Generating Positive Bounding Boxes for Balanced Training of Object Detectors

Kemal Oksuz, Baris Can Cam, Emre Akbas*, Sinan Kalkan*
Department of Computer Engineering
Middle East Technical University, Ankara, Turkey
{kemal.oksuz, can.cam, eakbas, skalkan}@metu.edu.tr

Abstract

In this document, we provide supplementary material that were omitted in the submitted manuscript due to space constraints. This supplementary material includes the proofs for the theorems, some more details on the BB generator including derivations for equations 4 and 5 in the paper and computing the feasible space in the bottom right part, and finally some implementation details. As for implementation details, we discuss how the pRoI generator is integrated into the training, how pRoI generator uses BB generator as a subroutine and configurations of BB sources.

1. The Properties of IoU$(B, \bar{B})$

Following upon the notation in Section 3 of the paper, we introduce the following properties. For clarity we assume that intersection of two boxes is greater than 0 and the last pixel is not taken into account (i.e. instead of $\mathrm{A}(B) = (x_2 - x_1 + 1)$, we adopted $\mathrm{A}(B) = (x_2 - x_1)$).

Theorem 1. $\mathrm{IoU}(B, \bar{B})$ *is scale-invariant.*

Proof. Assume that $k_x > 0$ and $k_y > 0$ are the scaling factors in the x and y axes respectively and B_s, $\bar{B}_s$ are the scaled boxes. We show that $\mathrm{IoU}(B_s, \bar{B}_s) = \mathrm{IoU}(B, \bar{B})$ as follows:

$$\mathrm{IoU}(B_s, \bar{B}_s) = \frac{I(B_s, \bar{B}_s)}{A(B_s) + A(\bar{B}_s) - I(B_s, \bar{B}_s)} \tag{1}$$

$$= \frac{(\min(k_x\bar{x_2}, k_x x_2) - \max(k_x\bar{x_1}, k_x x_1)) \times (\min(k_y\bar{y_2}, k_y y_2) - \max(k_y\bar{y_1}, k_y y_1))}{(k_x x_2 - k_x x_1) \times (k_y y_2 - k_y y_1) + (k_x\bar{x_2} - k_x\bar{x_1}) \times (k_y\bar{y_2} - k_y\bar{y_1}) - I(B_s, \bar{B}_s)} \tag{2}$$

$$= \frac{k_x(\min(\bar{x_2}, x_2) - \max(\bar{x_1}, x_1)) \times k_y(\min(\bar{y_2}, y_2) - \max(\bar{y_1}, y_1))}{k_x(x_2 - x_1) \times k_y(y_2 - y_1) + k_x(\bar{x_2} - \bar{x_1}) \times k_y(\bar{y_2} - \bar{y_1}) - I(kB, k\bar{B})} \tag{3}$$

$$= \frac{k_x k_y I(B, \bar{B})}{k_x k_y(x_2 - x_1) \times (y_2 - y_1) + k_x k_y(\bar{x_2} - \bar{x_1}) \times (\bar{y_2} - \bar{y_1}) - k_x k_y I(B, \bar{B})} \tag{4}$$

$$= \frac{k_x k_y I(B, \bar{B})}{k_x k_y \left((x_2 - x_1) \times (y_2 - y_1) + (\bar{x_2} - \bar{x_1}) \times (\bar{y_2} - \bar{y_1}) - I(B, \bar{B})\right)} \tag{5}$$

$$= \mathrm{IoU}(B, \bar{B}) \tag{6}$$

Eq. 1 defines the IoU and Eq. 2 replaces area and intersection definitions. In Eq. 3, we use the property that multiplying by a positive scalar does not change minimum and maximum of two numbers. Eq. 4 incorporates the intersection definition. Eq. 5 gets the denominator in the $k\hat{k}$ parenthesis, which simplifies the term to the definition of $IoU(B, \bar{B})$. □

Theorem 2. *IoU*$(B, \bar{B})$ *is translation-invariant.*

Proof. Assuming that $k_x \in \mathrm{R}$ and $k_y \in \mathrm{R}$ are the perturbation in the x and y axis respectively and B_t, $\bar{B}_t$ are the perturbed boxes. We show that $\mathrm{IoU}(B_t, \bar{B}_t) = \mathrm{IoU}(B, \bar{B})$ as follows:

*Equal contribution for senior authorship.

