# Soft-NMS – Improving Object Detection With One Line of Code

Navaneeth Bodla* Bharat Singh* Rama Chellappa Larry S. Davis
Center For Automation Research, University of Maryland, College Park
{nbodla,bharat,rama,lsd}@umiacs.umd.edu

## Abstract

*Non-maximum suppression is an integral part of the object detection pipeline. First, it sorts all detection boxes on the basis of their scores. The detection box $\mathcal{M}$ with the maximum score is selected and all other detection boxes with a significant overlap (using a pre-defined threshold) with $\mathcal{M}$ are suppressed. This process is recursively applied on the remaining boxes. As per the design of the algorithm, if an object lies within the predefined overlap threshold, it leads to a miss. To this end, we propose Soft-NMS, an algorithm which decays the detection scores of all other objects as a continuous function of their overlap with $\mathcal{M}$. Hence, no object is eliminated in this process. Soft-NMS obtains consistent improvements for the coco-style mAP metric on standard datasets like PASCAL VOC 2007 (1.7% for both R-FCN and Faster-RCNN) and MS-COCO (1.3% for R-FCN and 1.1% for Faster-RCNN) by just changing the NMS algorithm without any additional hyper-parameters. Using Deformable-RFCN, Soft-NMS improves state-of-the-art in object detection from 39.8% to 40.9% with a single model. Further, the computational complexity of Soft-NMS is the same as traditional NMS and hence it can be efficiently implemented. Since Soft-NMS does not require any extra training and is simple to implement, it can be easily integrated into any object detection pipeline. Code for Soft-NMS is publicly available on GitHub* http://bit.ly/2nJLNMu.

## 1. Introduction

Object detection is a fundamental problem in computer vision in which an algorithm generates bounding boxes for specified object categories and assigns them classification scores. It has many practical applications in autonomous driving [6, 9], video/image indexing [28, 22], surveillance [2, 11] etc. Hence, any new component proposed for the object detection pipeline should not create a computational bottleneck, otherwise it will be conveniently "ignored' in practical implementations. Moreover, if a complex module is introduced which requires re-training of models which leads to a little improvement in performance, it will also be ignored. However, if a simple module can improve performance without requiring any re-training of existing models, it would be widely adopted. To this end, we present a soft non-maximum suppression algorithm, as an alternative to the traditional NMS algorithm in the current object detection pipeline.

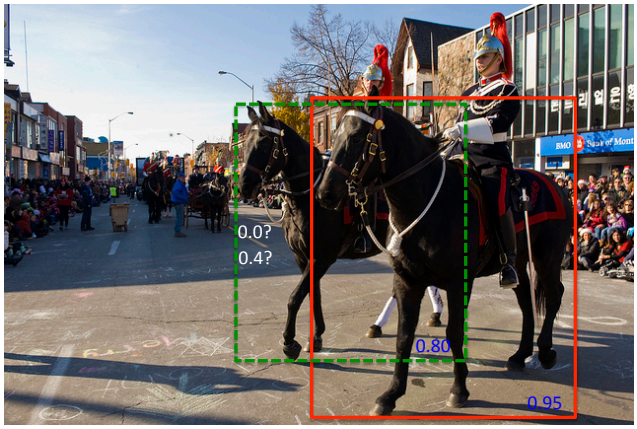


Figure 1. This image has two confident horse detections (shown in red and green) which have a score of 0.95 and 0.8 respectively. The green detection box has a significant overlap with the red one. Is it better to suppress the green box altogether and assign it a score of 0 or a slightly lower score of 0.4?

Traditional object detection pipelines [4, 8] employ a multi-scale sliding window based approach which assigns foreground/background scores for each class on the basis of features computed in each window. However, neighboring windows often have correlated scores (which increases false positives), so non-maximum suppression is used as a post-processing step to obtain final detections. With the advent of deep learning, the sliding window approach was replaced with category independent region proposals generated using a convolutional neural network. In state-of-the-art detectors, these proposals are input to a classification sub-network which assigns them class specific scores

*The first two authors contributed equally to this paper.

**Input** : $\mathcal{B} = \{b_1, .., b_N\}, \mathcal{S} = \{s_1, .., s_N\}, N_t$
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
$N_t$ is the NMS threshold

**begin**
  $\mathcal{D} \leftarrow \{\}$
  **while** $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    **for** $b_i$ *in* $\mathcal{B}$ **do**
      **if** $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$
      **end**
      NMS
      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$
      Soft-NMS
    **end**
  **end**
  **return** $\mathcal{D}, \mathcal{S}$
**end**

Figure 2. The pseudo code in red is replaced with the one in green in Soft-NMS. We propose to revise the detection scores by scaling them as a linear or Gaussian function of overlap.

