Region Proposal by Guided Anchoring

secortot

Two paradigms for object detection

- One-stage : SSD/ YOLO (Fast but not accurate enough)
  Places anchor boxes densely over feature maps/images

- Two-stage: R-CNN   (accurate but not fast enough)
  RPN and detection

What's the problem with anchor?

- Predefined scale and aspect ratio are not easy to fit all objects well. (even dimensional clustering in YOLO-V2[CVPR 2017])
- Limited by the number of targets in the image, cause serious imbalance between positive and negative samples. (He et al. Focal Loss)
- Need to generate dense anchors to guarantee the recall.

Overview

- Use semantic features to guide anchoring.
- A new anchoring scheme with the ability to predict non-uniform and arbitrary shaped anchors other than dense and predefined ones.
- Jointly predict the locations where the center of objects of interest are likely to exist as well as the scales and aspect ratios at different locations.
- Anchor-guided feature adaption by deformable convolution.
- Improve the performance, achieve 9.1% higher recall with 90% fewer anchors.

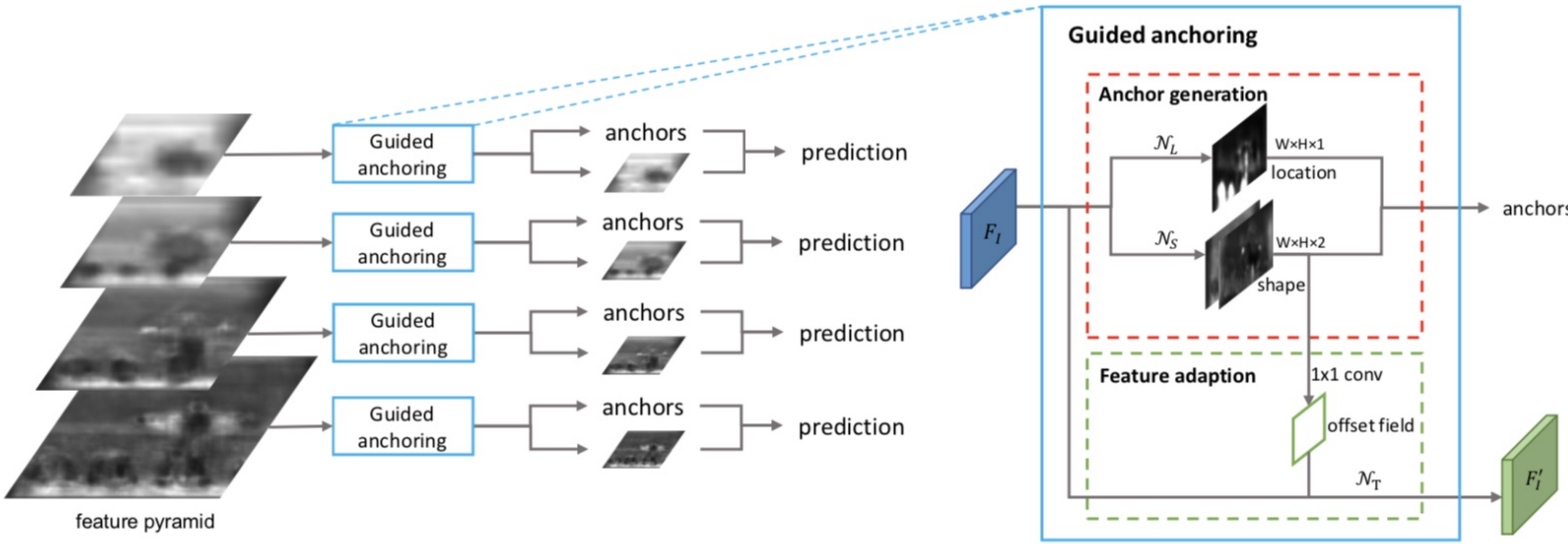


Figure 1: An illustration of our framework. For each output feature map in the feature pyramid, we use an anchor generation module wi
two branches to predict the anchor location and shape, respectively. Then a feature adaption module is applied to the original feature m

Guided Anchoring: Anchor Location Prediction

- Prior knowledge: Objects are not distributed evenly over the images. The scale of an object is also closely related to the imagery content, its location and geometry of the scene.

- The location prediction beam outputs a probability distribution map. The map which has the same size as the input feature map. The probability map is obtained by a 1*1 convolution on the output feature map, and the probability value is activated by the element-wise sigmoid.

- Locate anchor according to a threshold

- For (i, j) in feature map ,((i+1/2)*s, (j+1/2)*s) in image

Guided Anchoring-Anchor Shape Prediction

- Use transforms $w = \sigma \cdot s \cdot e^{dw}, \quad h = \sigma \cdot s \cdot e^{dh}.$ (2) to constrain network outputs ([-1,1]).
- Comprises a 1 × 1 convolutional layer that yields a two-channel map that contains the values of d_w and d_h.
- Has better ability to capture those extremely tall or wide objects.