$$\mathrm{IoU}(B_t, \bar{B}_t) = \frac{I(B_t, \bar{B}_t)}{A(B_t) + A(\bar{B}_t) - I(B_t, \bar{B}_t)} \tag{7}$$

$$= \frac{(\min(\bar{x_2} + k_x, x_2 + k_x) - \max(\bar{x_1} + k_x, x_1 + k_x)) \times (\min(\bar{y_2} + k_y, y_2 + k_y) - \max(\bar{y_1} + k_y, y_1 + k_y))}{((x_2 + k_x) - (x_1 + k_x)) \times ((y_2 + k_y) - (y_1 + k_y)) + ((\bar{x_2} + k_x) - (\bar{x_1} + k_x)) * ((\bar{y_2} + k_y) - (\bar{y_1} + k_y)) - I(B_t, \bar{B}_t)} \tag{8}$$

$$= \frac{(\min(\bar{x_2}, x_2) + k_x - \max(\bar{x_1}, x_1) - k_x) \times (\min(\bar{y_2}, y_2) + k_y - \max(\bar{y_1}, y_1) - k_y)}{(x_2 + k_x - x_1 - k_x) \times (y_2 + k_y - y_1 - k_y) + (\bar{x_2} + k_x - \bar{x_1} - k_x) \times (\bar{y_2} + k_y - \bar{y_1} - k_y) - I(B_t, \bar{B}_t)} \tag{9}$$

$$= \frac{(\min(\bar{x_2}, x_2) - \max(\bar{x_1}, x_1)) \times (\min(\bar{y_2}, y_2) - \max(\bar{y_1}, y_1))}{(x_2 - x_1) \times (y_2 - y_1) + (\bar{x_2} - \bar{x_1}) \times (\bar{y_2} - \bar{y_1}) - I(B_t, \bar{B}_t)} \tag{10}$$

$$= \frac{I(B, \bar{B})}{A(B) + A(\bar{B}) - I(B, \bar{B})} \tag{11}$$

$$= \mathrm{IoU}(B, \bar{B}) \tag{12}$$

Again, Eq. 8 replaces area and intersection definitions in the IoU definition. In Eq. 9, we use the property that adding a scalar to numbers adds a scalar to the minimum and maximum of two numbers. In Eq. 10, constants cancel each other and Eq. 11 replaces area and intersection for the $\mathrm{IoU}(B, \bar{B})$, which simplifies to the definition of $\mathrm{IoU}(B, \bar{B})$. □

2. Details of the Bounding Box Generator

In this section we present the derivation of the Equation 4 and 5, and explain the $\mathrm{findBRFeasibleSpace}(B, T, \mathrm{TL}(\bar{B}))$ function.

2.1. $\mathrm{findTLFeasibleSpace}(B, T)$ function

Here, we derive Equation 4 and 5 in the paper, and present the equations for the top-left space..

In order to derive Equation 4 depicting x^I_{max}, we bound the x coordinate first. It is obvious that $x^I_{min} = x_1$ due to the boundary of Region I. For x^I_{max}, we know that $\bar{y_1} = y_1$ again thanks to the region boundary. Therefore, since we have only one unknown, x^I_{max}, we use Eq. the definition of the IoU to determine its value in Eq. 13-18. Eq. 14 defines IoU based on Eq. 13. In Eq. 15, we set $\min(\bar{x_2}, x_2) = x_2$, $\max(\bar{x_1}, x_1) = x^I_{max}$, $\min(\bar{y_2}, y_2) = y_2$ and $\max(\bar{y_1}, y_1) = y_1$ by taking into the intersection definition in Region I. Also note that $\bar{x_1} = x^I_{max}$, $\bar{y_1} = y_1$, $\bar{x_2} = x_2$ and $\bar{y_2} = y_2$ in this case. In Eq. 16-18, we just rearrange the terms to have x^I_{max} as a left hand side term.

$$\mathrm{IoU}(B, \bar{B}) = \frac{I(B, \bar{B})}{A(B) + A(\bar{B}) - I(B, \bar{B})} \tag{13}$$

$$= \frac{(\min(\bar{x_2}, x_2) - \max(\bar{x_1}, x_1)) \times (\min(\bar{y_2}, y_2) - \max(\bar{y_1}, y_1))}{(x_2 - x_1) \times (y_2 - y_1) + (\bar{x_2} - \bar{x_1}) \times (\bar{y_2} - \bar{y_1}) - I(B, \bar{B})} \tag{14}$$

$$\Rightarrow T = \frac{(x_2 - x^I_{max}) \times (y_2 - y_1)}{(x_2 - x_1) \times (y_2 - y_1) + (x_2 - x^I_{max}) \times (y_2 - y_1) - (x_2 - x^I_{max}) \times (y_2 - y_1)} \tag{15}$$