[16, 24]. Another parallel regression sub-network refines the position of these proposals. This refinement process improves localization for objects, but also leads to cluttered detections as multiple proposals often get regressed to the same region of interest (RoI). Hence, even in state-of-the-art detectors, non-maximum suppression is used to obtain the final set of detections as it significantly reduces the number of false positives.

Non-maximum suppression starts with a list of detection boxes $\mathcal{B}$ with scores $\mathcal{S}$. After selecting the detection with the maximum score $\mathcal{M}$, it removes it from the set $\mathcal{B}$ and appends it to the set of final detections $\mathcal{D}$. It also removes any box which has an overlap greater than a threshold $N_t$ with $\mathcal{M}$ in the set $\mathcal{B}$. This process is repeated for remaining boxes $\mathcal{B}$. A major issue with non-maximum suppression is that it sets the score for neighboring detections to zero. Thus, if an object was actually present in that overlap threshold, it would be missed and this would lead to a drop in average precision. However, if we lower the detection scores as a function of its overlap with $\mathcal{M}$, it would still be in the ranked list, although with a lower confidence. We show an illustration of the problem in Fig 1.

Using this intuition, we propose a single line modification to the traditional greedy NMS algorithm in which we decrease the detection scores as an increasing function of overlap instead of setting the score to zero as in NMS. Intuitively, if a bounding box has a very high overlap with $\mathcal{M}$, it should be assigned a very low score, while if it has a low overlap, it can maintain its original detection score. This Soft-NMS algorithm is shown in Figure 2. Soft-NMS leads to noticeable improvements in average precision measured over multiple overlap thresholds for state-of-the-object detectors on standard datasets like PASCAL VOC and MS-COCO. Since Soft-NMS does not require any extra-training and is simple to implement, it can be easily integrated in the object detection pipeline.

## 2. Related Work

NMS has been an integral part of many detection algorithms in computer vision for almost 50 years. It was first employed in edge detection techniques [25]. Subsequently, it has been applied to multiple tasks like feature point detection [19, 12, 20], face detection [29] and object detection [4, 8, 10]. In edge detection, NMS performs edge thinning to remove spurious responses [25, 1, 31]. In feature point detectors [12], NMS is effective in performing local thresholding to obtain unique feature point detections. In face detection [29], NMS is performed by partitioning bounding-boxes into disjoint subsets using an overlap criterion. The final detections are obtained by averaging the co-ordinates of the detection boxes in the set. For human detection, Dalal and Triggs [4] demonstrated that a greedy NMS algorithm, where a bounding box with the maximum detection score is selected and its neighboring boxes are suppressed using a pre-defined overlap threshold improves performance over the approach used for face detection [29]. Since then, greedy NMS has been the *de-facto* algorithm used in object detection [8, 10, 24, 16].

It is surprising that this component of the detection pipeline has remained untouched for more than a decade. Greedy NMS still obtains the best performance when average precision (AP) is used as an evaluation metric and is therefore employed in state-of-the-art detectors [24, 16]. A few learning-based methods have been proposed as an alternative to greedy NMS which obtain good performance for *object class detection* [5, 26, 21]. For example, [26] first computes overlap between each pair of detection boxes. It then performs affinity propagation clustering to select exemplars for each cluster which represent the final detection boxes. A multi-class version of this algorithm is proposed in [21]. However, object class detection is a different problem, where object instances of all classes are evaluated simultaneously per image. Hence, we need to select a threshold for all classes and generate a fixed set of boxes. Since different thresholds may be suitable for different applications, in generic object detection, average precision is computed using a ranked list of all object instances in a particular class. Therefore, greedy NMS performs favourably to these algo-

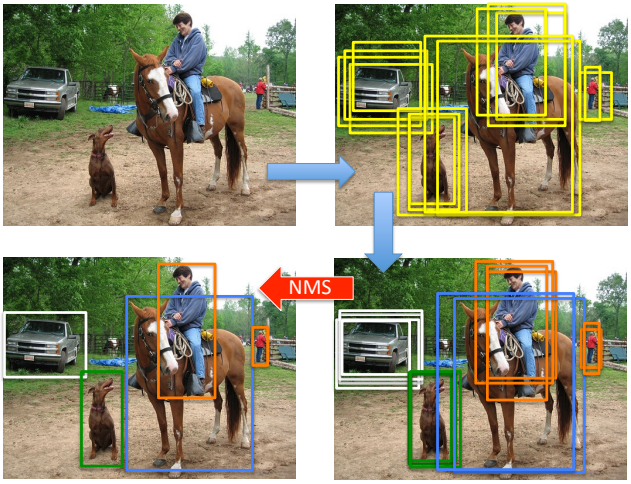


Figure 3. In object detection, first category independent region proposals are generated. These region proposals are then assigned a score for each class label using a classification network and their positions are updated slightly using a regression network. Finally, non-maximum-suppression is applied to obtain detections.

rithms on generic object detection metrics.

