Adaptive Object Detection Using Adjacency and Zoom Prediction

Yongxi Lu
University of California, San Diego
yol070@ucsd.edu

Tara Javidi
University of California, San Diego
tjavidi@ucsd.edu

Svetlana Lazebnik
University of Illinois at Urbana-Champaign
slazebni@illinois.edu

Abstract

State-of-the-art object detection systems rely on an accurate set of region proposals. Several recent methods use a neural network architecture to hypothesize promising object locations. While these approaches are computationally efficient, they rely on fixed image regions as anchors for predictions. In this paper we propose to use a search strategy that adaptively directs computational resources to sub-regions likely to contain objects. Compared to methods based on fixed anchor locations, our approach naturally adapts to cases where object instances are sparse and small. Our approach is comparable in terms of accuracy to the state-of-the-art Faster R-CNN approach while using two orders of magnitude fewer anchors on average. Code is publicly available.

1. Introduction

Object detection is an important computer vision problem for its intriguing challenges and large variety of applications. Significant recent progress in this area has been achieved by incorporating deep convolutional neural networks (DCNN) [15] into object detection systems [5, 8, 10, 12, 17, 23, 25].

An object detection algorithm with state-of-the-art accuracy typically has the following two-step cascade: a set of class-independent region proposals are hypothesized and are then used as input to a detector that gives each region a class label. The role of region proposals is to reduce the complexity through limiting the number of regions that need be evaluated by the detector. However, with recently introduced techniques that enable sharing of convolutional features [9, 12], traditional region proposal algorithms such as selective search [27] and EdgeBoxes [29] become the bottleneck of the detection pipeline.

An emerging class of efficient region proposal methods are based on end-to-end trained deep neural networks [5, 22]. The common idea in these approaches is to train a class-independent regressor on a small set of pre-defined anchor regions. More specifically, each anchor region is assigned the task of deciding whether an object is in its neighborhood (in terms of center location, scale and aspect ratio), and predicting a bounding box for that object through regression if that is the case. The design of anchors differs for each method. For example, MultiBox [5] uses 800 anchors from clustering, YOLO [21] uses a non-overlapping 7 by 7 grid, RPN [22] uses overlapping sliding windows. In these prior works the test-time anchors are not adaptive to the actual content of the images, thus to further improve accuracy for detecting small object instances a denser grid of anchors is required for all images, resulting in longer test time and a more complex network model.

We alternatively consider the following adaptive search strategy. Instead of fixing *a priori* a set of anchor regions, our algorithm starts with the entire image. It then recursively divides the image into sub-regions (see Figure 2) until it decides that a given region is unlikely to enclose any small objects. The regions that are visited in the process effectively serve as anchors that are assigned the task of predicting bounding boxes for objects nearby. A salient feature of our algorithm is that the decision of whether to divide a region further is based on features extracted from that particular region. As a result, the generation of the set of anchor regions is conditioned on the image content. For an image with only a few small objects most regions are pruned early in the search, leaving a few small anchor regions near the objects. For images that contain exclusively large instances, our approach gracefully falls back to existing methods that rely on a small number of large anchor regions. In this manner, our algorithm adaptively directs its computational resources to regions that are likely to contain objects. Figure 1 compares our algorithm with RPN.

To support our adaptive search algorithm, we train a deep

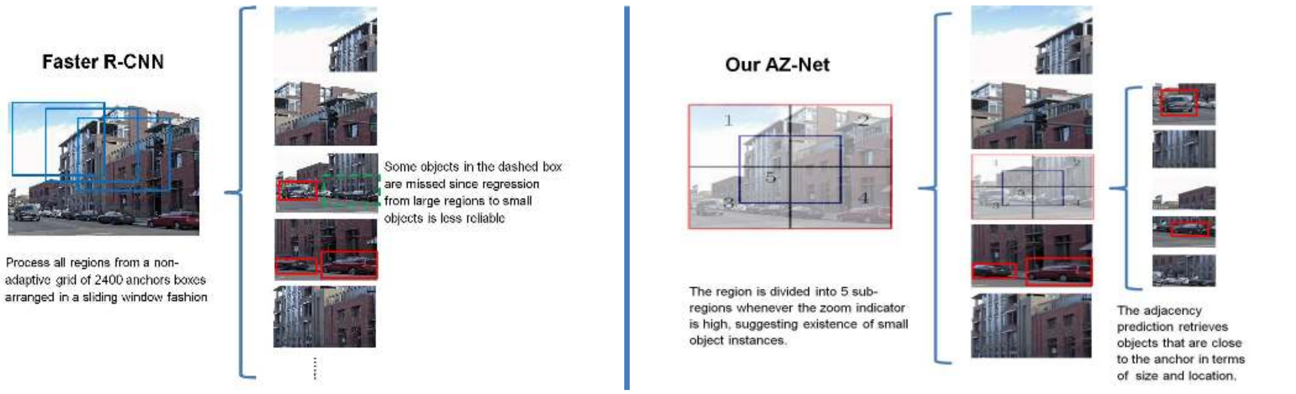
Faster R-CNN
Process all regions from a non-adaptive grid of 2400 anchors boxes arranged in a sliding window fashion
Some objects in the dashed box are missed since regression from large regions to small objects is less reliable
Our AZ-Net
1
2
5
3
4
The region is divided into 5 sub-regions whenever the zoom indicator is high, suggesting existence of small object instances.
The adjacency prediction retrieves objects that are close to the anchor in terms of size and location.

