

# Improving Object Localization with Fitness NMS and Bounded IoU Loss

Lachlan Tychsen-Smith, Lars Petersson  
CSIRO (Data61)

CSIRO-Synergy Building, Acton, ACT, 2601

Lachlan.Tychsen-Smith@data61.csiro.au, Lars.Petersson@data61.csiro.au

## Abstract

We demonstrate that many detection methods are designed to identify only a sufficiently accurate bounding box, rather than the best available one. To address this issue we propose a simple and fast modification to the existing methods called *Fitness NMS*. This method is tested with the *DeNet* model and obtains a significantly improved MAP at greater localization accuracies without a loss in evaluation rate, and can be used in conjunction with *Soft NMS* for additional improvements. Next we derive a novel bounding box regression loss based on a set of IoU upper bounds that better matches the goal of IoU maximization while still providing good convergence properties. Following these novelties we investigate RoI clustering schemes for improving evaluation rates for the *DeNet* wide model variants and provide an analysis of localization performance at various input image dimensions. We obtain a MAP of 33.6%@79Hz and 41.8%@5Hz for MSCOCO and a Titan X (Maxwell). Source code available from: <https://github.com/lachlants/denet>

## 1. Introduction

Multiclass object detection is defined as the joint task of localizing bounding boxes of instances and classifying their contents. In this paper, we address the problem of bounding box localization which has become increasingly important as the number of objects in view rises. In particular, by identifying a better fitting bounding box, current methods can better resolve the position, scale and number of unique instances in view. Modern object detection methods most often utilize a CNN [11] based bounding box regression and classification stage followed by a Non-Max Suppression method to identify unique object instances.

Introduced in R-CNN [7], bounding box regression enables each region of interest (RoI) to estimate an updated bounding box with the goal of better matching the nearest true instance. Prior work has demonstrated that this task can be improved with multiple bounding box regression

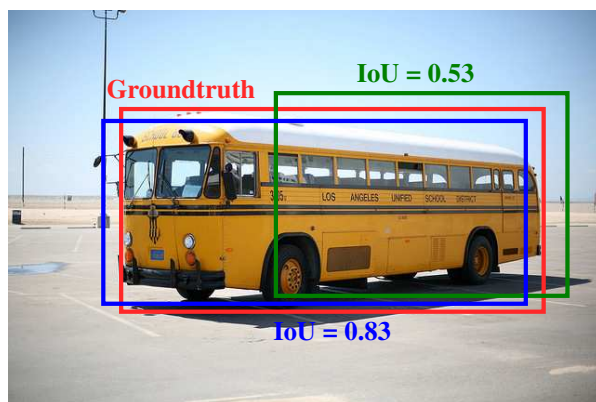


Figure 1: MSCOCO [15] sample depicting the groundtruth bounding box (red) and two predicted bounding boxes (green and blue). Both predicted bounding boxes have a sufficient overlap with the groundtruth, i.e.,  $\text{IoU} > 0.5$ . Standard detection designs do not directly discriminate between the predicted boxes, whereas *Fitness NMS* explicitly directs the model to select the blue box.

stages [5], increasing the number of (or carefully selecting) RoI anchors in the region proposal network [16] [14], and increasing the input image resolution (or using image pyramids) [13] [14]. Alternatively, the *DeNet* [19] method demonstrated enhanced precision by improving the localization of the sampling RoIs before bounding box regression. This was achieved via a novel corner-based RoI estimation method, replacing the typical region proposal network (RPN) [18] used in many other methods. This approach operates in real-time, demonstrated improved fine object localization and has the additional benefit of not requiring user-defined anchor bounding boxes. Despite only demonstrating our results with *DeNet*, we believe the novelties presented are equally applicable to other detectors.

Despite its simplicity, Non-Max Suppression (see Algorithm 1) provides highly efficient detection clustering with the goal of identifying unique bounding box instances. Two functions `same(.)` and `score(.)`, and the user defined con-

---

**Algorithm 1** Non-Max Suppression

---

```
1: procedure NMS( $B, c$ )
2:    $B_{nms} \leftarrow \emptyset$ 
3:   for  $b_i \in B$  do
4:      $discard \leftarrow \text{False}$ 
5:     for  $b_j \in B$  do
6:       if  $\text{same}(b_i, b_j) > \lambda_{nms}$  then
7:         if  $\text{score}(c, b_j) > \text{score}(c, b_i)$  then
8:            $discard \leftarrow \text{True}$ 
9:       if not  $discard$  then
10:         $B_{nms} \leftarrow B_{nms} \cup b_i$ 
11:   return  $B_{nms}$ 
```

---

stant  $\lambda_{nms}$  define the behaviour of this method. Typically,  $\text{same}(b_i, b_j) = \text{IoU}(b_i, b_j)$  and  $\text{score}(c, b_i) = \text{Pr}(c|b_i)$  where  $b_i, b_j$  are bounding boxes and  $c$  is the corresponding class. Conceptually,  $\text{same}(\cdot)$  uses the intersection-over-union to test whether two bounding boxes are best associated with the same true instance. While the  $\text{score}(\cdot)$  function is used to select which bounding box should be kept, in this case, the one with the greatest confidence in the current class. Recent work has proposed decaying the associated detection *score* instead of discarding detections [1] and learning the NMS function directly from data [8] [9].

In contrast, we investigate modifications to the  $\text{score}(\cdot)$  and  $\text{same}(\cdot)$  functions without changing the overall NMS algorithm. With *Fitness NMS* we modify the  $\text{score}(\cdot)$  function to better select bounding boxes which maximise their estimated IoU with the groundtruth, and with *Bounded IoU Loss* we modify the training of the inputs to  $\text{same}(\cdot)$  to provide tighter clusters. Following these novelties we provide an analysis of RoI clustering and input image sizes.