In another line of work, for detecting salient objects, a proposal subset optimization algorithm was proposed [30] as an alternative to greedy NMS. It performs a MAP-based subset optimization to jointly optimize the number and locations of detection windows. In salient object detection, the algorithm is expected to only find salient objects and not all objects. So, this problem is also different from generic object detection and again greedy NMS performs favourably when performance on object detection metrics is measured. For special cases like pedestrian detection, a quadratic unconstrained binary optimization (QUBO) solution was proposed which uses detection scores as a unary potential and overlap between detections as a pairwise potential to obtain the optimal subset of detection boxes [27]. Like greedy NMS, QUBO also applies a hard threshold to suppress detection boxes, which is different from Soft-NMS. In another learning-based framework for pedestrian detection, a determinantal point process was combined with individualness prediction scores to optimally select final detections [15]. To the best of our knowledge, for generic object detection, greedy NMS is still the strongest baseline on challenging object detection datasets like PASCAL VOC and MS-COCO.

## 3. Background

We briefly describe the object-detection pipeline used in state-of-the-art object detectors in this section. During inference, an object detection network performs a sequence of convolution operations on an image using a deep convolutional neural network (CNN). The network bifurcates into two branches at a layer $L$ — one branch generates region proposals while the other performs classification and regression by pooling convolutional features inside RoIs generated by the proposal network. The proposal network generates classification scores and regression offsets for anchor boxes of multiple scales and aspect ratios placed at each pixel in the convolutional feature map [24]. It then ranks these anchor boxes and selects the top $K$ ($\approx 6000$) anchors to which the bounding box regression offsets are added to obtain image level co-ordinates for each anchor. Greedy non-maximum suppression is applied to top $K$ anchors which eventually generates region proposals [1].

The classification network generates classification and regression scores for each proposal generated by the proposal network. Since there is no constraint in the network which forces it to generate a unique RoI for an object, multiple proposals may correspond to the same object. Hence, other than the first correct bounding-box, all other boxes on the same object would generate false positives. To alleviate this problem, non-maximum-suppression is performed on detection boxes of each class independently, with a specified overlap threshold. Since the number of detections is typically small and can be further reduced by pruning detections which fall below a very small threshold, applying non-maximum suppression at this stage is not computationally expensive. We present an alternative approach to this non-maximum suppression algorithm in the object detection pipeline. An overview of the object detection pipeline is shown in Fig 3.

## 4. Soft-NMS

Current detection evaluation criteria emphasise precise localization and measure average precision of detection boxes at multiple overlap thresholds (ranging from 0.5 to 0.95). Therefore, applying NMS with a low threshold like 0.3 could lead to a drop in average precision when the overlap criterion during evaluation for a true positive is 0.7 (we refer to the detection evaluation threshold as $O_t$ from here on). This is because, there could be a detection box $b_i$ which is very close to an object (within 0.7 overlap), but had a slightly lower score than $\mathcal{M}$ ($\mathcal{M}$ did not cover the object), thus $b_i$ gets suppressed by a low $N_t$. The likelihood of such a case would increase as the overlap threshold criterion is increased. Therefore, suppressing all nearby detection boxes with a low $N_t$ would increase the miss-rate.

Also, using a high $N_t$ like 0.7 would increase false positives when $O_t$ is lower and would hence drop precision averaged over multiple thresholds. The increase in false positives would be much higher than the increase in true positives for this case because the number of objects is typically

[1] We do not replace this non-maximum suppression in the object detection pipeline.

much smaller than the number of RoIs generated by a detector. Therefore, using a high NMS threshold is also not optimal.

To overcome these difficulties, we revisit the NMS algorithm in greater detail. The pruning step in the NMS algorithm can be written as a re-scoring function as follows,

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ 0, & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

Hence, NMS sets a hard threshold while deciding what should be kept or removed from the neighborhood of $\mathcal{M}$. Suppose, instead, we decay the classification score of a box $b_i$ which has a high overlap with $\mathcal{M}$, rather than suppressing it altogether. If $b_i$ contains an object not covered by $\mathcal{M}$, it won't lead to a miss at a lower detection threshold. However, if $b_i$ does not cover any other object (while $\mathcal{M}$ covers an object), and even after decaying its score it ranks above true detections, it would still generate a false positive. Therefore, NMS should take the following conditions into account,

- Score of neighboring detections should be decreased to an extent that they have a smaller likelihood of increasing the false positive rate, while being above obvious false positives in the ranked list of detections.
- Removing neighboring detections altogether with a low NMS threshold would be sub-optimal and would increase the miss-rate when evaluation is performed at high overlap thresholds.
- Average precision measured over a range of overlap thresholds would drop when a high NMS threshold is used.

We evaluate these conditions through experiments in Section 6.3.