Figure 1. Comparison of our proposed adaptive search algorithm with the non-adaptive RPN method. The red boxes show region proposals from adjacency predictions. Note that for small objects, RPN is forced to perform regression from much larger anchors, while our AZ-Net approach can adaptively use features from small regions.

neural network we call *Adjacency and Zoom Network (AZ-Net)*. Given an input anchor region, the AZ-Net outputs a scalar *zoom indicator* which is used to decide whether to further zoom into (divide) the region and a set of bounding boxes with confidence scores, or *adjacency predictions*. The adjacency predictions with high confidence scores are then used as region proposals for a subsequent object detector. The network is applied recursively starting from the whole image to generate an adaptive set of proposals.

To intuitively motivate the design of our network, consider a situation in which one needs to perform a quick search for a car. A good strategy is to first look for larger structures that could provide evidence for existence of smaller structures in related categories. A search agent could, for example, look for roads and use that to reason about where cars should be. Once the search nears the car, one could use the fact that seeing certain parts is highly predictive of the spatial support of the whole. For instance, the wheels provide strong evidence for a tight box of the car. In our design, the zoom indicator mimics the process of searching for larger structures, while the adjacency predictions mimic the process of neighborhood inference.

To validate this design we extensively evaluate our algorithm on Pascal VOC 2007 [6] with fine-grained analysis. We also report baseline results on the recently introduced MSCOCO [18] dataset. Our algorithm achieves detection mAP that is close to state-of-the-art methods at a fast frame rate. Code has been made publicly available at `https://github.com/luyongxi/az-net`.

In summary, we make the following contributions:

- We design a search strategy for object detection that adaptively focuses computational resources on image regions that contain objects.

- We evaluate our approach on Pascal VOC 2007 and MSCOCO datasets and demonstrate it is comparable to Fast R-CNN and Faster R-CNN with fewer anchor and proposal regions.

- We provide a fine-grained analysis that shows intriguing features of our approach. Namely, our proposal strategy has better recall for higher intersection-over-union thresholds, higher recall for smaller numbers of top proposals, and for smaller object instances.

This paper is organized as follows. In section 2 we survey existing literature highlighting the novelty of our approach. In Section 3 we introduce the design of our algorithm. Section 4 presents an empirical comparison to existing object detection methods on standard evaluation benchmarks, and Section 5 discusses possible future directions.

2. Previous Work

Lampert *et al.* [16] first proposed an adaptive branch-and-bound approach. More recently, Gonzeles-Garcia *et al.* [11], Caicedo and Lazebnik [3], and Yoo *et al.* [28] explored active object detection with DCNN features. While these approaches show the promise of using an adaptive algorithm for object detection, their detectors are class-wise and their methods cannot achieve competitive accuracy. Our approach, on the other hand, is multi-class and is comparable to state-of-the-art approaches in both accuracy and test speed.

The idea of using spatial context has been previously explored in the literature. Previous work by Torralba *et al.* [26] used a biologically inspired visual attention model [2], but our focus is on efficient engineering design. Divvala *et al.* [4] evaluated the use of context for localization, but their empirical study was performed on hand-crafted features and needs to be reexamined in combination with more accurate recent approaches.

Our method is closely related to recent approaches that

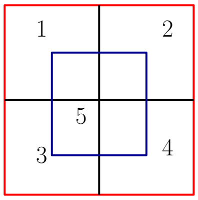

Figure 2. As illustrated, a given region is divided into 5 sub-regions (numbered). Each of these sub-regions is recursively divided if its zoom indicator is above a threshold.

