Improving Small Object Detection

Harish Krishna, C.V. Jawahar
CVIT, KCIS
International Institute of Information Technology
Hyderabad, India

***Abstract*—While the problem of detecting generic objects in natural scene images has been the subject of research for a long time, the problem of detection of small objects has been largely ignored. While generic object detectors perform well on medium and large sized objects, they perform poorly for the overall task of recognition of small objects. This is because of the low resolution and simple shape of most small objects. In this work, we suggest a simple yet effective upsampling-based technique that performs better than the current state-of-the-art for end-to-end small object detection. Like most recent methods, we generate proposals and then classify them. We suggest improvements to both these steps for the case of small objects.**

***Keywords*-object detection; super-resolution**

I. Introduction

Though the problem of object detection in natural scenes has seen a lot of research, especially since the development of deep ConvNets, it is far from being solved, particularly for the case of small objects. An object is considered small if it occupies only a tiny portion of the image (less than 1% of the image area). This problem is very relevant in many of the challenging research applications of today - like detecting pedestrians, traffic sings and cars on roads and areal imagery.

Detecting small objects is a challenging task. Firstly, it is very hard to distinguish small objects from generic clutter in the background. This makes it hard for many of the standard detectors that rely on 'objectness' due to the drastic increase in the number of possible locations. Secondly, the activations of small objects become smaller with each pooling layer as an image passes through a standard CNN architecture like VGG16. For example, if an object has a size of 32×32, it will represent at most 1 pixel after the block5_pool layer in VGG16. Such activations can be easily missed. Thirdly, most small objects have simple shapes that are not decomposable into smaller parts. On the other hand, popular CNN-based detectors excel at learning hierarchical features. Lastly, there is no large publically-available dataset for small objects. While MS COCO and VOC2012 have specific instances of objects being small, there are not any dedicated large datasets for small objects. Also, much of the prior experience and intuitions are on datasets with larger objects. While the mean Average Precision using the state-of-art end-to-end detectors on a dataset like PASCAL VOC is 76.3% [1], the state-of-art on a dataset with only small objects is just 27% [2].

Figure 1. The problem of small object detection is hard because of a much larger search space, background clutter and a weak signal after passing through standard convolutional layers. For example, the mouse (in the green box) is a small object and is hard to spot among the various other objects of similar sizes present.

A class of popular detection techniques in recent years involve suggesting several object proposal regions which are then classified by a deep CNN model [3]. These techniques are successful because the features obtained using a deep CNN are more discriminative than hand-engineered features. Unlike earlier times, when dense sliding windows were used to look at probable object regions, algorithms that look at low-level cues suggest much fewer and sparser windows. These proposal generation methods made way for a Region Proposal Network (RPN) [1], which was found to not only generate better proposals, but also greatly quicken the detection process when the weights of the convolutional layers are shared with that of the detector. In our work, we use a similar approach: an RPN generates proposals which are then classified by a deep CNN. The RPN as used in the de-facto standard detection algorithm, Faster RCNN [1], misses several small objects because of the large size of anchor boxes. Keeping this in mind, we study the size of anchor boxes for a dataset. We then show that this choice of anchor box size beats other existing methods.

Works like [4] suggest that the classification performance increases with the image size. One way to approach the problem would be to upsample the entire image and apply standard techniques on this image. However, the computational cost increases

exponentially as the size of the image increases. Instead, we can upsample small proposal regions. Here, we take inspiration from recent works that convert low resolution images to high resolution by hallucinating the intermediate values. We develop a method that upsamples proposal regions with the hope of improving the overall classification performance.

The main contributions of this work are as follows:

- We formulate finding the appropriate sizes of the anchor boxes mathematically and perform detailed experiments to show the effectiveness in their choice. We show that this gives us the state of the art end-to-end trainable network for this dataset.
- We show how network-based super resolution techniques can help improve performance

II. Related Work

In the pre-deep learning era, works used specially-crafted features for problems like vehicle detection in aerial imagery [5]. However, since the emergence of deep learning, the task of learning discriminative features has been usurped by CNNs.

