectors [36, 5, 6, 16, 26, 25] on both PASCAL VOC [8] and MS COCO [27] datasets.

4.1. Ablation Study

For a fair comparison, we adopt ResNet-101 as the backbone network for ablation studies. We train our model on the union set of VOC $2007 + 2012$ `trainval` and evaluate on the VOC2007 `test` set. The shorter side of image is set to be 600 pixels, as suggested in [15, 36, 5]. The training is performed for 60k iterations with effective mini-batch size 4 on 4 GPUs, where the learning rate is set as 10^{-3} for the first 40k iterations and 10^{-4} for the rest 20k iterations. First we investigate the proposed approach to understand each component (1) Region selection network, (2) Deep regionlet learning and (3) Soft regionlet selection by comparing it with several baselines:

1. Global region selection network (RSN). RSN only selects one global region and it is initialized as identity transformation (*i.e.* $\Theta_0 = [1, 0, 0; 0, 1, 0]$). This is equivalent to global regionlet learning within the RoI.
2. Offset-only RSN. We set the region selection network to only learn the offset by enforcing $\theta_1, \theta_2, \theta_4, \theta_5$ not to change during the training process. In this way, the region selection network only selects the rectangular region with offsets to the initialized region. This baseline is similar to the Deformable RoI Pooling in [6] and [32].
3. Non-gating selection: deep regionlet without soft selection. No soft regionlet selection is performed after the

[4] http://cocodataset.org/#detections-challenge2017

[5] The updated settings (2017) are different from the previous settings (2016, 2015) in [3, 26, 6, 5, 26], as it includes different train/val sets.

Methods	Global RSN	Offset-only RSN [6, 32]	Non-gating	Ours
mAP@0.5(%)	30.27	78.5	81.3 (+2.8)	82.0 (+3.5)

Table 1: Ablation study of each component in deep regionlet approach. Output size $H \times W$ is set to 4×4 for all the baselines

# of Regions \ Regionlets Density	2×2	3×3	4×4	5×5	6×6
$4(2 \times 2)$ regions	78.0	79.2	79.9	80.2	80.3
$9(3 \times 3)$ regions	79.6	80.3	80.9	81.5	81.3
$16(4 \times 4)$ regions	80.0	81.0	82.0	81.6	80.8

Table 2: Results of ablation studies when a region selection network (RSN) selects different number of regions and regionlets are learned at different level of density.

regionlet learning. In this case, each regionlet learned has the same contribution to the final feature representation.

Results are shown in Table 1. First, when the region selection network only selects one global region, the region selection network reduces to the single localization network [21]. In this case, regionlets will be extracted in a global manner. It is interesting to note that selecting only one region by the region selection network is able to converge, which is different from [36, 5]. However, the performance is extremely poor. This is because no discriminative regionlets could be explicitly learned within the region. More importantly, compared our approach and offset-only RSN with global RSN, the results clearly demonstrate that the region selection network (RSN) is ***indispensable*** in the deep regionlet approach.

Moreover, offset-only RSN could be viewed as similar to deformable RoI pooling in [6, 32]. These methods all learn the offset of the rectangle region with respect to its reference position, which lead to improvement over [36]. However, non-gating selection outperforms offset-only RSN by 2.8% with selecting non-rectangular region. The improvement demonstrates that non-rectangular region selection could provide more flexibility around the original reference region, thus could better model the non-rectangular objects with sharp shapes and deformable parts. Last but not least, by using the gate function to perform soft regionlet selection, the performance can be further improved by 0.7%.

Next, we present ablation studies on the following questions in order to understand more deeply on the region selection network and regionlet learning module:

1. How many regions should we learn by region selection network?
2. How many regionlets should we learn in one selected region (density is of size $H \times W$)?

How many regions should we learn by region selection network? We investigate how the detection performance varies when different number of regions are selected by the region selection network. All the regions are initialized as described in Section 3.3 without any overlap between regions. Without loss of generality, we report results for $4(2 \times 2)$, $9(3 \times 3)$ and $16(4 \times 4)$ regions in Table 2. We observe that the mean AP increases when the number of selected regions is increased from $4(2 \times 2)$ to $9(3 \times 3)$ for fixed regionlets learning number, but gets saturated with $16(4 \times 4)$ selected regions.