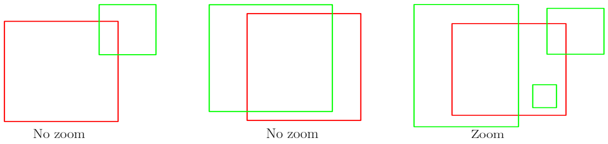

Figure 3. Illustration of desired zoom indicator for common situations. The green boxes are objects, and the red boxes are regions. Left: the object is small but it is mostly outside the region – there is no gain in zooming in. Middle: the object is mostly inside but its size is large relative to the region – there is no gain in zooming in. Right: there is a small object that is completely inside the region. In this case further division of the region greatly increases the chance of detection for that object.

use anchor regions for proposal generation or detection. For example, Erhan *et al*. [5] use 800 data-driven anchors for region proposals and Redmon *et al*. [21] use a fixed grid of 49 non-overlapping regions to provide class-wise detections. The former has the concern that these anchors could overfit the data, while the latter cannot achieve state-of-the-art performance without model ensembles. Our work is most related to the recent work by Ren *et al*. [22], which uses a set of heuristically designed 2400 overlapping anchor regions. Our approach uses a similar regression technique to predict multiple bounding boxes from an anchor region. However, our anchor regions are generated adaptively, making them intrinsically more efficient. In particular, we show that it is possible to detect small object instances in the scene without an excessive number of anchor regions. We propose to grow a tree of finer-grained anchor regions based on local image evidence, and design the regression model strategically on top of it. We extensively compare the output of our method against [22] in our experimental section and show the unique advantages of our approach.

This paper is a follow-up to the work published in the 53rd Annual Allerton Conference [20]. Here, we introduce a substantially improved algorithm and add extensive evaluations on standard benchmarks.

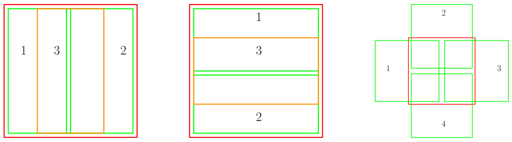

Figure 4. Illustration of sub-region priors. From left to right: vertical stripes, horizontal stripes, neighboring squares. The red rectangular box is the image. In the figure the numbered regions are template sub-regions. The gaps between sub-regions are exaggerated for better visualization. The vertical stripes are used to detect tall objects, the horizontal stripes are used to detect fat objects, while the neighboring squares are used to detect objects that fall in the gaps between anchor regions generated in the search process.

3. Design of the Algorithm

3.1. Overview of the Adaptive Search

Our object detection algorithm consists of two steps. In step 1, a set of class-independent region proposals are generated using Adaptive Search with AZ-Net (see Algorithm 1). In step 2, an object detector evaluates each region proposed in step 1 to provide class-wise detections. In our experiments the detector is Fast R-CNN.

Our focus is on improving step 1. We consider a recursive search strategy, starting from the entire image as the root region. For any region encountered in the search procedure, the algorithm extracts features from this region to compute the zoom indicator and the adjacency predictions. The adjacency predictions with confidence scores above a threshold are included in the set of output region proposals. If the zoom indicator is above a threshold, this indicates that the current region is likely to contain small objects. To detect these embedded small objects, the current region is divided into sub-regions in the manner shown in Figure 2. Each of these sub-regions is then recursively processed in the same manner as its parent region, until either its area or its zoom indicator is too small. Figure 1 illustrates this procedure.

In the following section, we discuss the design of the zoom indicator and adjacency prediction.

3.2. Design of Building Blocks

The zoom indicator should be large for a region only when there exists at least one object whose spatial support mostly lies within the region, and whose size is sufficiently small compared to the region. The reasoning is that we should zoom in to a region only when it substantially increases the chance of detection. For example, if an object is mostly outside the region, dividing the region further is unlikely to increase the chance of detecting that object. Similarly, if an object is large compared to the current region, the task of detecting this object should be handled by this

Algorithm 1: Adaptive search with AZ-Net.

```
Data: Input image $x$ (the whole image region $b_x$). $Y_k$ is the region proposed at step $k$. $Y^k$ are the accumulated region
      proposals up to step $k$. $Z_k$ are the regions to further zoom in to at step $k$. $B_k$ are anchor regions at step $k$.
Result: Region proposals at termination $Y^K$.
Initialization: $B_0 \leftarrow \{b_x\}$. $Y^0 \leftarrow \emptyset$, $k \leftarrow 0$
while ($B_k$ is not an empty set) do
    Initialize $Y_k$ and $Z_k$ as empty sets.
    foreach $b \in B_k$ do
        Compute adjacency predictions $A_b$ and the zoom indicator $z_b$ using AZ-Net.
        Include all $a \in A_b$ with high confidence scores into $Y_k$.
        Include $b$ into $Z_k$ if $z_b$ is above threshold.
    end
    $Y^k \leftarrow Y^{k-1} \cup Y_k$
    $B_{k+1} \leftarrow$ Divide-Regions$(Z_k)$
    $k \leftarrow k + 1$
end
$K \leftarrow k - 1$
```