**Rescoring Functions for Soft-NMS:** Decaying the scores of other detection boxes which have an overlap with $\mathcal{M}$ seems to be a promising approach for improving NMS. It is also clear that scores for detection boxes which have a higher overlap with $\mathcal{M}$ should be decayed more, as they have a higher likelihood of being false positives. Hence, we propose to update the pruning step with the following rule,

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

The above function would decay the scores of detections above a threshold $N_t$ as a linear function of overlap with $\mathcal{M}$. Hence, detection boxes which are far away from $\mathcal{M}$ would not be affected and those which are very close would be assigned a greater penalty.

However, it is not continuous in terms of overlap and a sudden penalty is applied when a NMS threshold of $N_t$ is reached. It would be ideal if the penalty function was continuous, otherwise it could lead to abrupt changes to the ranked list of detections. A continuous penalty function should have no penalty when there is no overlap and very high penalty at a high overlap. Also, when the overlap is low, it should increase the penalty gradually, as $\mathcal{M}$ should not affect the scores of boxes which have a very low overlap with it. However, when overlap of a box $b_i$ with $\mathcal{M}$ becomes close to one, $b_i$ should be significantly penalized. Taking this into consideration, we propose to update the pruning step with a Gaussian penalty function as follows,

$$s_i = s_i e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}, \forall b_i \notin \mathcal{D}$$

This update rule is applied in each iteration and scores of *all remaining* detection boxes are updated.

The Soft-NMS algorithm is formally described in Figure 2, where $f(iou(\mathcal{M}, b_i)))$ is the overlap based weighting function. The computational complexity of each step in Soft-NMS is $\mathcal{O}(N)$, where $N$ is the number of detection boxes. This is because scores for all detection boxes which have an overlap with $\mathcal{M}$ are updated. So, for $N$ detection boxes, the computational complexity for Soft-NMS is $\mathcal{O}(N^2)$, which is the same as traditional greedy-NMS. Since NMS is not applied on all detection boxes (boxes with a minimum threshold are pruned in each iteration), this step is not computationally expensive and hence does not affect the running time of current detectors.

Note that Soft-NMS is also a greedy algorithm and does not find the globally optimal re-scoring of detection boxes. Re-scoring of detection boxes is performed in a greedy fashion and hence those detections which have a high local score are not suppressed. However, Soft-NMS is a *generalized* version of non-maximum suppression and traditional NMS is a special case of it with a discontinuous binary weighting function. Apart from the two proposed functions, other functions with more parameters can also be explored with Soft-NMS which take overlap and detection scores into account. For example, instances of the generalized logistic function like the Gompertz function can be used, but such functions would increase the number of hyper-parameters.

## 5. Datasets and Evaluation

We perform experiments on two datasets, PASCAL VOC [7] and MS-COCO [17]. The Pascal dataset has 20 object categories, while the MS-COCO dataset has 80 object categories. We choose the VOC 2007 test partition to measure performance. For the MS-COCO dataset, sensitivity analysis is conducted on a publicly available minival set of 5,000 images. We also show results on the test-dev partition on

| Method | Training data | Testing data | AP 0.5:0.95 | AP @ 0.5 | AP small | AP medium | AP large | Recall @ 10 | Recall @ 100 |
|---|---|---|---|---|---|---|---|---|---|
| R-FCN [16] | train+val35k | test-dev | 31.1 | 52.5 | 14.4 | 34.9 | 43.0 | 42.1 | 43.6 |
| R-FCN + S-NMS G | train+val35k | test-dev | **32.4** | **53.4** | **15.2** | **36.1** | **44.3** | **46.9** | **52.0** |
| R-FCN + S-NMS L | train+val35k | test-dev | 32.2 | **53.4** | 15.1 | 36.0 | 44.1 | 46.0 | 51.0 |
| F-RCNN [24] | train+val35k | test-dev | 24.4 | 45.7 | 7.9 | 26.6 | 37.2 | 36.1 | 37.1 |
| F-RCNN + S-NMS G | train+val35k | test-dev | **25.5** | 46.6 | **8.8** | **27.9** | **38.5** | **41.2** | 45.3 |
| F-RCNN + S-NMS L | train+val35k | test-dev | **25.5** | **46.7** | **8.8** | **27.9** | 38.3 | 40.9 | **45.5** |
| D-RFCN [3] | trainval | test-dev | 37.4 | 59.6 | 17.8 | 40.6 | 51.4 | 46.9 | 48.3 |
| D-RFCN S-NMS G | trainval | test-dev | **38.4** | **60.1** | **18.5** | **41.6** | **52.5** | **50.5** | **53.8** |
| D-RFCN + MST | trainval | test-dev | 39.8 | 62.4 | 22.6 | 42.3 | 52.2 | 50.5 | 52.9 |
| D-RFCN + MST + S-NMS G | trainval | test-dev | **40.9** | **62.8** | **23.3** | **43.6** | **53.3** | **54.7** | **60.4** |