How many regionlets should we learn in one selected region? Next, we investigate how the detection performance varies when different number of regionlets are learned in one selected region by varying H and W. Without loss of generality, we set $H = W$ throughout our experiments and vary the H value from 2 to 6. In Table 2, we report results when we set the number of regionlets at $4(2 \times 2)$, $9(3 \times 3)$, $16(4 \times 4)$, $25(5 \times 5)$, $36(6 \times 6)$ before the regionlet pooling construction.

First, it is observed that increasing the number of regionlets from $4(2 \times 2)$ to $25(5 \times 5)$ results in improved performance. As more regionlets are learned from one region, more spatial and shape information from objects could be learned. The proposed approach could achieve the best performance when regionlets are extracted at $16(4 \times 4)$ or $25(5 \times 5)$ density level. It is also interesting to note that when the density increases from $25(5 \times 5)$ to $36(6 \times 6)$, the performance degrades slightly. When the regionlets are learned at a very high density level, some redundant spatial information may be learned without being useful for detection, thus affecting the region proposal-based decision to be made. Throughout all the experiments in the paper, we report the results from 16 selected regions from region selection network and set output size $H \times W = 4 \times 4$.

4.2. Experiments on PASCAL VOC

In this section, we compare our results with traditional regionlet method [41] and several state-of-the-art deep learning-based object detectors as follows: Faster R-CNN [36], SSD [28], R-FCN [5], soft-NMS [3], DP-

Methods	training data	mAP@0.5(%)	training data	mAP@0.5(%)
Regionlet [41]	07	41.7	07 + 12	N/A
Faster R-CNN [36]	07	70.0	07 + 12	73.2
R-FCN [5]	07	69.6	07 + 12	76.6
SSD 512 [28]	07	71.6	07 + 12	76.8
Soft-NMS [3]	07	71.1	07 + 12	76.8
Ours	07	***73.0***	07 + 12	***79.2***
Ours with soft-NMS	07	**73.8**	07 + 12	**80.1**

Table 3: Detection results on PASCAL VOC using VGG16 as backbone architecture. Training data: "07": VOC2007 `trainval`, "07 + 12": union set of VOC2007 and VOC2012 `trainval`. With soft-NMS denotes we apply the soft-NMS in the test stage.

FCN [32] and DCN [6].

We follow the standard settings as in [36, 5, 3, 6] and report mean average precision (mAP) scores using IoU thresholds at 0.5 and 0.7. For the first experiment training from VOC2007 `trainval`, we start learning rate at 10^{-3} for the first 40k iterations, then we decrease it to 10^{-4} for the rest 20k iterations with single GPU. Next, due to more training data, increasing the number of iteration is needed on the union of VOC2007 and VOC2012 `trainval`. We perform the same training process as described in Section 4.1. Moreover, we use 300 RoIs at test stage from a single-scale image testing with setting the image shorter side to be 600. For a fair comparison, we do not deploy the multi-scale training/testing or online hard example mining(OHEM) [38], although it is shown in [3, 6] that such enhancements could lead to the performance boost.

The results on VOC2007 `test` using VGG16 [39] backbone are shown in Table 3. We first compare with traditional regionlet method [41] and several state-of-the-art object detectors [36, 28, 3] when training from small size dataset (VOC2007 `trainval`). Next, we evaluate our method as we increase the training dataset (union set of VOC 2007 and 2012 `trainval`). With the power of deep CNNs, the deep regionlet approach has significantly improved the detection performance over the traditional regionlet method [41]. We also observe that more data always helps. Moreover, it is encouraging that soft-NMS [3] is only applied in the test stage without modification in the training stage, which could directly improve over [36] by 1.1%. In summary, our method consistently outperform all the compared methods and the performance could be further improved if we replace NMS with its better variant soft-NMS [3].

Next, we change the network backbone from VGG16 [39] to ResNet-101 [18] and present corresponding results in Table 4. In addition, we also compare with DCN [6] and DP-FCN [32].

First, compared to the performance in Table 3 using VGG16 [39] network, the mAP can be significantly increased by using deeper networks like ResNet-101 [18]. Second, comparing with DP-FCN [32] and Deformable ROI Pooling in DCN [6], we outperform these two methods by **3.9**% and **2.7**% respectively. This provides the empirical support that our deep regionlet learning method could be treated as a *generalization* of [6, 32], as discussed in Section 3.5. In addition, the results demonstrate that selecting *non-rectangular* regions from our method provide more capabilities including *scaling*, *shifting* and *rotation* to learn the feature representations. In summary, without bells and whistles, our method achieves state-of-the-art performance on object detection task when using ResNet-101 as backbone network.