Many approaches have emerged in recent times that do not use region proposals. YOLO [6] divides the image into a grid and predicts class labels and bounding boxes for each cell of the grid. Another interesting approach is Single Shot Detection [7] which fixes boxes of various scales where objects may lie and scores presence of objects for each such box during test time. However, proposal-based methods have been shown to outperform all proposal-free methods as far as recall and accuracy are concerned [8]. [9] and [10] were popular methods for proposal generation that used low and mid level features. The idea of using deep networks to suggest proposals has gained traction in recent years. While Deepbox [11] reranks proposals generated by Edgeboxes [10], DeepProposal [12] uses an inverse cascade that goes from the final to the initial convolutional layers of the CNN. The Regional Proposal Network introduced in [1] can share convolutional layers with the classifier network. Here, anchor boxes of multiple scales and aspect ratios slide across the feature map from the last convolutional layer. The RPN acts as an attention mechanism and tells the detector where to look.

[3] was made fast in [13] with the introduction of the RoIPooling layer which maps images of any dimension to a feature map of fixed dimension. Faster RCNN [1] is essentially two components - an RPN which feeds to Fast RCNN. All these approaches predict a bounding box and probability of belonging to that class for every class. It was observed that sharing weights between the proposal network and detector not only significantly reduces running time, but also improves performance.

Several detectors [14] [15] [16] have emerged that build upon the faster RCNN framework. However, most of them only fleetingly mention the case of small objects. [17] [18] and [19] look at modifying the fast RCNN architecture for the problem of logo, face and pedestrian detection respectively, all having instances of small objects.

Small object RCNN [2] is perhaps the first paper to focus on the problem of small object detection. They introduce a small dataset, an evaluation metric and provide a baseline score. They suggest modifications to the Region Proposal Network and show an improvement in recall and mean average precision. Specifically, they suggest choosing smaller anchor box sizes and attaching the the anchor boxes to conv4_3 rather than conv5_3 of VGG16. They go on to argue that the RoI pooling layer may not preserve much information of small objects and hence follow the RCNN framework. In this work, we build upon their ideas and show how to make changes to perform just as well in an end-to-end pipeline.

The low and high resolution spaces are different and hence a one-network-fits-all approach may not work, since most large datasets are heavily skewed towards large and high resolution objects. Papers like [20] and [21] which address activity recognition in low resolution videos, map both low and high resolution to a common space. [20] shows that the problem of working in such low resolutions is harder as the space is very sensitive to even slight translations. [21] uses high resolution to assist learning filters for the low resolution domain. We use a different approach in this work. We leverage the fact that classifiers work well in the high resolution domain. we use super-resolution to transfer the task of classification to a network pretrained on high resolution images.

The problem of image denoising and image super-resolution are well studied and have seen numerous approaches in the deep learning era. [22] showed how to train an end-to-end neural network for the task of Single Image Super Resolution. [23] and [24] use recursive CNNs and sub-pixel level convolution while [25] uses an auto-encoder model. We use [25] because of the relative ease to experiment with it.

III. Approach

A. Proposal Generation

We make several modifications to the Faster RCNN [1] Region Proposal Network so that it performs well for the specific tasks of small objects. [13] suggested using powers of two like $128^2, 256^2, 512^2$ for anchor box sizes. While these anchor box sizes were shown to work for large objects, these are too large for small objects.

We follow [17] to theoretically estimate the size of the anchor boxes. The performance metric commonly used in detection to decide if a proposal is correct is to see if the Intersection over Union is greater than a threshold (typically 0.5). More formally, like in Figure 3, if S_{gt} is the side length of the ground truth object and S_A is the side length of an anchor object, and d is the displacement of the two boxes, the IoU is defined as

$$\frac{(S_{gt} - d)^2}{S_{gt}^2 + S_A^2 - (S_{gt} - d)^2} \tag{1}$$

We want the quantity described in equation 1 to always be greater than a threshold t. In other words, we want

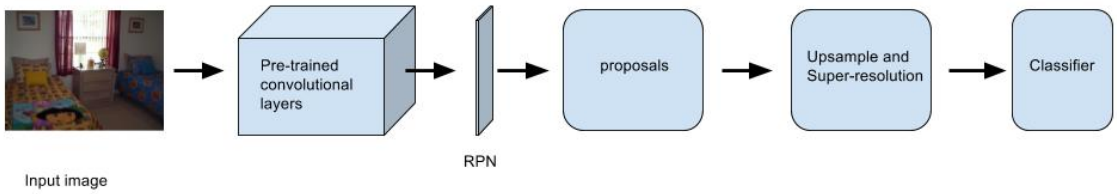

Figure 2. The pipeline for our approach - discriminative features that can be used for proposal generation and classification are obtained by passing the image through a standard pre-trained deep convolutional net. A region proposal network generates regions of interest based on the objectness of the region. These proposals are upsampled and flow through a super-resolution network after which they are classified.