$$\Rightarrow (x_2 - x_1) \times (y_2 - y_1) \times T = (x_2 - x^I_{max}) \times (y_2 - y_1) \tag{16}$$

$$\Rightarrow x^I_{max} = x_2 - \frac{(x_2 - x_1) \times (y_2 - y_1) \times T}{(y_2 - y_1)} \tag{17}$$

$$\Rightarrow x^I_{max} = x_2 - (x_2 - x_1) \times T \tag{18}$$

Now since we know the values of $\bar{x_1}$ based on the bounds, we can derive the Equation 5 (in the paper) for any $\bar{y_1}$ value in equations by moving within bounds. Since $I(B, \bar{B}) = (x_2 - \bar{x_1}) \times (y_2 - y_1)$, it does not rely on $\bar{y_1}$ and we directly use $I(B, \bar{B})$ in the following equations:

$$\mathrm{IoU}(B, \bar{B}) = \frac{I(B, \bar{B})}{A(B) + (x_2 - \bar{x_1}) \times (y_2 - \bar{y_1}) - I(B, \bar{B})} \tag{19}$$

$$\Rightarrow T \times (x_2 - \bar{x_1}) \times (y_2 - \bar{y_1})$$
$$= I(B, \bar{B}) + T \times I(B, \bar{B}) - T \times A(B) \tag{20}$$

$$\Rightarrow \bar{y_1} = y_2 - \frac{\frac{I(B, \bar{B})}{T} + I(B, \bar{B}) - A(B)}{(x_2 - \bar{x_1})} \tag{21}$$

Table 1: Top-Left space bounds and equations. See Fig. 4 in the paper.

Region	Min Bound	Max Bound	Equation
I	$\bar{x_1} = x_1$	$\bar{x_1} = x_2 - (x_2 - x_1) \times T$	$\bar{y_1} = y_2 - \frac{\frac{I(B,\bar{B})}{T} + I(B,\bar{B}) - A(B)}{(x_2 - \bar{x_1})}$
II	$\bar{y_1} = y_1$	$\bar{y_1} = y_2 - \frac{A(B) \times T}{x_2 - x_1}$	$\bar{x_1} = x_2 - \frac{I(B,\bar{B}) \times A(B)}{(y_2 - \bar{y_1})}$
III	$\bar{y_1} = y_1$	$\bar{y_1} = y_2 - \frac{A(B) \times T}{x_2 - x_1}$	$\bar{x_1} = x_2 - \frac{\frac{I(B,\bar{B})}{T} - A(B) + I(B,\bar{B})}{(y_2 - \bar{y_1})}$
IV	$\bar{y_1} = \frac{(y_2 \times (T-1)) + y_1}{T}$	$\bar{y_1} = y_1$	$\bar{x_1} = x_2 - \frac{A(B)}{T \times (y_2 - \bar{y_1})}$

Table 2: Bottom-Right space bounds.

Region	Min Bound	Max Bound
I	$\bar{y_2} = \frac{T \times A(B) + T \times (x_2 - \alpha) \times \beta + \beta \times (x_2 - \alpha) - T \times \bar{y_1} \times (x_2 - \bar{x_1})}{((T+1) \times (x_2 - \alpha) - T \times (x_2 - \bar{x_1}))}$	$\bar{y_2} = y_2$
II	$\bar{x_2} = x_2$	$\bar{x_2} = \bar{x_1} + \frac{\frac{I(B,\bar{B})}{T} - A(B) + I(B,\bar{B})}{(y_2 - \bar{y_1})}$
III	$\bar{y_2} = y_2$	$\bar{y_2} = \bar{y_1} + \frac{\frac{I(B,\bar{B})}{T} - A(B) + I(B,\bar{B})}{(x_2 - \bar{x_1})}$
IV	$\bar{x_2} = \frac{T \times A(B) + T \times (y_2 - \beta) \times \alpha + \alpha \times (y_2 - \beta) - T \times \bar{x_1} \times (y_2 - \bar{y_1})}{((T+1) \times (y_2 - \beta) - T \times (y_2 - \bar{y_1}))}$	$\bar{x_2} = x_2$

Table 1 presents all of the equations derived using the same methodology.