Methods	mAP@0.5 / @0.7(%)
Faster R-CNN [36]	78.1 / 62.1
SSD [28]	76.8 / N/A
DP-FCN [32]	78.1 / N/A
ION [2]	79.4 / N/A
LocNet [13]	78.4 / N/A
Deformable ConvNet [6]	78.6 / 63.3
Deformale ROI Pooling [6]	78.3 / 66.6
Deformable ConvNet + ROI Pooling [6]	79.3 / 66.9
Ours	***82.0 / 67.0***
Ours with soft-NMS	**83.1 / 67.9**

Table 4: Detection results on PASCAL VOC using ResNet-101 [18] as backbone acchitecture. Training data: union set of VOC 2007 and 2012 `trainval`. With soft-NMS denotes we apply the soft-NMS in the test stage.

Results evaluated on VOC2012 `test` are shown in Table 5. We follow the same experimental settings as in [5, 36, 11, 28, 32] and train our model using VOC"07++12", which consists of VOC2007 `trainvaltest` and VOC2012 `trainval` set. It can be seen that our method outperform all the other competing methods. In particular, we outperform DP-FCN [32], which further proves the generalization of our method over [32].

4.3. Experiments on MS COCO

In this section, we evaluate the proposed deep regionlet approach on MS COCO [27] dataset and compare with other state-of-the-art object detectors: Faster R-CNN [36],

Methods	FRCN [36]	YOLO9000 [35]	FRCN OHEM	DSSD [11]	SSD* [28]
mAP@0.5(%)	73.8	73.4	76.3	76.3	78.5
Methods	ION [2]	R-FCN [5]	DP-FCN [32]	Ours	Ours with soft-NMS
mAP@0.5(%)	76.4	77.6	79.5	***80.4***	**81.2**

Table 5: Detection results on VOC2012 `test` set using training data "07++12": the union set of 2007 *trainvaltest* and 2012 *trainval*. SSD* denotes the new data augmentation

Methods	Training Data	mmAP 0.5:0.95	mAP @0.5	mAP small	mAP medium	mAP large
Faster R-CNN [36]	trainval	24.4	45.7	7.9	26.6	37.2
SSD*[28]	trainval	31.2	50.4	10.2	34.5	49.8
DSSD [11]	trainval	33.2	53.5	13.0	35.4	51.1
R-FCN [5]	trainval	30.8	52.6	11.8	33.9	44.8
Deformable F-RCNN [6]	trainval	33.1	50.3	11.6	34.9	51.2
Deformable R-FCN [6]	trainval	34.5	55.0	14.0	37.7	50.3
Mask R-CNN [16]	trainval	37.3	59.6	19.8	40.2	48.8
RetinaNet500 [26]	trainval	34.4	53.1	14.7	38.5	49.1
Ours	trainval	**39.3**	**59.8**	**21.7**	**43.7**	50.9

Table 6: Object detection results on MS COCO 2017 `test-dev` using ResNet-101 [18] as backbone acchitecture. Training data: union set of 2017 `train` and 2017 `val` set. SSD* denotes the new data augmentation

SSD [28], R-FCN [5], Deformable F-RCNN/R-FCN [6], Mask R-CNN [16], RetinaNet [26].

We adopt ResNet-101 [18] as the backbone architecture of all the methods for a fair comparison. Following the settings in [16, 6, 26, 5], we set the shorter edge of the image to 800 pixels. The training is performed for 280k iterations with effective mini-batch size 8 on 8 GPUs. We first train the model with 10^{-3} learning rate for the first 160k iterations, followed by learning rate 10^{-4} and 10^{-5} for another 80k iterations and the rest 40k iterations. 5 scales and 3 aspect ratios are deployed for anchors. We report results using either the released models or the code from the original authors. It is noted that we only deploy single-scale image training (no scale jitter) without the iterative bounding box average throughout all the experiments, although these enhancements could further boost performance (mmAP).

Table 6 shows the results on 2017 `test-dev` set[6], which contains $20,288$ images. Compared with the baseline methods Faster R-CNN [36], R-FCN [5] and SSD [28], both Deformable F-RCNN/R-FCN [6] and our method provide huge improvements over [36, 5, 28] (+3.7% and **+8.5**%). Moreover, it can be seen that our method outperform Deformable F-RCNN/R-FCN [6] by wide margin(∼**4**%). This observation further supports that our deep regionlet learning module could be treated as a *generalization* of [6, 32], as discussed in Section 3.5. It is also noted that the most recent state-of-the-art object detectors Mask R-CNN[7] [16] also utilized multi-task training with segmentation labels and in-network feature pyramid (FPN), we still outperform [16] by **2.0**%. In addition, the main contribution focal loss in [26], which overcomes the obstacle caused by the imbalance of positive/nagetive samples, is complimentary to our method. We believe it can be applied in our method to further boost the performance. In summary, compared with Mask R-CNN [16] and RetinaNet[8] [26], our method achieves state-of-the-art performance on MS COCO when using ResNet-101 as a backbone network.