$\min \text{IoU} \geq t$. We will only consider the case when $S_A \geq S_{gt}$. Solving for S_A, we get,

$$(S_{gt} - d)^2(1 + t^{-1}) - S_{gt}^2 \geq S_A^2 \tag{2}$$

The size of S_{gt} is dependent on our dataset. The worst possible overlap happens when the stride is largest. The value of d is dependent on the number of downsampling layers the image undergoes. Since we are fixing the anchor boxes after the fifth convolutional block, $d = 16$.

Thus, from equation 2, we get an upper limit for the size of an anchor box for a given ground truth image. This also gives us a bound on the size of the ground truth image for which our method will work, since S_A^2 needs to be positive. We also note from [17] that $S_{gt}/S_A \leq 1/\sqrt{2}$. Here, we assume the bounding boxes to be squares. The same kind of relationship holds for other aspect ratios also. We analyze the size of objects in our dataset (Figure 4) and get our anchor boxes as $\{16, 25, 32, 45, 64, 90\}$.

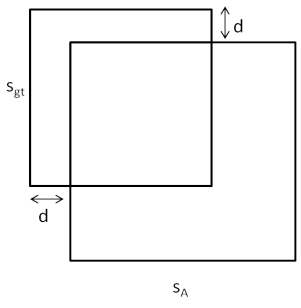

Figure 3. The choice of the size of anchor boxes is such that for all possible ground truth sizes and strides, an anchor box will have an overlap greater than a threshold.

Unlike [2], we train the RPN in an end-to-end manner to predict bounding boxes as well as class scores. This has been shown to improve performance [3]. Also, unlike them, we add batch normalization layers after every block in our VGG16 convnet.

B. Upsampling

According to [2], an RoI pooling layer loses discriminative features. They find that classifying upscaled proposals gives a better performance than classifying patches despite the fact that aggressively upscaling adds undesirable artifacts and results in a noisy image. We instead leverage recent work in super-resolution to denoise the image and get a high resolution image.

A fairly common approach in super-resolution is to first upscale the low resolution image and use a CNN to denoise the image. The upscaling operation uses a handcrafted filter like bilinear or bicubic interpolation. Such filters are but special cases of a deconvolutional layer as argued in[26]. A deconvolutional layer can be thought of as a convolutional layer with fractional stride. A network with deconvolutional layers might learn more complex, non-linear upsampling, specific to the dataset.

We use an implementation of [25] which uses a convolutional-deconvolutional network with skip connections. The usage of skip connections in the autoencoder makes the network easier to train while also ensuring that the deconvolutional layers can use the semantic information captured by the convolutional layers. The network is pretrained on a large dataset like Imagenet.We train a CNN on the super-resolved train images like in [2]. We use the VGG16 weights for the convolutional layers of a classifier which we use to classify the upsampled test proposals.

IV. Experiments and Discussion

A. Dataset and Performance Metric

The dataset used in our experiments is the Small Object Dataset introduced in [2]. This is a collection of 4925 images from Microsoft COCO and the SUN dataset. Ten categories were chosen such that a typical instance of the object was no larger than 30cm in the physical world. Among all images which contained these classes, only those images which contained objects occupying a small area were chosen. The dataset is quite challenging for the following reasons :

- A significant percentage of the instances occupy less than 16 $\times$ 16 pixels (Figure 4). This is on average 0.2% of the image area. In contrast, datasets like VOC have objects that occupy 14% of the image area on average.
- There is class imbalance - while the category mouse has 1739 instances, the category tissue box has only 100 instances.
- The absence of high-resolution images for these categories is a major drawback. If high-resolution images were present,.
- The small size of the dataset with just 6000 train instances limits proper fine-tuning of existing methods for the case of small objects, let alone train full end-to-end systems from scratch.