## 2. Experimental Setup

### 2.1. Operating Parameters and Definitions

Here we introduce some important parameters and definitions used throughout the paper:

- **Sampling Region-of-Interest (RoI)**: a bounding box generated before classification. Only applicable to two-stage detectors (e.g., R-CNN variants, etc).
- **Intersection-over-Union (IoU)**: the intersection area divided by the union area for two bounding boxes.
- **Matching IoU ( $\Omega$ )**: indicates the necessary IoU overlap between a model generated bounding box and a groundtruth bounding box before they are considered the same detection. A higher matching IoU indicates increased requirements for fine object localization. Typically, there are two matching IoUs, one

used during training ( $\Omega_{\text{train}}$ ) and one used for testing ( $\Omega_{\text{test}}$ ). The training matching IoU ( $\Omega_{\text{train}}$ ) is used to construct the target probability distribution for sampling RoIs, e.g., in DeNet given a ground-truth instance  $(c_t, b_t)$  if  $\text{IoU}(b_s, b_t) \geq \Omega_{\text{train}}$  then  $\text{Pr}(c_t|b_s) = 1$ . For Pascal VOC [3]  $\Omega_{\text{test}} = 0.5$  while MSCOCO [15] considers a range  $\Omega_{\text{test}} \in [0.5, 0.95]$ .

- **Clustering IoU ( $\lambda_{nms}$ )**: indicates the IoU required between two detections before one is removed via Non Max Suppression, e.g., if  $\text{IoU}(b_0, b_1) > \lambda_{nms}$  and  $\text{Pr}(c|b_0) > \text{Pr}(c|b_1)$  then the detector *hit*  $(c, b_1)$  is removed. In the DeNet model, the default value is  $\lambda_{nms} = 0.5$ , however, we vary this factor in some figures as indicated.

### 2.2. Datasets, Training and Testing

For validation experiments, we combine Pascal VOC [3] 2007 `trainval` and VOC 2012 `trainval` datasets to form the training data and test on Pascal VOC 2007 `test`. For testing, we train on MSCOCO [15] `trainval` and use the `test-dev` dataset for evaluation. Following DeNet [19] and SSD [16], all timing results are provided for an Nvidia Titan X (Maxwell) GPU with cuDNN v5.1 and a batch size of 8. Furthermore, unless stated otherwise, the same learning schedule, hyper-parameters, augmentation methods, etc were used as in the DeNet paper [19].

## 3. Fitness Non-Max Suppression

In this section, we highlight a flaw in the Non-Max Suppression based instance estimation, namely, the reliance on a single matching IoU. Following this analysis, we propose and demonstrate a novel *Fitness NMS* algorithm.

### 3.1. Matching IoUs

In Table 1 and 2 we provide the MAP for the DeNet-34 (skip) [19] model trained with a variety of matching IoUs (keeping all other variables constant). We observe that increasing the train matching IoU can be used to significantly improve MAP at greater test matching IoUs at the expense of lower matching IoUs. In particular, we note an improvement of 1.2% MAP@[0.5:0.95] for MSCOCO by selecting a training matching IoU near the center of the test matching IoU range. These results demonstrate a problem in many detection methods, i.e., they only operate optimally at a single matching IoU, when  $\Omega_{\text{train}} = \Omega_{\text{test}}$ . This property poses a problem when we want the detector to provide the *best* available bounding box rather than just a *sufficient* one, i.e., we wish to select the bounding box which maximizes IoU rather than just satisfying  $\text{IoU} > \Omega_{\text{test}}$ . In the following, we develop the hypothesis that this problem stems from the applied instance estimation method, i.e., Non-Max Suppression acting on class probabilities.

$\Omega_{\text{train}}$	MAP@ $\Omega_{\text{test}}$ (%)				
	0.5	0.6	0.7	0.8	0.9
0.5	<b>75.0</b>	68.9	58.2	39.5	12.0
0.7	73.7	<b>69.0</b>	<b>60.6</b>	<b>45.8</b>	17.2
0.9	62.9	59.0	52.8	42.8	<b>26.4</b>

Table 1: VOC 2007 *test* results for DeNet-34 model with a variable matching IoU. The model only performs optimally when both training and testing IoUs are equal.

$\Omega_{\text{train}}$	MAP@ $\Omega_{\text{test}}$ (%)			MAP@Area (%)		
	0.5:0.95	0.5	0.75	S	M	L
0.5	29.5	<b>47.9</b>	31.1	<b>8.8</b>	30.9	47.0
0.7	<b>30.7</b>	45.9	<b>33.5</b>	7.9	<b>33.6</b>	<b>50.9</b>

Table 2: Comparing MSCOCO *test-dev* results for DeNet-34 model with a training matching IoU of 0.5 and 0.7. Training with an IoU near the center of the testing IoU range improves MAP[0.5:0.95] by +1.2%.

### 3.2. Detection Clustering

To indicate the *score* of a bounding box  $b_j$ , many detection models (including DeNet [19]) apply:

$$\text{score}(c, b_j) = \Pr(c|b_j) \quad (1)$$

where  $\Pr(c|b_j)$  provides the probability that instance  $(c, b_j)$  is within the groundtruth assignment for the image. In practice, the distribution  $\Pr(c|b_j)$  is trained to indicate whether a groundtruth instance  $(c, b_t)$  has an overlap satisfying  $\text{IoU}(b_j, b_t) > \Omega_{\text{train}}$ . Following class estimation, the Non-Max Suppression (NMS) algorithm is applied to  $\text{score}(c, b)$  over the bounding boxes  $b$  to identify the most likely set of unique instance candidates (see Algorithm 1). In a typical production scenario, the clustering IoU  $\lambda_{\text{nms}}$  is manually optimized to obtain the maximum recall within a predefined number of output detections over a validation set. In this context, the goal of the NMS algorithm is to simultaneously reduce the number of output detections (via deduplication) while maximizing the recall at the desired matching IoU (singular or range).