Anchor-Guided Feature Adaptation

- Different scales of objects even at the same level.
- Large objects ought to obtain larger receptive field, small objects vice versa.
- $\mathbf{f}_i' = \mathcal{N}_T(\mathbf{f}_i, w_i, h_i)$ by 3 * 3 deformable convolution.
- Classification and bounding boxes regression.

Training

- Loss function: Multi-task loss.
- Location; Shape; Classification; Regression.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{loc} + \lambda_2 \mathcal{L}_{shape} + \mathcal{L}_{cls} + \mathcal{L}_{reg}. \tag{4}$$

Anchor location targets

- Use the GT bounding box to guide label generation.
- A binary label map where 1 represents a valid location to place an anchor and 0 otherwise.
- More anchors near the target center, fewer otherwise.
- Anchor location target represents for positive, ignore or negative.

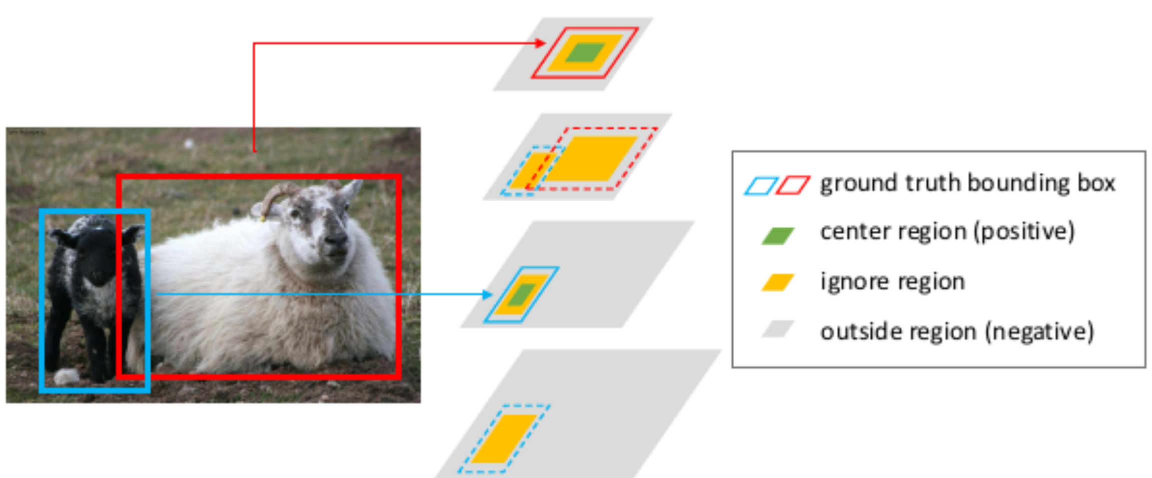
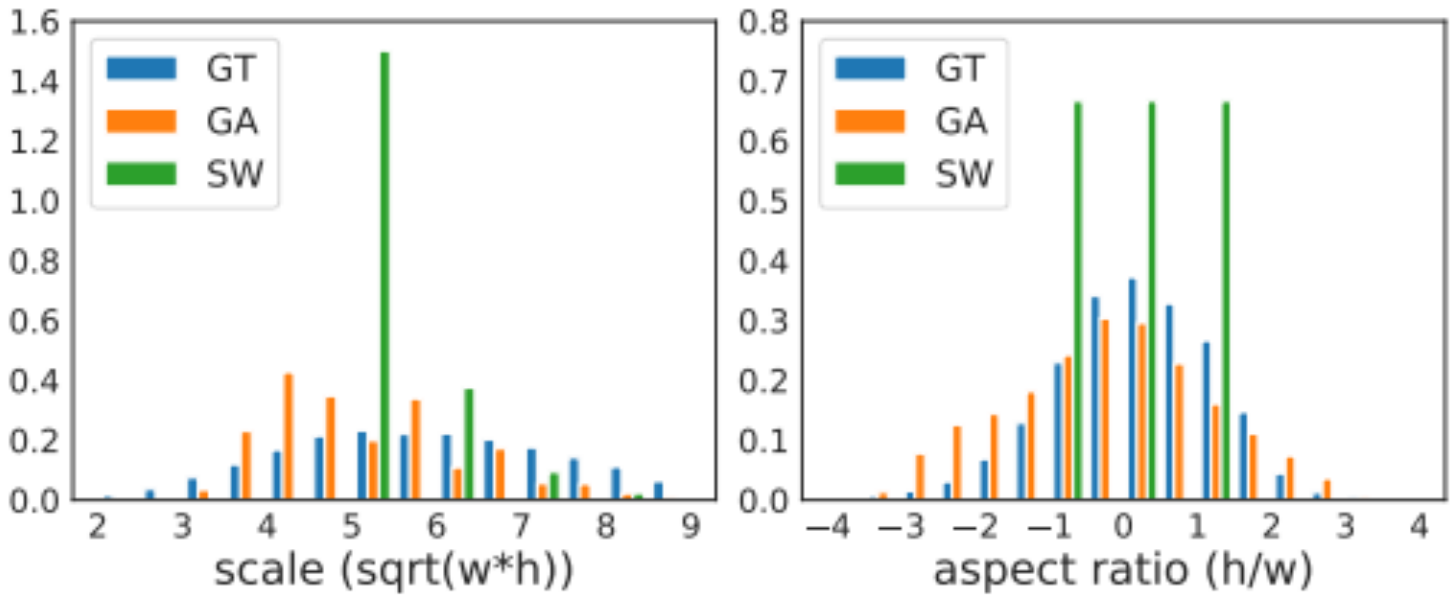