Table 1. Results on MS-COCO test-dev set for R-FCN, D-RFCN and Faster-RCNN (F-RCNN) which use NMS as baseline and our proposed Soft-NMS method. G denotes Gaussian weighting and L denotes linear weighting. MST denotes multi-scale testing.

| Method | AP | AP@0.5 | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [24]+NMS | 37.7 | 70.0 | 37.8 | 44.6 | 34.7 | 24.4 | 23.4 | 50.6 | 50.1 | 45.1 | 25.1 | 42.6 | 36.5 | 40.7 | 46.8 | 39.8 | 38.2 | 17.0 | 37.8 | 36.4 | 43.7 | **38.9** |
| [24]+S-NMS G | **39.4** | **71.2** | 40.2 | **46.6** | **36.7** | 25.9 | **24.9** | **51.9** | 51.6 | **48.0** | **25.3** | 44.5 | **37.3** | 42.6 | **49.0** | 42.2 | 41.6 | 17.7 | 39.2 | **39.3** | **45.9** | 37.5 |
| [24]+S-NMS L | **39.4** | **71.2** | **40.3** | **46.6** | 36.3 | **27.0** | 24.2 | 51.2 | **52.0** | 47.2 | **25.3** | **44.6** | 37.2 | **45.1** | 48.3 | **42.3** | **42.3** | **18.0** | **39.4** | 37.1 | 45.0 | 38.7 |
| [16]+NMS | 49.8 | 79.4 | 52.8 | 54.4 | 47.1 | 37.6 | 38.1 | 63.4 | 59.4 | 62.0 | 35.3 | 56.0 | 38.9 | 59.0 | 54.5 | 50.5 | 47.6 | 24.8 | 53.3 | 52.2 | 57.4 | 52.7 |
| [16]+S-NMS G | 51.4 | **80.0** | **53.8** | **56.0** | 48.3 | 39.9 | 39.4 | **64.7** | 61.3 | 64.7 | **36.3** | **57.0** | **40.2** | 60.6 | 55.5 | 52.1 | **50.7** | **26.5** | 53.8 | 53.5 | 59.3 | 53.8 |
| [16]+S-NMS L | **51.5** | **80.0** | 53.2 | 55.8 | **48.9** | **40.0** | **39.6** | 64.6 | **61.5** | **65.0** | **36.3** | 56.5 | **40.2** | **61.3** | **55.6** | **52.9** | 50.3 | 26.2 | **54.3** | **53.6** | **59.5** | **53.9** |

Table 2. Results on Pascal VOC 2007 test set for off-the-shelf standard object detectors which use NMS as baseline and our proposed Soft-NMS method. Note that COCO-style evaluation is used.

the MS-COCO dataset which consists of 20,288 images.

To evaluate our method, we experimented with three state-of-the-art detectors, namely, Faster-RCNN [24], R-FCN [16] and Deformable-RFCN. For the PASCAL dataset, we selected publicly available pre-trained models provided by the authors. The Faster-RCNN detector was trained on VOC 2007 train set while the R-FCN detector was trained on VOC 2007 and 2012. For MS-COCO also, we use the publicly available model for Faster-RCNN. However, since there was no publicly available model trained on MS-COCO for R-FCN, we trained our own model in Caffe [14] starting from a ResNet-101 CNN architecture [13]. Simple modifications like 5 scales for RPN anchors, a minimum image size of 800, 16 images per minibatch and 256 ROIs per image were used. Training was done on 8 GPUs in parallel. Note that our implementation obtains 1.9% better accuracy than that reported in [16] without using multi-scale training or testing. Hence, this is a strong baseline for R-FCN on MS-COCO. Both these detectors use a default NMS threshold of 0.3. In the sensitivity analysis section, we also vary this parameter and show results. We also trained deformable R-FCN with the same settings. At a threshold of 10e-4, using 4 CPU threads, it takes 0.01s per image for 80 classes. After each iteration, detections which fall below the threshold are discarded. This reduces computation time. At 10e-2, run time is 0.005 seconds on a single core. We set maximum detections per image to 400 on MS-COCO and the evaluation server selects the top 100 detections per class for generating metrics (we confirmed that the coco evaluation server was not selecting top 100 scoring detections per image till June 2017). Setting maximum detections to 100 reduces coco-style AP by 0.1.

## 6. Experiments

In this section, we show comparative results and perform sensitivity analysis to show robustness of Soft-NMS compared to traditional NMS. We also conduct specific experiments to understand why and where does Soft-NMS perform better compared to traditional NMS.