In Figure 2, we demonstrate the performance of the standard NMS algorithm with the DeNet-34 (skip) model at varying clustering IoUs ( $\lambda_{\text{nms}}$ ) and matching IoUs ( $\Omega_{\text{test}}$ ). Note that the *Without NMS* line indicates an upper bound on recall and has an order of magnitude more detections, i.e., approx. 2.5M. We observe that this method performs well when  $\Omega_{\text{train}} = \Omega_{\text{test}} = 0.5$ , quickly converging to the *Without NMS* line as detections are added, e.g., there is a delta of approximately 7% recall between the *Without NMS* line and 32K detections. However, with greater localization accuracies  $\Omega_{\text{test}} \gg 0.5$  the performance worsens and

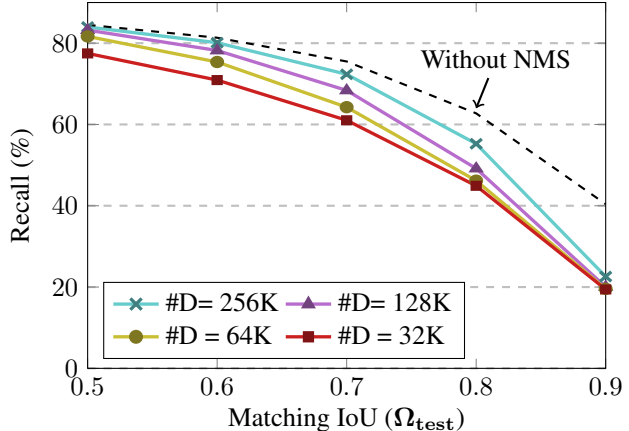


Figure 2: VOC 2007 *test* recall for DeNet-34 model with  $\Omega_{\text{train}} = 0.5$ . The clustering IoU ( $\lambda_{\text{nms}}$ ) was varied to obtain the desired number of output detections ( $\#D$ ). Note the large delta between the *Without NMS* and other lines at high matching IoUs (e.g.,  $\Omega_{\text{test}} = 0.9$ ).

converges slowly as detections are added. In particular, we note the recall with  $\Omega_{\text{test}} = 0.9$  dropped from near 40% to less than 20% as detections were culled via the NMS algorithm. The large delta between the *Without NMS* line and other results at high matching IoUs indicates that the correct bounding boxes are being identified during ROI prediction, however, the NMS algorithm is quickly discarding them.

These results suggest that naively applying NMS with current detection models is suboptimal for high localization accuracies. We propose that this phenomenon *partially* stems from the fact that the DeNet model is trained with a single matching IoU, i.e., during training  $\Pr(c = c_t|b_s) = 1$  if the sampling bounding box  $b_s$  overlaps a groundtruth instance  $(c_t, b_t)$  with  $\text{IoU}(b_s, b_t) \geq 0.5$ . This makes the class distribution a poor discriminator between bounding boxes of better or worse IoU since it is trained to equal one whether the IoU is 0.5 or 0.9, etc.

### 3.3. Novel Fitness NMS Method

To address the issues described in the previous sections we propose augmenting Equation 1 with an additional expected *fitness* term:

$$\text{score}(c, b_j) = \Pr(c|b_j)E[f_j|c] \quad (2)$$

where  $f_j$ , referred to as *fitness* in this paper, is a discrete random variable indicating the IoU overlap between the bounding box  $b_j$  and any overlapping groundtruth instance, and  $E[f_j|c]$  is the expected value of  $f_j$  given the class  $c$ . With this formulation, bounding boxes with both greater estimated IoU overlap and class probability will be favored in the clustering algorithm.

In our implementation, the *fitness*  $f_j$  can take on  $F$  values ( $F = 5$  in this paper) and is mapped via:

$$\rho_j = \max_{b \in B_T} \text{IoU}(b_j, b) \quad (3)$$

$$f_j = \lfloor F(2\rho_j - 1) \rfloor \quad (4)$$

where  $0 \leq \rho_j \leq 1$  is the maximum IoU overlap between  $b_j$  and the set of groundtruth bounding boxes  $B_T$ . If  $\rho_j$  is less than 0.5, the bounding box  $b_j$  is assigned the *null* class with no associated fitness. From these definitions the expected value is given by:

$$E[f_j|c] = \sum_{n=0}^{n < F} \lambda_n \Pr(f_j = n|c) \quad (5)$$

where  $\lambda_n = \frac{1}{2} + \frac{n}{2F}$  is the minimum IoU overlap associated with a fitness of  $f_j = n$ . In estimating  $\Pr(f_j|c)$  from supervised training data we investigated two variants:

- (a) **Independent Fitness ( $\mathbf{I}_F$ ):** Model  $\Pr(c|b_j)$  and  $\Pr(f_j|c)$  independently and assume fitness is independent of class, i.e.,  $\Pr(f_j|c) = \Pr(f_j)$ . For training, we replaced the final layer of a pretrained DeNet model with one which estimates  $\Pr(f_j|b_j)$  and  $\Pr(c|b_j)$ , both of which are jointly optimized via separate cross entropy losses. The method required one additional hyperparameter, the cost weighting for the fitness probability which was set to 0.1. With this method, the model produces  $|C| + |F| + 2$  outputs per RoI.
- (b) **Joint Fitness ( $\mathbf{J}_F$ ):** Estimate fitness and class jointly with  $\Pr(c, f_j|b_j)$ , and apply the following identity:

$$\text{score}(c, b_j) = \sum_{n=0}^{n < F} \lambda_n \Pr(c, f_j = n|b_j) \quad (6)$$

For training, we replaced the final layer of a pretrained DeNet model with one which jointly estimates  $\Pr(c, f_j|b_j)$  instead of  $\Pr(c|b_j)$ . Note the softmax is normalised over both class and fitness. The training method required no additional hyperparameters and the same cost weighting was used as in the original layer. Since the *null* class has no associated fitness, this method requires  $|C||F| + 1$  model outputs per RoI.

In Table 3, we compare the MAP of a DeNet-34 (skip) model on Pascal VOC 2007 *test* augmented with the novel Fitness NMS methods over a range of matching IoUs ( $\Omega_{\text{test}}$ ). The Baseline model was retrained starting from an already trained model such that all variants experienced the same number of training epochs.

The Joint Fitness NMS method demonstrates a clear lead at high matching IoUs with no observable loss for low matching IoUs. In Figure 3, we provide the recall for the

NMS Method	MAP@ $\Omega_{\text{test}}$ (%)				
	0.5	0.6	0.7	0.8	0.9
Baseline	75.3	70.4	61.0	45.3	18.4
Ind. Fitness	75.4	70.2	63.2	48.2	25.3
Joint Fitness	<b>75.5</b>	<b>70.8</b>	<b>64.0</b>	<b>49.7</b>	<b>27.9</b>

Table 3: Pascal VOC 2007 *test* results for DeNet-34 (skip) with varying NMS methods and  $\Omega_{\text{test}}$ . Baseline is the original model and NMS method. Fitness NMS methods improves MAP at high IoU ( $\Omega_{\text{test}}$ ).

Model	Eval. Rate	MAP@ $\Omega_{\text{test}}$ (%)		
		0.5:0.95	0.5	0.75
DeNet-34 <b>S</b>	82 Hz	29.5	<b>47.9</b>	31.1
DeNet-34 <b>S+I<sub>F</sub></b>	81 Hz	31.1	47.7	33.1
DeNet-34 <b>S+J<sub>F</sub></b>	78 Hz	<b>31.4</b>	<b>47.9</b>	<b>33.5</b>
DeNet-34 <b>W</b>	44 Hz	30.0	48.9	31.8
DeNet-34 <b>W+I<sub>F</sub></b>	44 Hz	32.1	48.5	34.2
DeNet-34 <b>W+J<sub>F</sub></b>	44 Hz	<b>32.6</b>	<b>49.2</b>	<b>34.9</b>
DeNet-101 <b>W</b>	17 Hz	33.8	53.4	36.1
DeNet-101 <b>W+I<sub>F</sub></b>	17 Hz	35.2	52.9	37.7
DeNet-101 <b>W+J<sub>F</sub></b>	17 Hz	<b>36.5</b>	<b>53.9</b>	<b>39.1</b>

Table 4: MSCOCO *test-dev* results with varying NMS methods. Models are designated **S** for the *skip* variant and **W** for the *wide* variant, and **I<sub>F</sub>** utilize the *independent* variant and **J<sub>F</sub>** the *joint* variant of Fitness NMS. Our NMS methods improve MAP when  $\Omega_{\text{test}} = 0.75$ .

Joint Fitness NMS method and the recall delta between Joint Fitness NMS and Baseline for the same model. These results directly demonstrate the improved recall at various operating points obtained by the Fitness NMS methods at fine localization accuracies with negligible losses for coarse localization.

In Table 4, we provide MSCOCO *test-dev* results for DeNet-34 (skip), DeNet-34 (wide) and DeNet-101 (wide) models augmented with the Fitness NMS methods. The Independent Fitness method consistently demonstrated a marginal loss in precision at  $\Omega_{\text{test}} = 0.5$  with a significant improvement at  $\Omega_{\text{test}} = 0.75$ , however, the Joint Fitness NMS method demonstrated no measurable loss at  $\Omega_{\text{test}} = 0.5$  and provided a greater improvement for  $\Omega_{\text{test}} = 0.75$ . In particular, these methods provide a significant improvement for large objects which, most likely, have many associated sampling RoIs. The DeNet-101 model produced a large improvement moving from the *independent* to *joint* variant, suggesting that the increased modelling *capacity* it provides is better utilized with the *joint* method. Overall, both Fitness NMS methods produced a significant improvement for MAP@[0.5:0.95] between 1.9-2.7%.