We evaluate with the commonly used performance metric in detection: a predicted bounding box is considered a correct detection if the Intersection over Union (IoU) overlap with the ground truth bounding box is greater than 0.5. The performance of the whole detection algorithm is measured using mean Average Precision (mAP), which essentially denotes the area under the precision-recall curve.

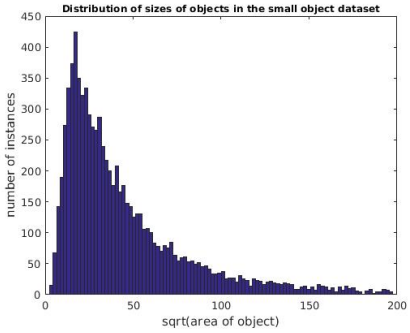

Figure 4. Size distribution in the dataset. Most objects are too small to be detected by the default anchor box sizes

B. Proposal Generation

We evaluate the choice of anchor box sizes against the default choice in Faster RCNN and those used by [2]. We use the standard faster RCNN framework with VGG16 as the backbone. The anchor boxes are attached to the conv5 layer of VGG16. The aspect ratios are the same as that of faster-RCNN, namely, 1:1, 1:2 and 2:1. The network is finetuned with a learning rate of 0.001 and a gamma of 0.1 for 50000 iterations. We compare the mAP upon taking the top 1000 proposals ranked on confidence of the proposal belonging to a non-background class for every test image.

While [2] uses the RCNN framework, we prefer experimenting with faster RCNN. The advantage of using Faster-RCNN is that apart from being much faster during testing and training, it does not require the storage of generated proposals which takes up a lot of memory. Also, we empirically found that attaching the anchor boxes to conv5 performs better than attaching to conv4 or conv3 when training end-to-end.

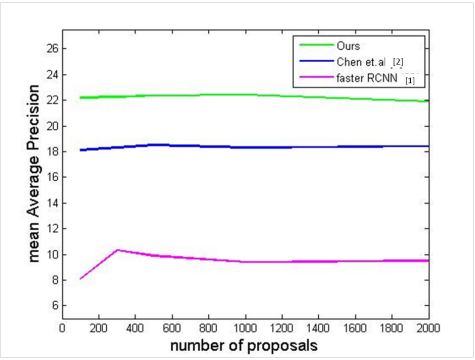

Figure 5. mAP vs number of proposals: our choice of anchors performs better than other methods even for much fewer proposals. It is interesting that the performance stagnates or even decreases with more proposals being considered.

We see that our choice of anchors performs better than the default faster RCNN (Table 1). This is expected since the smallest anchor box of size 128 is much bigger than all instances in the dataset. [2] choose anchor boxes of size 16,40 and 100. This performs much better than the default values. The anchor box sizes we propose cover the entire range of the small object sizes in the dataset.

To show that these anchors are adequate, we add two more anchor boxes of sizes 40 and 100. We observe that despite adding more anchor boxes, the performance slightly reduces to 21.9%. This is due to the larger number of proposals generated, which would include more proposals of generic objects that are part of the background. Because of the relatively simpler shapes of small objects, objects of these background classes might be confused for classes in our dataset.

Table I
SIZE OF ANCHORS VS PERFORMANCE. OUR CHOICE OF ANCHORS PERFORMS BETTER THAN THE DEFAULT FASTER RCNN ANCHORS AND THOSE USED IN [2] IN THE END-TO-END FASTER RCNN PIPELINE.

Anchor Box sizes	mean Average Precision (%)
faster RCNN [1]	9.4
Chen et al. [2]	18.3
Ours	22.4

C. Number of Proposals

We next compare the quality of the generated proposals with the number of proposals we consider for every test image and the choices for anchor box sizes. We continue to use the faster RCNN framework. Proposals that belong to a non-background class are ranked based on the classifier score and the top-k are chosen for the calculation of mAP.

As observed in Figure 6, we notice that our choice of anchor boxes beats [2] and the default Faster RCNN at all choices of number of proposals. We attribute the decrease in performance with more proposals to the observation that most true positives occur in the top few proposals itself, while there is an explosion in the number of false positives as more proposals are considered. These false

Figure 6. Exemplar results of our method on the small object dataset. The detections are shown in green boxes. The last row shows failure cases. In the first image, an armchair handle is classified as a mouse due to their similarity in shape. The second image shows a missed detection of the clock because it was too faint. The phone in the third image has two components quite far apart for our proposal generation method to consider as a single object.