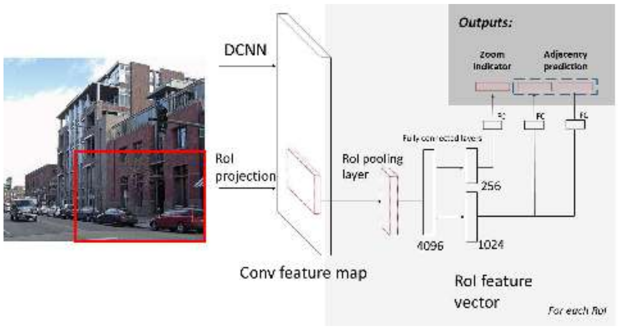

Figure 5. Illustration of the AZ-Net architecture.

region or its parents. In the latter case, further division of the region not only wastes computational resources, but also introduces false positives in the region proposals. Figure 3 shows common situations and the desirable behavior of the zoom indicator.

The role of adjacency prediction is to detect one or multiple objects that overlap with the anchor region sufficiently by providing tight bounding boxes. The adjacency prediction should be aware of the search geometry induced by the zoom indicator. More specifically, the adjacency prediction should perform well on the effective anchor regions induced by the search algorithm. For this purpose we propose a training procedure that is aware of the adaptive search scheme (discussed in Section 3.3). On the other hand, its design should explicitly account for typical geometric configurations of objects that fall inside the region, so that the training can be performed in a consistent fashion. For this reason, we propose to make predictions based on a set of sub-region priors as shown in Figure 4. Note that we also include the anchor region itself as an additional prior. We make sub-region priors large compared to the anchor under the intuition that if an object is small, it is best to wait until the features extracted are at the right scale to make bounding box predictions.

3.3. Implementation

We implement our algorithm using the Caffe [14] framework, utilizing the open source infrastructure provided by the Fast R-CNN repository [9]. In this section we introduce the implementation details of our approach. We use the Fast R-CNN detector since it is a fast and accurate recent approach. Our method should in principle work for a broad class of object detectors that use region proposals.

We train a deep neural network as illustrated in Figure 5. Note that in addition to the sub-region priors as shown in Figure 4, we also add the region itself as a special prior region, making in total 11 adjacency predictions per anchor. For the convolutional layers, we use the VGG16 model [23] pre-trained on ImageNet data. The fully-connected layers are on top of a region pooling layer introduced in [9] which allows efficient sharing of convolutional layer features.

The training is performed as a three-step procedure. First, a set of regions is sampled from the image. These samples should contain hard positive and negative examples for both the zoom indicator and the adjacency prediction. Finally, the tuples of samples and labels are used in standard stochastic gradient descent training. We now discuss how the regions are sampled and labeled, and the loss function we choose.

3.3.1 Region Sampling and Labeling

Since a typical image only has a few object instances, to provide sufficient positive examples for adjacency predic-

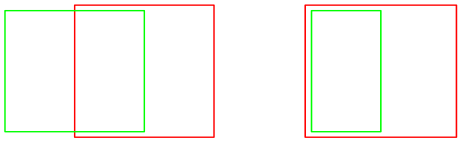

Figure 6. Illustration of the inverse matching procedure. The red box is the inverse match for the object (green box). The left figure shows inverse matching of a neighboring square, the right figure shows inverse matching of a vertical stripe.

tions our method inversely finds regions that will see a ground truth object as a perfect fit to its prior sub-regions (see Figure 6 for illustration). This provides $k \times 11$ training examples for each image, where k is the number of objects.

To mine for negative examples and hard positive examples, we search the input image as in Algorithm 1. Note that the algorithm uses zoom indicators from the AZ-Net. Instead of optimizing AZ-Net with an on-policy approach (that uses the intermediate AZ-Net model to sample regions), which might cause training to diverge, we replace the zoom prediction with the zoom indicator label. However, we note that using the zoom label directly could cause overfitting, since at test time the algorithm might encounter situations where a previous zoom prediction is wrong. To improve the robustness of the model, we add noise to the zoom label by flipping the ground truth with a probability of 0.3. We found that models trained without random flipping are significantly less accurate. For each input image we initiate this procedure with five sub-images and repeat it multiple times. We also append horizontally flipped images to the dataset for data augmentation.