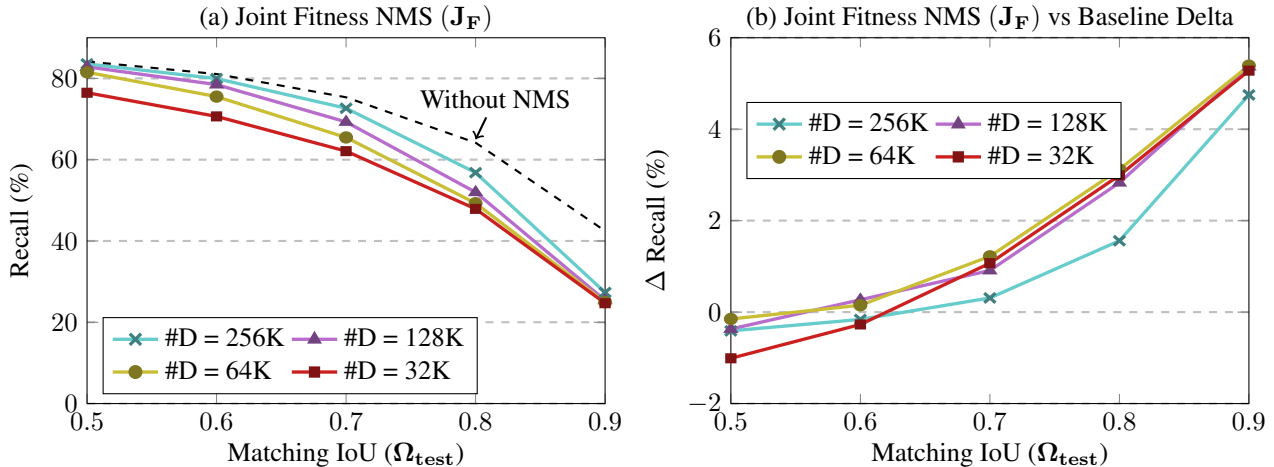


Figure 3: VOC 2007  $t_{est}$  recall with DeNet-34: (a) with Joint Fitness NMS, and (b) delta obtain by switching from Joint Fitness NMS to standard NMS. We varied the clustering IoU to obtain the desired number of detections (#D). Improved recall with IoU > 0.6 was observed with Joint Fitness NMS enabled, peaking at approx. +5% as  $\Omega_{test} = 0.9$ .

Model	Eval. Rate	MAP@ $\Omega_{test}$ (%)		
		0.5:0.95	0.5	0.75
DeNet-34 <b>S</b>	80 Hz	30.4	<b>48.4</b>	32.6
DeNet-34 <b>S+J<sub>F</sub></b>	81 Hz	<b>32.2</b>	48.1	<b>34.8</b>
DeNet-34 <b>W</b>	43 Hz	31.0	49.4	33.2
DeNet-34 <b>W+J<sub>F</sub></b>	41 Hz	<b>33.4</b>	<b>49.7</b>	<b>36.0</b>
DeNet-101 <b>W</b>	17 Hz	35.1	50.9	37.8
DeNet-101 <b>W+J<sub>F</sub></b>	15 Hz	<b>37.2</b>	<b>54.2</b>	<b>40.2</b>

Table 5: Applying Gaussian variant ( $\sigma = 0.5$ ) of Soft NMS [1] method to models in Table 4. MAP improved by 0.7-1.3% for both standard and Joint Fitness NMS models.

A minor loss in evaluation rate was observed for Joint Fitness NMS due to the increased number of values in the final softmax (401 vs 81). Applying the *joint* variant to a single-stage detector [16] [14] would likely result in a greater evaluation rate penalty due to much greater number of softmax’s required in current designs (one for each candidate RoI). In these applications, the *independent* variant might be a better candidate.

### 3.3.1 Applying Soft-NMS

In Table 5 we demonstrate that the Fitness NMS methods can be used in parallel with the Soft NMS method [1]. Our results indicate an improvement between 0.7 and 1.3% MAP for models utilizing both the standard and Fitness NMS methods. This highlights that the Soft NMS and Fitness NMS methods address tangential problems. Note that subsequent models in the paper do **not** use Soft NMS.

## 4. Bounding Box Regression

In this section, we discuss the role of bounding box regression in detection models. We then derive a novel bounding box regression loss based on a set of upper bounds which better matches the goal of maximising IoU while maintaining good convergence properties (smoothness, robustness, etc) for gradient descent optimizers.

### 4.1. Same-Instance Classification

Since its introduction in R-CNN [7], bounding box regression has provided improved localization for many two-stage [18] [13] [2] [19] and single-stage detectors [14] [16] [17]. With bounding box regression each RoI is able to identify an updated bounding box which better matches the nearest ground-truth instance. Due to the improved accuracy of the sampling RoIs in DeNet compared to R-CNN derived methods, it is natural to assume a reduced need for bounding box regression. However, we demonstrate the contrary in Figure 4, which provides the change in recall obtained with the model in Figure 2 with and without bounding box regression enabled. We observe improved recall with bounding box regression enabled for most data points except notably the *Without NMS* and 256K lines at high matching IoUs. These results indicate that the RoIs have improved recall, however, as NMS is applied (other lines) the regressed bounding boxes perform better. Furthermore, these results demonstrate that bounding box regression is still important in the DeNet model under typical operating conditions (red and yellow lines).

We propose that the improved recall obtained by the bounding box regression enabled DeNet model is **not** primarily due to improving the position and dimensions of in-



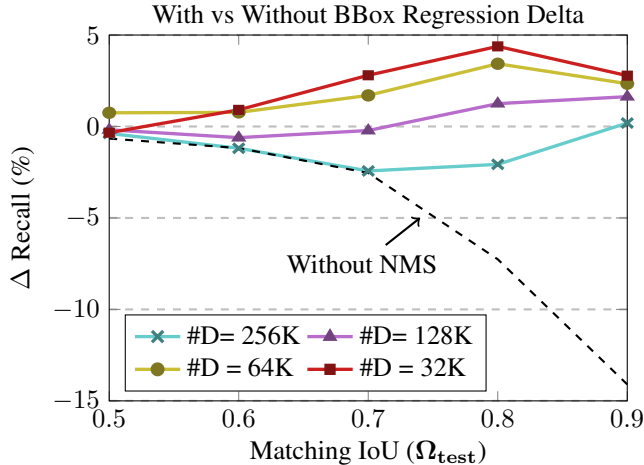


Figure 4: VOC 2007  $\text{test}$  recall delta for DeNet-34 (skip) with and without bounding box regression. We varied the clustering IoU ( $\lambda_{\text{nms}}$ ) to obtain the desired number of detections ( $\#D$ ). Bounding box regression provides improved recall for high matching IoUs and low detection numbers.