5. Conclusion

In this paper, we present a novel deep regionlet-based approach for object detection. The proposed region selection network can select *non-rectangular* region within the detection bounding box, and hence an object with rigid shape and deformable parts can be better modeled. We also design the deep regionlet learning module so that both the selected regions and the regionlets can be learned simultaneously. Moreover, the proposed system can be trained in a fully end-to-end manner without additional efforts. Finally, we extensively evaluate our approach on two detection benchmarks for generic object detection. Experimental results shows competitive performance over state-of-the-art.

6. Acknowledgement

This research is based upon work supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC)

[6] MS COCO server does not accept 2016 and 2015 test-dev evaluation. As a result, we are not able to report results on 2016, 2015 test-dev set.

[7] Note [16] reported best result using ResNeXt-101-FPN [43], we only compare the results in [16] using ResNet-101 [18] backbone for fair comparison.

[8] [26] reported best result using multi-scale training for 1.5× longer iterations, we only compare the results without scale jitter during training.

contract number D17PC00345. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation theron. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of IARPA, DOI/IBC or the U.S. Government.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 469–481, 2004. 1, 3

[2] S. Bell, C. L. Zitnick, K. Bala, and R. B. Girshick. Inside-Outside Net: Detecting objects in context with skip pooling and recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2883, 2016. 8, 9

[3] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS - improving object detection with one line of code. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5562–5570, 2017. 1, 6, 7, 8

[4] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. *CoRR*, abs/1712.00726, 2017. 3

[5] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 379–387, 2016. 1, 2, 3, 6, 7, 8, 9

[6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 1, 2, 3, 5, 6, 7, 8, 9

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 1, 3

[8] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 6

[9] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester. Cascade object detection with deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248, 2010. 3, 5

[10] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 3, 6

[11] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017. 3, 4, 8, 9

[12] M. Gao, R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Dynamic zoom-in network for fast object detection in large images. *CoRR*, abs/1711.05187, 2017. 1, 3

[13] S. Gidaris and N. Komodakis. LocNet: Improving localization accuracy for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 789–798, 2016. 8

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 3

[15] R. B. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 1, 2, 3, 6

[16] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 2, 3, 6, 9

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision (ECCV)*, pages 346–361, 2014. 3, 6

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 3, 6, 8, 9

[19] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. *CoRR*, abs/1711.11575, 2017. 3

[20] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3296–3297, 2017. 1

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 4, 5, 7

[22] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3370–3377, 2012. 6

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105. 2012. 1, 3, 6

[24] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006. 6

[25] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 3, 6

[26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 1, 2, 3, 6, 9

[27] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 2, 6, 8

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37, 2016. 1, 3, 4, 7, 8, 9

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1, 6

[30] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999. 1

[31] S. Mallat. *A Wavelet Tour of Signal Processing, 2nd Edition*. Academic Press, 1999. 6

[32] T. Mordan, N. Thome, M. Cord, and G. Henaff. Deformable part-based fully convolutional network for object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 1, 2, 3, 5, 6, 7, 8, 9

[33] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, K. Wang, J. Yan, C. C. Loy, and X. Tang. DeepID-Net: Object detection with deformable part based convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1320–1334, 2017. 1

[34] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 3

[35] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. 9

[36] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 1, 2, 3, 4, 6, 7, 8, 9

[37] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2014. 3

[38] A. Shrivastava, A. Gupta, and R. B. Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016. 8

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 6, 8

[40] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001. 1, 3

[41] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 17–24, 2013. 1, 2, 3, 7, 8

[42] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2071–2084, 2015. 1

[43] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017. 9

[44] R. Yu, X. Chen, V. I. Morariu, and L. S. Davis. The role of context selection in object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. 1, 3

[45] R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1068–1076, 2017. 1

[46] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. *CoRR*, abs/1711.06897, 2017. 3

[47] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille. Single-shot object detection with enriched semantics. *CoRR*, abs/1712.00433, 2017. 2, 3