### 6.1. Results

In Table 1 we compare R-FCN and Faster-RCNN with traditional non-maximum suppression and Soft-NMS on MS-COCO. We set $N_t$ to 0.3 when using the linear weighting function and $\sigma$ to 0.5 with the Gaussian weighting function. It is clear that Soft-NMS (using both Gaussian and linear weighting function) improves performance in all cases, especially when AP is computed at multiple overlap thresholds and averaged. For example, we obtain an improvement of 1.3% and 1.1% respectively for R-FCN and Faster-RCNN, which is significant for the MS-COCO dataset. Note that we obtain this improvement by just changing the NMS algorithm and hence it can be applied easily on multiple detectors with minimal changes. We perform the same experiments on the PASCAL VOC 2007 test set, shown in Table 1. We also report average precision averaged over multiple overlap thresholds like MS-COCO. Even on PASCAL VOC 2007, Soft-NMS obtains an improvement of 1.7% for both Faster-RCNN and R-FCN. For detectors like SSD [18] and YOLOv2 [23] which are not proposal based, with the linear function, Soft-NMS only obtains an improvement of 0.5%. This is because proposal

| $N_t$ | AP @ 0.5 | AP @ 0.6 | AP @ 0.7 | AP @ 0.8 | $\sigma$ | AP @ 0.5 | AP @ 0.6 | AP @ 0.7 | AP @ 0.8 |
|---|---|---|---|---|---|---|---|---|---|
| 0.3 | 0.5193 | 0.4629 | 0.3823 | 0.2521 | 0.1 | 0.5202 | 0.4655 | 0.3846 | 0.2533 |
| 0.4 | **0.5238** | 0.4680 | 0.3840 | 0.2524 | 0.3 | **0.5293** | 0.4765 | 0.3960 | 0.2619 |
| 0.5 | 0.5227 | **0.4708** | 0.3869 | 0.2526 | 0.5 | 0.5274 | **0.4777** | 0.3997 | 0.2669 |
| 0.6 | 0.5127 | 0.4690 | **0.3895** | 0.2527 | 0.7 | 0.5232 | 0.4757 | **0.4001** | 0.2695 |
| 0.7 | 0.4894 | 0.4535 | 0.3860 | **0.2535** | 0.9 | 0.5186 | 0.4727 | 0.3992 | 0.2710 |
| 0.8 | 0.4323 | 0.4048 | 0.3569 | 0.2520 | 1.1 | 0.5136 | 0.4691 | 0.3976 | **0.2713** |

Table 3. Sensitivity Analysis across multiple overlap thresholds $N_t$ and parameters $\sigma$ for NMS and Soft-NMS using R-FCN on coco minival. Best performance at each $O_t$ is marked in bold for each method.

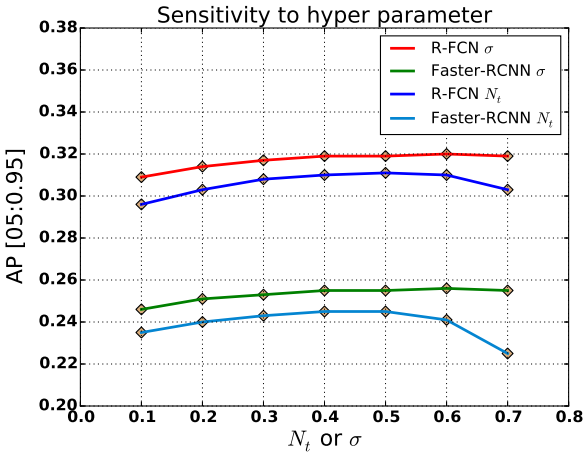


Figure 4. R-FCN Sensitivity to hyper parameters $\sigma$ (Soft-NMS) and $N_t$ (NMS)

based detectors have higher recall and hence Soft-NMS has more potential to improve recall at higher $O_t$.

From here on, in all experiments, when we refer to Soft-NMS, it uses the Gaussian weighting function. In Fig 6, we also show per-class improvement on MS-COCO. It is interesting to observe that Soft-NMS when applied on R-FCN improves maximum performance for animals which are found in a herd like zebra, giraffe, sheep, elephant, horse by 3-6%, while there is little gain for objects like toaster, sports ball, hair drier which are less likely to co-occur in the same image.

### 6.2. Sensitivity Analysis

Soft-NMS has a $\sigma$ parameter and traditional NMS has an overlap threshold parameter $N_t$. We vary these parameters and measure average precision on the minival set of MS-COCO set for each detector, see Fig 4. Note that AP is stable between 0.3 to 0.6 and drops significantly outside this range for both detectors. The variation in AP in this range is around 0.25% for traditional NMS. Soft-NMS obtains better performance than NMS from a range between 0.1 to 0.7. Its performance is stable from 0.4 to 0.7 and better by $\sim$1% for each detector even on the best NMS threshold selected by us on the coco-minival set. In all our experiments, we set $\sigma$ to 0.5, even though a $\sigma$ value of 0.6 seems to give better performance on the coco minival set. This is because we conducted the sensitivity analysis experiments later on and a difference of 0.1% was not significant.