Assignment of labels for the zoom indicator follows the discussion of Section 3. The label is 1 if there exists an object with 50% of its area inside the region and the area is at most 25% of the size of the region. Note that here we use a loose definition of inclusion to add robustness for objects falling between boundaries of anchors. For adjacency prediction, we set a threshold in the intersection-over-union (IoU) score between an object and a region. A region is assigned to detect objects with which it has sufficient overlap. The assigned objects are then greedily matched to one of the sub-regions defined by the priors shown in Figure 4. The priority in the matching is determined by the IoU score between the objects and the sub-regions. We note that in this manner multiple predictions from a region are possible.

3.3.2 Loss Function

As shown in Figure 5, the AZ-Net has three output layers. The zoom indicator outputs from a sigmoid activation function. To train it we use the cross-entropy loss function popular for binary classification. For the adjacency predictions, the bounding boxes are parameterized as in Fast R-CNN [22]. Unlike in Fast R-CNN, to provide multiple predictions from any region, the confidence scores are not normalized to a probability vector. Correspondingly we use smooth L1-loss for bounding box output and element-wise cross-entropy loss for confidence score output. The three losses are summed together to form a multi-task loss function.

3.3.3 Fast R-CNN Detectors

The detectors we use to evaluate proposal regions are Fast R-CNN detectors trained using AZ-Net proposals. As in [22], we implement two versions: one with unshared convolutional features and the other that shares convolutional features with AZ-Net. The shared version is trained using alternating optimization.

4. Experiments

We evaluate our approach on Pascal VOC 2007 [6] and MSCOCO [18] datasets. In addition to evaluating the accuracy of the final detectors, we also perform detailed comparisons between the RPN approach adopted in Faster R-CNN and our AZ-Net on VOC 2007. At the end of the section, we give an analysis of the efficiency of our adaptive search strategy.

4.1. Results on VOC 2007

To set up a baseline comparison, we evaluate our approach using the standard average precision (AP) metric for object detection. For AP evaluation we use the development kit provided by the VOC 2007 object detection challenge. We compare our approach against the recently introduced Fast R-CNN [9] and Faster R-CNN [22] systems, which achieve state-of-the-art performance in standard benchmarks, such as VOC 2007 [6] and VOC 2012 [7]. A comparison is shown in Table 1. The results suggest that our approach is comparable to or better than these methods.

4.2. Quality of Region Proposals

We preform a detailed analysis of the quality of region proposals from our AZ-Net, highlighting a comparison to the RPN network used in Faster R-CNN. For all our experiments, we analyze the recall on Pascal VOC 2007 test set using the following definition: An object is counted as retrieved if there exists a region proposal with an above-threshold IoU with it. The recall is then calculated as the proportion of the retrieved objects among all ground truth object instances. To accurately reproduce the RPN approach, we downloaded the region proposals provided on the Faster R-CNN repository [1]. We used the results from