2.2. $\mathrm{findBRFeasibleSpace}(B, T, \mathrm{TL}(\bar{B}))$ Function

We follow the same approach for the bottom right corner with the top left corner. However, different from top-left space this step is required also consider the point generated top-left point. Note that the size of the polygon in the bottom-right space is affected by the distance between $\mathrm{TL}(\bar{B})$ and $\mathrm{TL}(B)$. Maximum bottom-right polygon size, with exactly the same size of the top-left polygon, is achieved when $\mathrm{TL}(\bar{B}) = \mathrm{TL}(B)$. Conversely, bottom-right polygon degenerates to a point at $\mathrm{BR}(B)$ if the sampled $\mathrm{TL}(\bar{B})$ hits the border of the top-left polygon.

We add two additional parameters for the sake of clarity: $\alpha = \max(\bar{x_1}, x_1)$, $\beta = \max(\bar{y_1}, y_1)$, $\hat{\alpha} = \min(\bar{x_2}, x_2)$ and $\hat{\beta} = \min(\bar{y_2}, y_2)$. The bounds and the equations are derived by the same methodology that is illustrated in the first step presented in Tables 2 and 3 respectively.

3. Implementation Details

3.1. Integrating pRoI Generator into the Training

The training of the two-stage object detectors involves 3 different networks as shown in Fig. 1. The first network is the feature extractor (i.e. ResNet[3]) which presents the base features to the second network, the proposal generator (i.e. RPN [4]), and the third network, which is the object detector (i.e. R-CNN [2], R-FCN [1]). The feature extractor is trained with the gradients back-propagated from the proposal generator and the object detector. The proposal generator is trained by a subset of the anchor-ground truth combinations (chosen by Sample Anchors to Train RPN in Figure 1) and a subset of these RPN proposals (i.e. RoIs) (chosen by Sample RoIs to Train R-CNN in Figure 1) are fed into the R-CNN after a series of operations including NMS and RoI Pooling that do not include learnable parameters. Finally, the loss is back-propagated through the entire network to update the parameters. However, the RoIs from the RPN is limited in number and diversity, which can impact the analysis and training. To address this, pRoI generator aims to generate RoIs with any desired property and in any number. Note that the gradients can also be back-propagated to the feature extractor as in the conventional training (i.e. RPN) since positive RoI Generator uses ground truths to generate an RoI in a similar manner to the conventional training, but differently it can generate boxes with the desired properties. During training, only for positive RoIs, pRoI Generator does not use the modules that are under the transparent red rectangle in Figure 1. However, during test time, our method follows the conventional approach, namely the RoIs from RPN are used due to the fact that no ground truth information is available during testing.

3.2. Connection Between $\mathrm{genRoIs}()$ and $\mathrm{generateBB}()$

As described in the text, $\mathrm{generateBB}()$ is a low-level function and any approach uses generated BBs approach is to rely on this function. That's why it is a subroutine of $\mathrm{genRoIs}()$. The main idea in our implementation

Table 3: Bottom-right space equations.

Region	Equation
I	$\bar{x_2} = \bar{x_1} + \frac{\frac{I(B,\bar{B})}{T} - A(B) + I(B,\bar{B})}{\bar{y_2} - \bar{y_1}}$
II	$\bar{y_2} = \bar{y_1} + \frac{\frac{I(B,\bar{B})}{T} - A(B) + I(B,\bar{B})}{\bar{x_2} - \bar{x_1}}$
III	$\bar{x_2} = \frac{T \times A(B) + \alpha \times T \times (\hat{\beta} - \beta) + \alpha \times (\hat{\beta} - \beta) - T \times \bar{x_1} \times (\bar{y_2} - \bar{y_1})}{(T+1) \times (\hat{\beta} - \beta) - T \times (\bar{y_2} - \bar{y_1}))}$
IV	$\bar{y_2} = \frac{T \times A(B) + \beta \times T \times (\hat{\alpha} - \alpha) + \beta \times (\hat{\alpha} - \alpha) - T \times \bar{y_1} \times (\bar{x_2} - \bar{x_1})}{(T+1) \times (\hat{\alpha} - \alpha) - T \times (\bar{x_2} - \bar{x_1}))}$