dividual bounding boxes, but instead due to its interaction with the NMS algorithm. That is, to test if two RoIs ( $b_0, b_1$ ) are best associated with the same instance, we check if  $\text{IoU}(\beta_0, \beta_1) > \lambda_{\text{nms}}$  where  $(\beta_0, \beta_1)$  are the associated updated bounding boxes (after bounding box regression). Therefore, if  $b_0$  and  $b_1$  share the same nearest groundtruth instance they are significantly more likely to satisfy the comparison with bounding box regression enabled. As a result, bounding box regression is particularly important for estimating unique instances in cluttered scenes where sampling RoIs can easily overlap multiple true instances.

## 4.2. Bounded IoU Loss

Here we propose a novel bounding box loss and compare it to the R-CNN method used so far. This new loss aims to maximise the IoU overlap between the RoI and the associated groundtruth bounding box, while providing good convergence properties for gradient descent optimizers.

Given a sampling RoI  $b_s = (x_s, y_s, w_s, h_s)$ , an associated groundtruth target  $b_t = (x_t, y_t, w_t, h_t)$ , and an estimated bounding box  $\beta = (x, y, w, h)$  the widely used R-CNN formulation [6] provides the following cost functions:

$$\text{Cost}_x = L_1\left(\frac{\Delta x}{w_s}\right) \quad (7)$$

$$\text{Cost}_w = L_1\left(\ln\left(\frac{w}{w_t}\right)\right) \quad (8)$$

where  $\Delta x = x - x_t$  and  $L_1(z)$  is the Huber Loss [10] (also known as smooth L1 loss [6]). Note that we restrict this analysis to the X position and width for the sake of brevity,

the Y position and height equations can be identified with suitable substitutions. The Huber loss is defined by:

$$L_\tau(z) = \begin{cases} \frac{1}{2}z^2 & |z| < \tau \\ \tau|z| - \frac{1}{2}\tau^2 & \text{otherwise} \end{cases} \quad (9)$$

Ideally, we would train the bounding box regression to directly minimize the IoU, e.g.,  $\text{Cost} = L_1(1 - \text{IoU}(b, b_t))$ , however, CNN's struggle to minimize this loss under gradient descent due to the high non-linearity, multiple degrees of freedom and existence of multiple regions of zero gradient of the IoU function. As an alternative, we consider the set of cost functions defined by:

$$\text{Cost}_i = 2L_1(1 - \text{IoU}_B(i, b_t)) \quad (10)$$

where  $\text{IoU}(b, b_t) \leq \text{IoU}_B(i, b_t)$  is an upper bound of the IoU function with the free parameter  $i \in \{x, y, w, h\}$ . We construct the upper bound by considering the maximum IoU attainable with the unconstrained free parameter, e.g.,  $\text{IoU}_B(x, b_t)$  provides the IoU as a function of  $x$  when  $y = y_t$ ,  $w = w_t$  and  $h = h_t$ . We obtain the following bounds on the IoU function:

$$\text{IoU}_B(x, b_t) = \max\left(0, \frac{w_t - 2|\Delta x|}{w_t + 2|\Delta x|}\right) \quad (11)$$

$$\text{IoU}_B(w, b_t) = \min\left(\frac{w}{w_t}, \frac{w_t}{w}\right) \quad (12)$$

In Figure 5, we compare the original R-CNN cost functions and the proposed Bounded IoU Loss. Since bounding box regression is only applied to boxes which have  $\text{IoU}(b_s, b_t) \geq 0.5$ , the following constraints apply:

$$\frac{|x_s - x_t|}{w_t} \leq \frac{1}{6} \quad (13)$$

$$\frac{1}{2} \leq \frac{w_s}{w_t} \leq 2 \quad (14)$$

Note that these constraints can be obtained by applying the position and scale upper bounds derived above and the identity  $\text{IoU}_B \geq \text{IoU} \geq 0.5 \Rightarrow \text{IoU}_B \geq 0.5$ . Assuming that the estimated bounding box  $\beta$  has a better IoU than the sampling bounding box  $b_s$ , we provide these bounds in Figure 5. We observe that the proposed cost for bounding box width has a nearly identical shape to the one used in the original algorithm, however, the proposed positional cost has a different shape and is considerably larger across the operating range. Furthermore, the positional cost is robust in the sense that as  $|\Delta x| \rightarrow \infty$  the cost approaches a constant rather than diverging like the original cost, this may prove advantageous for applications which have large positional deltas. This analysis suggests that with the aim of maximizing IoU, a greater emphasis should be placed on position than the original R-CNN cost implies. Though not