[1] https://github.com/ShaoqingRen/faster_rcnn

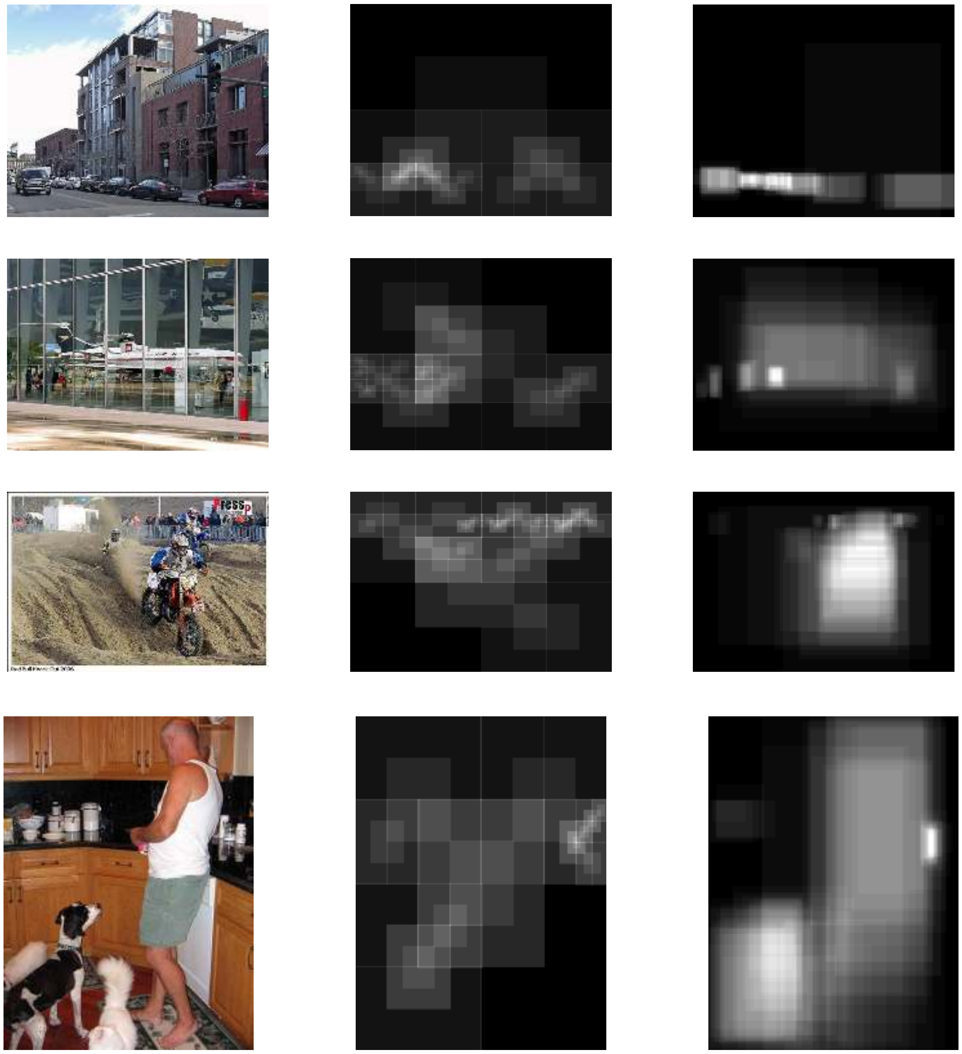

Figure 7. Example outputs of our algorithm. The left column shows the original image. The middle column shows the anchor regions induced by our adaptive search. The right column shows the top 100 adjacency predictions made around the anchor regions. The anchor regions and the adjacency predictions are superimposed into a figure at the same resolution of the original image. We note that the anchor regions and the region proposals in our approach are shared across object categories. For example, for the last image, the algorithm generates anchor regions at proper scales near the dogs, the person, and the bottles.

Method	boxes	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
AZ-Net	231	70.2	73.3	78.8	69.2	**59.9**	48.7	**81.4**	82.8	83.6	47.5	77.3	62.9	**81.1**	83.5	78.0	75.8	38.0	68.7	67.2	79.0	66.4
AZ-Net*	228	**70.4**	73.9	79.9	68.8	58.9	49.1	80.8	**83.3**	**83.7**	47.2	75.8	63.8	80.6	**84.4**	**78.9**	75.8	**39.2**	70.2	67.4	78.4	**68.3**
RPN	300	69.9	70.0	**80.6**	**70.1**	57.3	49.9	78.2	80.4	82.0	**52.2**	75.3	**67.2**	80.3	79.8	75.0	**76.3**	39.1	68.3	67.3	**81.1**	67.6
RPN*	300	68.5	74.1	77.2	67.7	53.9	**51.0**	75.1	79.2	78.9	50.7	**78.0**	61.1	79.1	81.9	72.2	75.9	37.2	**71.4**	62.5	77.4	66.4
FRCNN	2000	68.1	**74.6**	79.0	68.6	57.0	39.3	79.5	78.6	81.9	48.0	74.0	67.4	80.5	80.7	74.1	69.6	31.8	67.1	**68.4**	75.3	65.5

Table 1. Comparison on VOC 2007 test set using VGG-16 for convolutional layers. The results of RPN are reported in [22]. The results for Fast R-CNN are reported in [9]. The AZ-Net and RPN results are reported for top-300 region proposals, but in AZ-Net many images have too few anchors to generate 300 proposals. * indicates results without shared convolutional features. All listed methods use DCNN models trained on VOC 2007 trainval.

a model reportedly trained on VOC 2007 trainval. Correspondingly we compare it against our model trained on VOC 2007 trainval set. The comparisons concerning top-N regions are performed by ranking the region proposals in order of their confidence scores.

Figure 8 shows a comparison of recall at different IoU

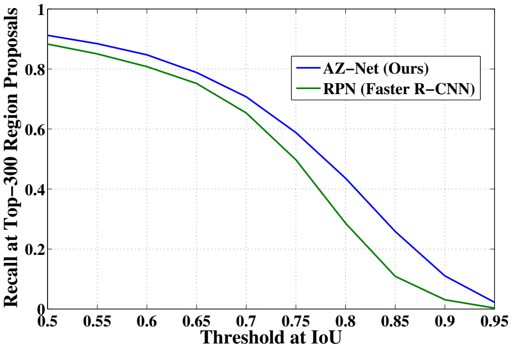

Figure 8. Comparison of recall of region proposals generated by AZ-Net and RPN at different intersection over union thresholds on VOC 2007 test. The comparison is performed at top-300 region proposals. Our approach has better recall at large IoU thresholds, which suggests that AZ-Net proposals are more accurate in localizing the objects.