Figure 2: Anchor location target for multi-level features. We assign ground truth objects to different feature levels according to their scale, and define CR, IR and OR respectively. (Best viewed in color.)

Anchor shape targets

- Anchor selection for predefined anchor can use IOU, but how for guided anchor?
- $$\mathrm{vIoU}(a_{\mathbf{wh}}, \mathrm{gt}) = \max_{w>0,h>0} \mathrm{IoU}_{normal}(a_{wh}, \mathrm{gt}),$$
- Sample common values of *w* and *h* (*9* in experiment).
- A variant of bounded IOU loss.
- $$\mathcal{L}_{shape} = \mathcal{L}_1(1 - \min(\frac{w}{w_g}, \frac{w_g}{w})) + \mathcal{L}_1(1 - \min(\frac{h}{h_g}, \frac{h_g}{h})).$$



(a) (b)

Figure 6: (a) Anchor scale and (b) aspect ratio distributions of different anchoring schemes. The x-axis is reduced to log-space by apply $\log_2(\cdot)$ operator. GT, GA, SW indicates ground truth, guided anchoring, sliding window, respectively.

GA anchor is more similar to the distribution of GT than sliding window.

Advantages of GA-RPN proposals over RPN proposals:

1. Larger positive proposals.
2. Emphasis the effect of the ratio of high-IoU proposals.

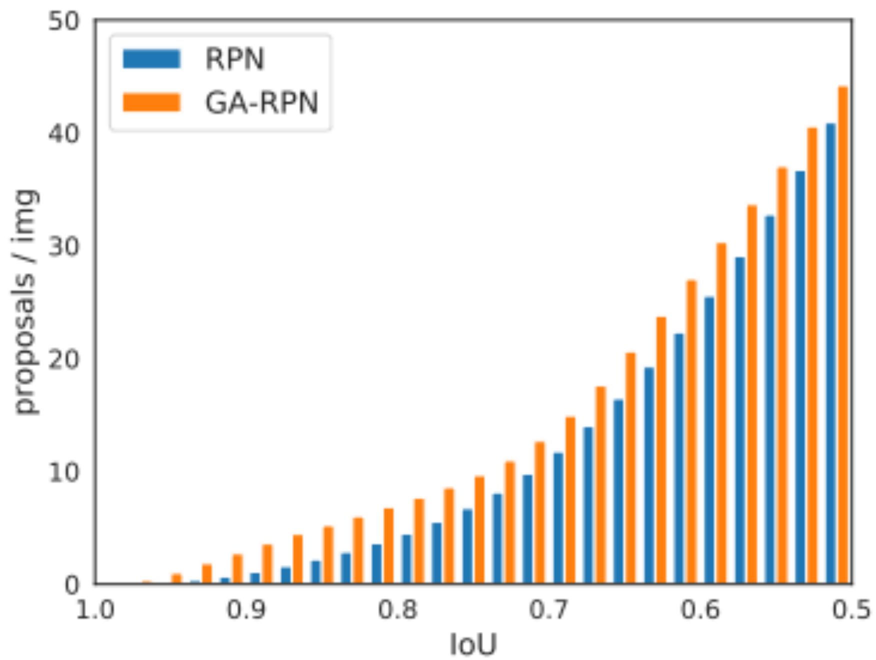


Figure 3: IoU distribution of RPN and GA-RPN proposals. We show the accumulated proposal number with increasing IoUs.