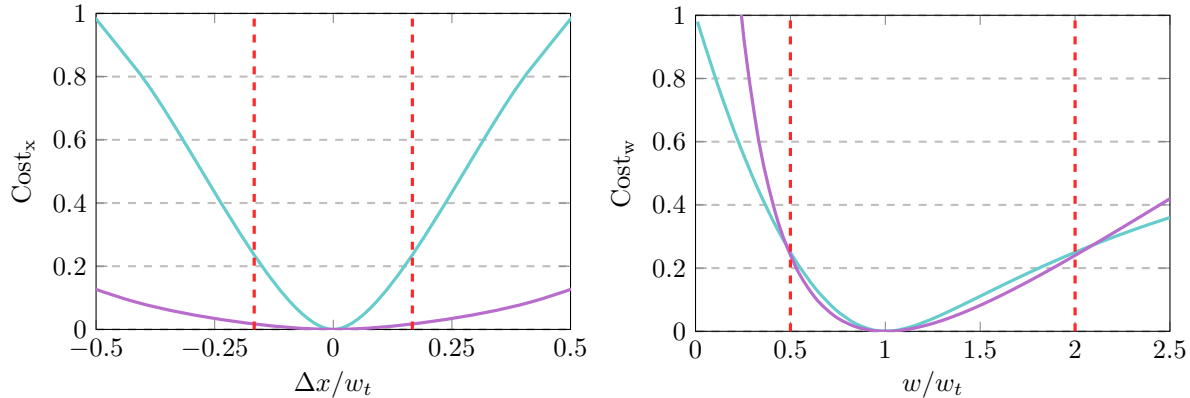


Figure 5: Comparing R-CNN bounding box regression loss (purple) and proposed Bounded IoU Loss (blue). Red dotted lines indicate the typical operating range given  $\text{IoU} \geq 0.5$ . Bounded IoU Loss places a much greater emphasis on position.

Model	MAP@ $\Omega_{\text{test}}$ (%)		
	0.5:0.95	0.5	0.75
DeNet-34 <b>S+J<sub>F</sub></b>	31.4	47.9	33.5
DeNet-34 <b>S+J<sub>F</sub>+B<sub>IoU</sub></b>	<b>32.1</b>	<b>48.6</b>	<b>34.3</b>
DeNet-34 <b>W+J<sub>F</sub></b>	32.6	49.2	34.9
DeNet-34 <b>W+J<sub>F</sub>+B<sub>IoU</sub></b>	<b>33.6</b>	<b>50.2</b>	<b>36.0</b>
DeNet-101 <b>W+J<sub>F</sub></b>	36.5	53.9	39.1
DeNet-101 <b>W+J<sub>F</sub>+B<sub>IoU</sub></b>	<b>37.0</b>	<b>54.3</b>	<b>39.5</b>

Table 6: MSCOCO  $\text{test-dev}$  results with novel Bounded IoU Loss ( $\mathbf{B}_{\text{IoU}}$ ) for bounding box regression. Our novel loss consistently improved MAP across all categories.

tested in this paper, we note that the R-CNN positional cost can be changed to  $Cost_x = 15L_{0.16}(\Delta x/w_s)$  to obtain a very similar shape over the defined operating range.

In Table 6, we demonstrate the performance of the DeNet models with the Bounded IoU Loss ( $\mathbf{B}_{\text{IoU}}$ ). The model training was identical to the Joint Fitness models except with the bounding box regression cost factor changed from 1.0 to 0.125. This value was determined by matching the magnitude of the bounding box regression training loss after a few iterations to the value obtained previously (no parameter search was used). We observe a consistent improvement across varying matching thresholds ( $\Omega_{\text{test}}$ ) for all models, particularly the DeNet-34 variants.

## 5. RoI Clustering

The *wide* variant DeNet models generate a corner distribution with a  $128 \times 128$  spatial resolution producing 67M candidate bounding boxes [19]. To accommodate the increased numbers of candidates the original DeNet model increased the number of RoIs from 576 to 2304, resulting in a significant detrimental effect on the evaluation rate, e.g.,

Model	Eval. Rate	MAP@ $\Omega_{\text{test}}$ (%)		
		0.5:0.95	0.5	0.75
DeNet-34 <b>W</b>	71 Hz	33.1	48.7	35.6
DeNet-34 <b>W+C</b>	79 Hz	33.6	49.8	<b>36.0</b>
DeNet-34 <b>W+N</b>	59 Hz	<b>33.7</b>	<b>50.7</b>	35.8
DeNet-101 <b>W</b>	22 Hz	36.3	52.8	39.0
DeNet-101 <b>W+C</b>	24 Hz	<b>36.9</b>	53.9	<b>39.5</b>
DeNet-101 <b>W+N</b>	21 Hz	<b>36.9</b>	<b>54.6</b>	39.3

Table 7: MSCOCO  $\text{test-dev}$  results comparing clustering methods; **C** indicates corner clustering with  $m = 1$ , **N** indicates standard NMS clustering with 0.7 threshold. Models were tested with 576 RoIs vs 2304 in other results. Both clustering methods improved MAP.

DeNet-34 (wide) observed a near 50% reduction in evaluation rate over the original model. To ameliorate this issue, we investigated two simple and fast clustering methods, one applying NMS on the corner distribution and the other applying NMS to the output bounding boxes. In the corner NMS method, when a corner at position  $(x, y)$  of type  $k$  is identified, i.e.,  $\Pr(s|k, y, x) > \lambda_C$ , the algorithm checks if it has the maximum probability in a local  $(2m + 1) \times (2m + 1)$  region.