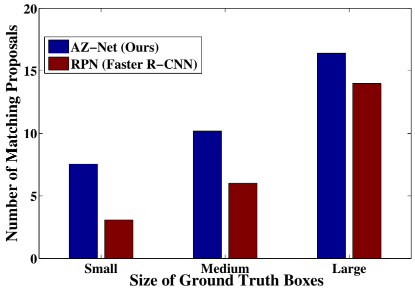

Figure 9. Number of proposals matched to ground truth (with IoU= 0.5). This shows proposals from AZ-Net are more concentrated around true object locations.

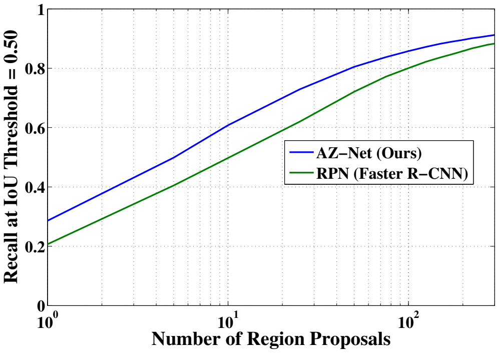

Figure 10. Comparison of recall of region proposals generated by AZ-Net and RPN at different number of region proposals on VOC 2007 test. The comparison is performed at IoU threshold 0.5. Our approach has better early recall. In particular, it reaches 0.6 recall with only 10 proposals.

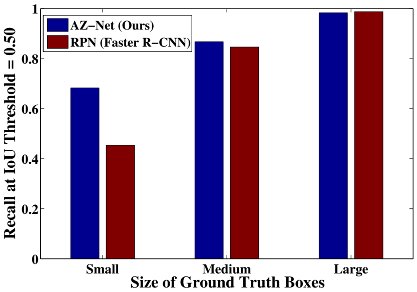

Figure 11. Comparison of recall of region proposals generated by AZ-Net and RPN for objects of different sizes on VOC 2007 test. The comparison is performed at IoU threshold 0.5 with top-300 proposals. Our approach has significantly better recall for small objects.

Method	Anchor Regions	Region proposals	Runtime (ms)
AZ-Net	62	231	171
AZ-Net*	44	228	237
RPN	2400	300	198
RPN*	2400	300	342
FRCNN	N/A	2000	1830

Table 2. Numbers related to the efficiency of the object detection methods listed in Table 1. The runtimes for RPN and Fast R-CNN are reported for a K40 GPU [22]. Our runtime experiment is performed on a GTX 980Ti GPU. The K40 GPU has larger GPU memory, while the GTX 980Ti has higher clock rate. * indicates unshared convolutional feature version.

thresholds. Our AZ-net has consistently higher recall than RPN, and the advantage is larger at higher IoU thresholds. This suggests our method generates bounding boxes that in general overlap with the ground truth objects better. The proposals are also more concentrated around objects, as shown in Figure 9.

Figure 10 shows a plot of recall as a function of the number of proposals. A region proposal algorithm is more efficient in covering objects if its area under the curve is larger. Our experiment suggests that our AZ-Net approach has a better early recall than RPN. That means our algorithm in general can recover more objects with the same number of region proposals.

Figure 11 shows a comparison of recall for objects with different sizes. The "small object" has an area less than 32^2, a "medium object" has an area between 32^2 and 96^2, and a "large object" has an area greater than 96^2, same as the definition in MSCOCO [18]. Our approach achieves higher recall on the small object subset. This is because when small objects are present in the scene our adaptive search strategy generates small anchor regions around them, as shown in Figure 7.

4.3. Efficiency of Adaptive Search

Our approach is efficient in runtime, as shown in Table 2. We note that this is achieved even with several severe in-

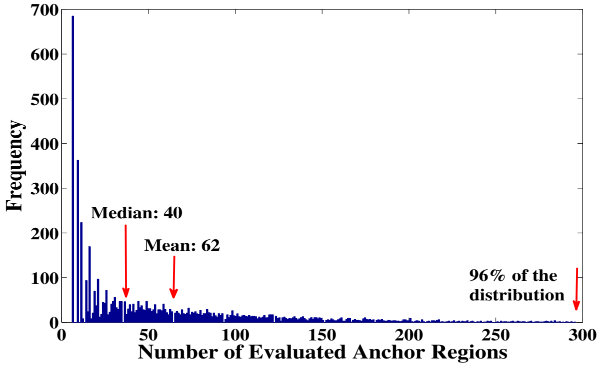

Figure 12. Distribution of the number of anchor regions evaluated on VOC 2007 test set. For most images a few dozen anchor regions are required. Note that anchors are shared across categories.

efficiencies in our implementation. First, for each image our algorithm requires several rounds of fully connected layer evaluation, which induces expensive memory transfer between GPU and CPU. Secondly, the Faster R-CNN approach uses convolutional computation for the evaluation of anchor regions, which is highly optimized compared to the RoI pooling technique we adopted. Despite these inefficiencies, our approach still achieves high accuracy at a state-of-the-art frame rate, using lower-end hardware. With improved implementation and model design we expect our algorithm to be significantly faster.