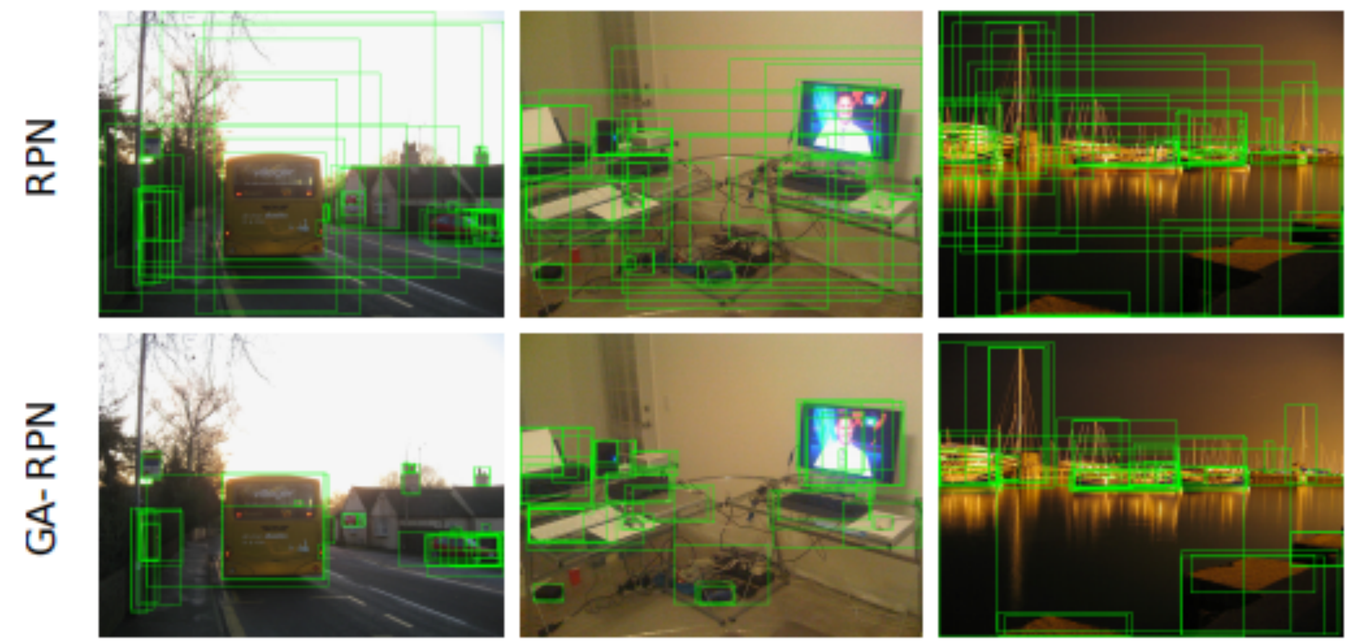


Figure 5: Examples of RPN proposals (top row) and GA-RPN proposals (bottom row).

Table 2: Detection results on MS COCO 2017 *test-dev*.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Fast R-CNN	37.1	59.6	39.7	20.7	39.5	47.1
GA-Fast-RCNN	**39.4**	59.4	42.8	21.6	41.9	50.4
Faster R-CNN	37.1	59.1	40.1	21.3	39.8	46.5
GA-Faster-RCNN	**39.8**	59.2	43.5	21.8	42.6	50.7
RetinaNet	35.9	55.4	38.8	19.4	38.9	46.5
GA-RetinaNet	**37.1**	56.9	40.0	20.1	40.1	48.0

Table 1: Region proposal results on MS COCO.

Method	Backbone	AR_{100}	AR_{300}	AR_{1000}	AR_S	AR_M	AR_L	runtime (s/img)
SharpMask [27]	ResNet-50	36.4	-	48.2	6.0	51.0	66.5	0.76 (unfair)
GCN-NS [25]	VGG-16 (SyncBN)	31.6	-	60.7	-	-	-	0.10
AttractioNet [11]	VGG-16	53.3	-	66.2	31.5	62.2	77.7	4.00
ZIP [18]	BN-inception	53.9	-	67.0	31.9	63.0	78.5	1.13
RPN	ResNet-50-FPN	47.5	54.7	59.4	31.7	55.1	64.6	**0.09**
	ResNet-152-FPN	51.9	58.0	62.0	36.3	59.8	68.1	0.16
	ResNeXt-101-FPN	52.8	58.7	62.6	37.3	60.8	68.6	0.26
RPN+9 anchors	ResNet-50-FPN	46.8	54.6	60.3	29.5	54.9	65.6	0.09
RPN+Iterative	ResNet-50-FPN	49.7	56.0	60.0	34.7	58.2	64.0	0.10
RefineRPN	ResNet-50-FPN	50.2	56.3	60.6	33.5	59.1	66.9	0.11
GA-RPN	ResNet-50-FPN	**59.2**	**65.2**	**68.5**	**40.9**	**67.8**	**79.0**	0.13