### 6.3. When does Soft-NMS work better?

**Localization Performance** Average precision alone does not explain us clearly when Soft-NMS obtains significant gains in performance. Hence, we present average precision of NMS and Soft-NMS when measured at different overlap thresholds. We also vary the NMS and Soft-NMS hyper-parameters to understand the characteristics of both these algorithms. From Table 3, we can infer that average precision decreases as NMS threshold is increased. Although it is the case that for a large $O_t$, a high $N_t$ obtains slightly better performance compared to a lower $N_t$ — AP does not drop significantly when a lower $N_t$ is used. On the other hand, using a high $N_t$ leads to significant drop in AP at lower $O_t$ and hence when AP is averaged at multiple thresholds, we observe a performance drop. Therefore, a better performance using a higher $N_t$ does not generalize to lower values of $O_t$ for traditional NMS.

However, when we vary $\sigma$ for Soft-NMS, we observe a different characteristic. Table 3 shows that even when we obtain better performance at higher $O_t$, performance at lower $O_t$ does not drop. Further, we observe that Soft-NMS performs significantly better ($\sim$2%) than traditional NMS irrespective of the value of the selected $N_t$ at higher $O_t$. Also, the best AP for any hyper-parameter ($N_t$ or $\sigma$) for a selected $O_t$ is always better for Soft-NMS. This comparison makes it very clear that across all parameter settings, the best $\sigma$ parameter for Soft-NMS performs better than a hard threshold $N_t$ selected in traditional NMS. Further, when performance across all thresholds is averaged, since a single parameter setting in Soft-NMS works well at multiple values of $O_t$, overall performance gain is amplified. As expected, low values of $\sigma$ perform better at lower $O_t$ and higher values of sigma perform better at higher $O_t$. Unlike NMS, where higher values of $N_t$ lead to very little improvement in AP, higher values of $\sigma$ lead to significant improvement in AP at a higher $O_t$. Therefore, a larger $\sigma$ can be used

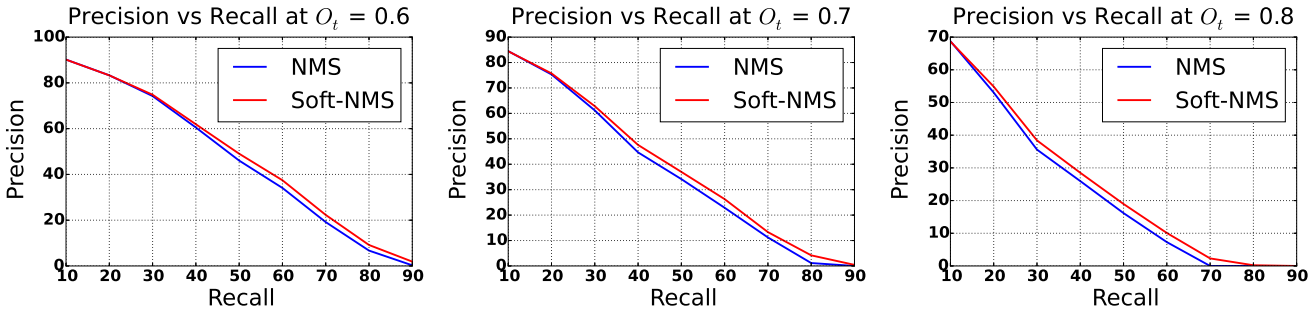


Figure 5. R-FCN : Precision vs Recall at multiple overlap thresholds $O_t$

to improve performance of the detector for better localization which is not the case with NMS, as a larger $N_t$ obtains very little improvement.

**Precision vs Recall** Finally, we would like to also know at what recall values is Soft-NMS performing better than NMS at different $O_t$. Note that we re-score the detection scores and assign them lower scores, so we do not expect precision to improve at a lower recall. However, as $O_t$ and recall is increased, Soft-NMS obtains significant gains in precision. This is because, traditional NMS assigns a zero score to all boxes which have an overlap greater than $N_t$ with $\mathcal{M}$. Hence, many boxes are missed and therefore precision does not increase at higher values of recall. Soft-NMS re-scores neighboring boxes instead of suppressing them altogether which leads to improvement in precision at higher values of recall. Also, Soft-NMS obtains significant improvement even for lower values of recall at higher values of $O_t$ because near misses are more likely to happen in this setting.