Table II
RESULTS OF OUR END-TO-END METHOD ON THE 10-CLASS SMALL OBJECT DATASET. THE LAST COLUMN IS THE WEIGHTED AVERAGE PRECISION. THE FIRST ROW IS THE END-TO-END TRAINED FASTER RCNN NETWORK WITH OUR ANCHORS. THE SECOND ROW GIVES THE PERFORMANCE WITH THE RCNN PIPELINE AND UPSCALING [2]. THE THIRD ROW SHOWS THE IMPROVEMENT WITH SUPER-RESOLUTION.

Method	Mouse	Phone	Switch	Outlet	Clock	T. paper	T. box	Faucet	Plate	Jar	Average
Faster RCNN	57.7	14.3	15.4	22.1	26.0	31.7	8.1	35.1	11.9	3.1	22.6
RPN and upscaling	56.8	16.4	31.1	29.4	31.9	29.4	23.4	31.3	9.3	4.2	24.8
RPN and super-resolution	60.1	16.9	16.2	23.5	30.3	34.1	12.8	38.0	15.2	4.7	25.2

positives arise because of the similarity in shape of generic background objects with the classes of interest.

D. Super-resolution

To investigate the effect of using supervised upsampling techniques, we parallel the approach followed by [2]. Here, we use the RCNN framework wherein we use our trained RPN with our choice of anchor box sizes to generate region proposals. We then upsample the proposals and use our trained classifier to rerank the scores for each proposal. The results are summarized in Table 2, where we observe that a super-resolution network improves performance. This improvement can be attributed to how the filters learned by convolutional layers don't perform just as well on low-resolution images. Low-resolution images, when upscaled, have blurry edges and are rather pixelated. However, these datasets are trained on medium and high resolution images and hence the filters work best for this resolution. This difference in resolution is mitigated by super-resolution.

V. CONCLUSION

In this work, we explore the choice of anchor sizes for small object region proposal generation. We also showed how deep super-resolution methods improve the performance for small object classification. It will be interesting to see how super-resolution and cues such as context, segmentation mask and saliency which have been shown

to work for generic object detection, can be incorporated in an end-to-end faster RCNN framework for the case of small objects.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS 2015*.

[2] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-cnn for small object detection," in *13th ACCV Proceedings*, 2017.

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

[4] C. Eggert, A. Winschel, D. Zecha, and R. Lienhart, "Saliency-guided selective magnification for company logo detection," in *ICPR 2016*.

[5] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery," *J. Vis. Comun. Image Represent.*, 2016.

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

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *arXiv 1512.02325*, 2015.

[8] 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," *arXiv: 1611.10012*, 2016.

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

[10] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV 2014*.

[11] W. Kuo, B. Hariharan, and J. Malik, "Deepbox: Learning objectness with convolutional networks," in *ICCV 2015*.

[12] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool, "Deepproposals: Hunting objects and actions by cascading deep convolutional layers," *IJCV*, 2017.

[13] R. B. Girshick, "Fast R-CNN," *arXiv 1504.08083*, 2015.

[14] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: towards accurate region proposal generation and joint object detection," in *CVPR 2016*.

[15] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *arXiv 1703.06870*, 2017.

[16] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *CVPR 16*.

[17] C. Eggert, D. Zecha, S. Brehm, and R. Lienhart, "Improving small object proposals for company logo detection," in *ICMR 2017*.

[18] P. Hu and D. Ramanan, "Finding tiny faces," *arXiv*, 2017.

[19] J. Li, X. Liang, S. Shen, T. Xu, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *arXiv 1510.08160*, 2015.

[20] M. S. Ryoo, K. Kim, and H. Jong Yang, "Extreme Low Resolution Activity Recognition with Multi-Siamese Embedding Learning," *ArXiv e-prints*, 2017.

[21] J. Chen, J. Wu, J. Konrad, and P. Ishwar, "Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions," in *WACV*, 2017.

[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *ECCV 2014*.

[23] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," *arXiv*, 2016.

[24] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," *arXiv 1511.04491*, 2015.

[25] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," *arXiv:1606.08921*, 2016.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR 2015*.