In Table 7, we compare the corner clustering method, standard NMS method (with a 0.7 threshold), and no clustering when the number of RoIs is reduced to 576 for the DeNet wide models. The results demonstrate a 30% to 80% improvement in evaluation rate due to the decreased number of RoI classifications needed. We found both clustering methods improved upon no clustering and obtained very similar MAP@[0.5:0.95] results, however, the standard NMS method was slightly slower due to an increased CPU load. Relative to the NMS method, the corner cluster-

Model	Mean Input Pixels	Eval. Rate	MAP@ $\Omega_{\text{test}}$ (%)			MAP@Area (%)		
			0.5:0.95	0.5	0.75	S	M	L
FPN [13]	0.91 M	-	36.2	59.1	39.0	18.2	39.0	48.2
D-RFCN [12]	0.51 M	-	36.1	56.7	-	14.8	39.8	52.2
Soft NMS [1]	0.91 M	-	38.4	60.1	-	18.5	41.6	52.5
RetinaNet [14]	0.91 M	5 Hz	39.1	59.1	42.3	21.8	42.7	50.2
<b>Ours</b> ( $\times 384$ )	0.11 M	37 Hz	32.3	47.9	34.5	8.7	33.7	54.5
<b>Ours</b> ( $\times 512$ )	0.19 M	24 Hz	36.9	53.9	39.5	13.8	39.3	56.5
<b>Ours</b> ( $\times 768$ )	0.42 M	11 Hz	39.5	58.0	42.6	18.9	43.5	54.1
<b>Ours</b> ( $\times 1024$ )	0.75 M	6 Hz	38.8	58.3	41.8	21.3	43.4	49.4
<b>Ours</b> ( $\times 1280$ )	1.18 M	4 Hz	36.3	56.2	39.0	<b>22.1</b>	42.1	42.3
<b>Ours</b> ( $\times 1536$ )	1.69 M	3 Hz	32.9	52.5	35.0	22.0	39.9	34.4
<b>Ours</b> ( $\times 512, \times 1024$ )	0.94 M	5 Hz	<b>41.8</b>	<b>60.9</b>	<b>44.9</b>	21.5	<b>45.0</b>	<b>57.5</b>

Table 8: MSCOCO *test-dev* results for DeNet-101 model with Joint Fitness NMS, Bounded IoU Loss and corner based RoI clustering. The maximum input dimension size was varied over 384, 512, 768, 1024, 1280 and 1536 with the ratio between the number of sampling RoIs and mean input pixels kept constant. For comparison at a similar pixel density and evaluation rate as RetinaNet we merged the  $\times 512$  and  $\times 1024$  results to obtain the bottom row.

ing method appears to perform slightly better at high matching IoU and worse at low matching IoUs.

## 6. Input Image Scaling

So far our model has been demonstrated at low input image size, i.e., rescaling the *largest* input dimension to 512 pixels. In comparison, state-of-the-art detectors typically use an input image with the *smallest* side resized to 600 or 800 pixels. This smaller input image has provided our model with an improved evaluation rate at the cost of object localization accuracy. In the following, we relax the constraint on evaluation rate to demonstrate localization accuracies when computational resources are less constrained.

In Table 8, we provide MSCOCO results for the DeNet-101 (wide) model with Fitness NMS, Bounded IoU Loss and Corner Clustering with the largest input image dimension varied from 384 to 1536. Note that the model was not retrained with these settings, only tested. Since the benchmarked methods use different input image scaling methods, we provide the mean input pixels per sample (ignoring black borders) calculated over the MSCOCO *test-dev* dataset. These results demonstrate that input image size is very important for small and medium sized object localization in MSCOCO, however, we observed a loss in precision for large objects as scale is increased. This precision asymmetry suggests that multi-scale evaluation is likely optimal with the current design.

## 7. Conclusion

We highlighted an issue in common detector designs and propose a novel *Fitness NMS* method to address it. This method significantly improves MAP at high localization ac-

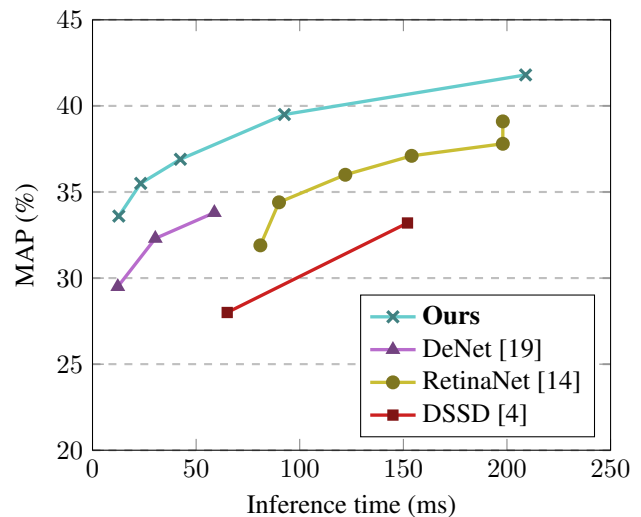


Figure 6: MSCOCO *test-dev* MAP vs Inference Time comparison selecting top DeNet-34 and DeNet-101 results.

curacies without a loss in evaluation rate. Following this we derive a novel bounding box loss better suited to IoU maximisation while still providing convergence properties suitable to gradient descent. Combining these results with a simple RoI clustering method, we obtain highly competitive MAP vs Inference Times on MSCOCO (see Figure 6). These results highlight that with these modification a two-stage detector can be made highly competitive with single-stage methods in terms of MAP vs inference time. Though not demonstrated empirically in this paper, we believe the novelties presented in this paper are equally applicable to other bounding box detector designs.



## References

- [1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms – improving object detection with one line of code. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [2] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [3] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.
- [4] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD : Deconvolutional single shot detector. *CoRR*, abs/1701.06659, 2017.
- [5] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015.
- [6] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.
- [7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [8] J. Hosang, R. Benenson, and B. Schiele. A convnet for non-maximum suppression. In *German Conference on Pattern Recognition*, pages 192–204. Springer, 2016.
- [9] J. Hosang, R. Benenson, and B. Schiele. Learning non-maximum suppression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3), 2017.
- [10] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, Mar. 1964.
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [12] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.
- [13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.
- [14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [15] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
- [18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [19] L. Tychsen-Smith and L. Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.