An interesting aspect that highlights the advantages of our approach is the small number of anchor regions to evaluate. To further understand this aspect of our algorithm, we show in Figure 12 the distribution of anchor regions evaluated for each image. For most images our method only requires a few dozen anchor regions. This number is much smaller than the 2400 anchor regions used in RPN [22] and the 800 used in MultiBox [5]. Future work could further capitalize on this advantage by using an expensive but more accurate per-anchor step, or by exploring applications to very high-resolution images, for which traditional non-adaptive approaches will face intrinsic difficulties due to scalability issues. Our experiment also demonstrates the possibility of designing a class-generic search. Unlike per-class search methods widely used in previous adaptive object detection schemes [3, 28] our anchor regions are shared among object classes, making it efficient for multi-class detection.

4.4. Results on MSCOCO

We also evaluated our method on MSCOCO dataset and submitted a "UCSD" entry to the MSCOCO 2015 detection challenge. Our post-competition work greatly improved accuracy with more training iterations. A comparison with other recent methods is shown in Table 3. Our model is trained with minibatches consisting of 256 regions sampled from one image, and 720k iterations in total. The results for RPN(VGG16) reported in [22] were obtained with an 8-GPU implementation that effectively has 8 and 16 images per minibatch for RPN and Fast R-CNN respectively, each trained at 320k training iterations. Despite the much shorter effective training iterations, our AZ-Net achieves similar mAP with RPN(VGG16) and is more accurate when evaluated on the MSCOCO mAP metric that rewards accurate localization.

Method	AP	AP IoU=0.50
FRCNN (VGG16) [9]	19.7	35.9
FRCNN (VGG16) [22]	19.3	39.3
RPN (VGG16)	21.9	42.7
RPN (ResNet)	37.4	59.0
AZ-Net (VGG16)	22.3	41.0

Table 3. The detection mAP on MSCOCO 2015 test-dev set. The RPN (ResNet) entry won the MSCOCO 2015 detection challenge. Updated leaderboard can be found in `http://mscoco.org`.

Our best post-competition model is still significantly outperformed by the winning "MSRA" entry. Their approach is a Faster-R-CNN-style detection pipeline, replacing the VGG-16 network with an ultra-deep architecture called Deep Residual Network [13]. They also report significant improvement from using model ensembles and global contextual information. We note that these developments are complementary to our contribution.

5. Conclusion and Future Work

This paper has introduced an adaptive object detection system using adjacency and zoom predictions. Our algorithm adaptively focuses its computational resources on small regions likely to contain objects, and demonstrates state-of-the-art accuracy at a fast frame rate.

The current method can be further extended and improved in many aspects. Better pre-trained models [13] can be incorporated into the current system for even better accuracy. Further refining the model to allow single-pipeline detection that directly predicts class labels, as in YOLO [21] and the more recent SSD [19] method, could significantly boost testing frame rate. Recent techniques that improve small object detection, such as the contextual model and skip layers adopted in Inside-Outside Net [1], suggest additional promising directions. It is also interesting to consider more aggressive extensions. For instance, it might be advantageous to use our search structure to focus high-resolution convolutional layer computation on smaller regions, especially for very high-resolution images.

Acknowledgment

This work is supported by the National Science Foundation grants CIF-1302438, CCF-1302588 and CNS-1329819, as well as Xerox UAC, and the Sloan Foundation.

References

[1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *arXiv preprint arXiv:1512.04143*, 2015.

[2] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, 2013.

[3] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[4] S. K. Divvala, D. Hoiem, J. H. Hays, A. Efros, M. Hebert, et al. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009.

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2155–2162. IEEE, 2014.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[8] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[9] R. Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[11] A. Gonzalez-Garcia, A. Vezhnevets, and V. Ferrari. An active search strategy for efficient object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3022–3031, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2129–2142, 2009.

[17] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.

[20] Y. Lu and T. Javidi. Efficient object detection for high resolution images. *arXiv preprint arXiv:1510.01257*, 2015.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[25] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013.

[26] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006.

[27] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[28] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon. Attentionnet: Aggregating weak directions for accurate object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.