## 6.4. Qualitative Results

We show a few qualitative results in Fig 7 using a detection threshold of 0.45 for images from the COCO-validation set. The R-FCN detector was used to generate detections. It is interesting to observe that Soft-NMS helps in cases when bad detections (false positives) have a small overlap with a good detection (true positive) and also when they have a low overlap with a good detection. For example, in the street image (No.8), a large wide bounding box spanning multiple people is suppressed because it had a small overlap with multiple detection boxes with a higher score than it. Hence, its score was reduced multiple times because of which it was suppressed. We observe a similar behaviour in image No.9. In the beach image (No.1), the score for the larger bounding box near the woman's handbag is suppressed below 0.45. We also see that a false positive near the bowl in the kitchen image (No.4) is suppressed. In other cases, like for zebra, horse and giraffe images (images 2,5,7 and 13), the detection boxes get suppressed with NMS while Soft-NMS assigns a slightly lower score for neighboring boxes

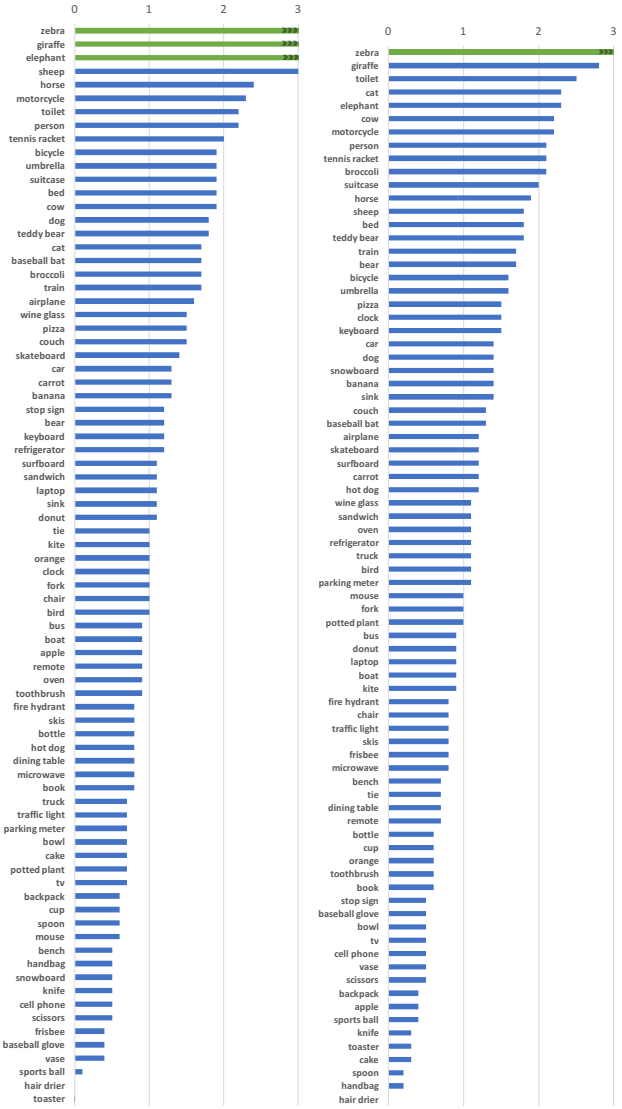

Figure 6. Per class improvement in AP for MS-COCO using Soft-NMS for R-FCN is shown in the left and for Faster-RCNN is shown on the right. Green bars indicate improvements beyond 3 %

because of which we are able to detect true positives above a detection threshold of 0.45.

## References


[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 2

[2] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt,


Figure 7. Qualitative results: Image pairs are shown in which the left image with blue bounding boxes is for traditional NMS, while the right image with red bounding boxes is for Soft-NMS. Examples above the blue line are successful cases and below are failure cases. Even in failure cases, Soft-NMS assigns a significantly lower score than the bounding box with a higher score. Note that for image No. 14, the classes in consideration are for "person" and not "dog", similarly in 15 it is "bench" and in 21 it is "potted plant".

et al. A system for video surveillance and monitoring. 2000. 1

[3] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017. 5

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 1, 2

[5] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 95(1):1–12, 2011. 2

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009. 1

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 1, 2

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1

[10] 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. 2

[11] I. Haritaoglu, D. Harwood, and L. S. Davis. W/sup 4: real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, 2000. 1

[12] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 2

[13] 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*, pages 770–778, 2016. 5

[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. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 5

[15] D. Lee, G. Cha, M.-H. Yang, and S. Oh. Individualness and determinantal point processes for pedestrian detection. In *European Conference on Computer Vision*, pages 330–346. Springer, 2016. 3

[16] 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. 2, 5

[17] 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. 4

[18] 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. 5

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[20] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004. 2

[21] D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell. Spatial semantic regularisation for large scale object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2003–2011, 2015. 2

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 1

[23] 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. 5

[24] 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. 2, 3, 5

[25] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971. 2

[26] R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian Conference on Computer Vision*, pages 290–306. Springer, 2014. 2

[27] S. Rujikietgumjorn and R. T. Collins. Optimized pedestrian detection for multiple and occluded people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3690–3697, 2013. 3

[28] J. Sivic, A. Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In *iccv*, volume 2, pages 1470–1477, 2003. 1

[29] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 2

[30] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Unconstrained salient object detection via proposal subset optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5733–5742, 2016. 3

[31] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 2