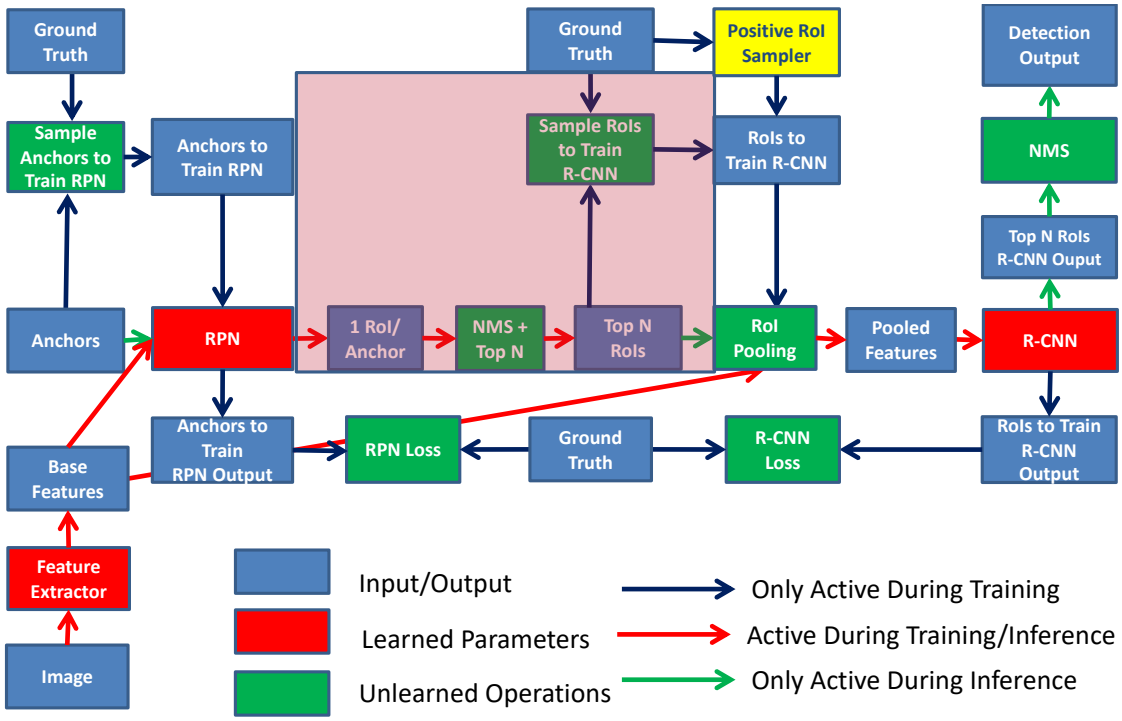

Figure 1: Conventional Faster R-CNN Training and our modification. During training, Positive RPN RoIs are not utilized and thus the modules presented under the large red rectangle are not used for positive RoIs. These RoIs are generated by the Positive RoI Generator shown in yellow box.

is to generate bounding boxes by iteratively calling the $\mathrm{generateBB}()$ for $RoINum$ times.

Apart from $RoINum$, the number of RoIs to be generated, there are two main input sets to the $\mathrm{genRoIs}$ function. Firstly, GTs and $perGTRoI$ together have the information about the box coordinates and the number of RoIs to be generated from each ground truth box. Therefore, for i^{th} ground truth box (GTs_i), we call $\mathrm{generateBB}()$ function for $perGTRoI_i$ times. And the second set of input comprises ψ_{IoU} and W_{IoU}, which together have information about the weights of each IoU interval. Therefore, for determining an IoU for each box, we first generate $perGTRoI$ number of samples of IoU intervals using the multinomial distribution defined in W_{IoU} and then, for each resulting interval, we again sample uniformly an IoU within its limits. These IoUs are clipped from 0.95 in order to prevent the problems arising from the precision problem in the $\mathrm{samplePolygon}()$ acceptance process. This sampling strategy distributes the input IoUs over an interval evenly. Finally, we randomly shuffle this set of IoUs and associate them to the ground truths, which completes the generation of the ground truth and desired IoU pairs as the input of the $\mathrm{generateBB}()$ function.

Table 4: The configurations of W_{IoU} for the different tables in the paper.

Table	RoI Source	$IoU = 0.5$	$IoU = 0.6$	$IoU = 0.7$	$IoU = 0.8$	$IoU = 0.9$
1	Right Skew	0.02	0.10	0.20	0.30	0.38
1	Balanced	0.33	0.17	0.18	0.17	0.15
1	Left Skew	0.73	0.12	0.15	0.05	0
4	Balanced, IoU=0.5	0.33	0.17	0.18	0.17	0.15
4	Balanced, IoU=0.6	0	0.38	0.20	0.22	0.20
4	Balanced, IoU=0.7	0	0	0.48	0.25	0.27
4	Balanced, IoU=0.8	0	0	0	0.64	0.36
4	Balanced, IoU=0.9	0	0	0	0	1

3.3. Configurations of W_{IoU}

The configurations of the W_{IoU} (i.e. the distribution over $\psi_{IoU} = [0.5, 0.6, 0.7, 0.8, 0.9]$) used for the experiments is shown in Table 4.

References

[1] 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)*, 2016.

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

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] S. Ren, K. He, R. 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.