Directly Optimizing IoU for Bounding Box Localization

Mofassir ul Islam Arif[1], Mohsan Jameel[1], and Lars Schmidt-Thieme[1]

Information Systems and Machine Learning Lab (ISMLL)
Univeristy of Hildesheim, Hildsheim, Germany
{mofassir,mohsan.jameel,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Object detection has seen remarkable progress in recent years with the introduction of Convolutional Neural Networks (CNN). Object detection is a multi-task learning problem where both the position of the objects in the images as well as their classes needs to be correctly identified. The idea here is to maximize the overlap between the ground-truth bounding boxes and the predictions i.e. the Intersection over Union (IoU). In the scope of work seen currently in this domain, IoU is approximated by using the Huber loss as a proxy but this indirect method does not leverage the IoU information and treats the bounding box as four independent, unrelated terms of regression. This is not true for a bounding box where the four coordinates are highly correlated and hold a semantic meaning when taken together. The direct optimization of the IoU is not possible due to its non-convex and non-differentiable nature. In this paper, we have formulated a novel loss namely, the Smooth IoU, which directly optimizes the IoUs for the bounding boxes. This loss has been evaluated on the Oxford IIIT Pets, Udacity self-driving car, PASCAL VOC, and VWFS Car Damage datasets and has shown performance gains over the standard Huber loss.

Keywords: Object Detection · IoU Loss · Faster RCNN.

1 Introduction

Object detection is a multi-task learning problem with the goal of correctly identifying the object in the image while also localizing the object into a bounding box, therefore the end result of the object detection is to classify and localize the object. As with all machine learning models, the optimization is dictated by a loss that updates a loss towards a local optimum solution. The family of object detection models [5] [16] [9] [15] is accompanied by multi-task [2] losses which are made up of a localization loss $\mathcal{L}_{loc}$ and a classification loss $\mathcal{L}_{cls}$, for each stage. For the first stage the $\mathcal{L}_{loc}$ is used to distinguish between the raw proposals from a Region Proposal Network (RPN) usually modeled by a Fully Convolutional Network (FCN) [10], and the ground truth bounding boxes. The aim here is to separate the background and the foreground, based on the bounding boxes, therefore, the classification loss $\mathcal{L}_{cls}$ becomes a binary classification problem between the foreground and the background. The output of this stage is passed to the second stage where second stage localization and classification losses are used. In the second stage, bounding box localization deals with the actual objects rather than the background and foreground. Similarly the second stage classification loss is now a K-way softmax where K is the number of classes. For each stage, these losses are jointly optimized during training by forming a linear combination of the two, therefore the total loss for each stage is:

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{cls} \tag{1}$$

During training, ground-truth bounding boxes are used to train the model to learn the features of the objects that are present within the constraints of the boxes. Traditionally the two-stage methods rely on the Huber loss [1] for bounding box localization in both stages. Eq. 2 shows the Huber loss, its popularity in R-CNN, Fast RCNN, Faster-RCNN, SSD and many others is due to its robustness against outliers. In our case, the outliers would be the bounding boxes that are very far away from the ground-truth.

$$L_\delta(z) = \begin{cases} \frac{1}{2}z^2, & \text{if } |z| < \delta \\ \delta|z| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \tag{2}$$

$$BB_{regression} = \min_{\theta} \mathcal{L}(\beta, \hat{\beta}(\theta)) \tag{3}$$

Here z is the L1 loss between the ground-truth and predicted bounding boxand δ is a threshold parameter. The bounding box localization therefore, is treated as a regression problem as seen in Eq. 3. Where β is the ground truth bounding box and $\hat{\beta}(\theta)$ is the prediction model, parametrized by θ which are the parameters learned during the training phase, the output is predicted bounding boxes. Each bounding box is a tuple $((x_1, y_1), (x_2, y_2))$ which represents the coordinates on the diagonal of the box. This regression problem deals with each of the four parameters of the bounding box as independent and unrelated items however semantically that is not the case since the four coordinates of the bounding box are highly correlated and need to be treated as a single entity.

Huber loss, used in bounding box localization, has a quadratic behavior for values $|z| < \delta$ which enables faster convergence when the difference (location, size, scale) between the ground-truth and predictions become small. For the regions where the difference between the boxes is greater than the threshold δ the Huber loss evaluates the L1 loss which has been shown to be less aggressive against outliers, this prevents exploding gradients due to large penalties, a behavior that is seen by the penalty incurred by the squared loss. While this loss has shown to be a good surrogate by casting the maximization of IoU between ground-truth and predicted bounding boxes as a four-point regression, it does not use the IoU information during optimization. Furthermore, as stated earlier, it conducts the bounding box regression without considering the parameters of a bounding box to be highly correlated items which hold a semantic meaning

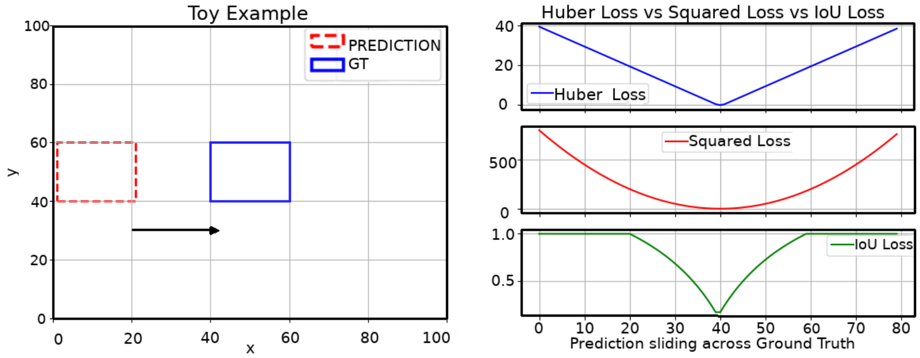

Fig. 1. The left figure mimicks the behaviour of a model predicting incrementally correct bounding boxes. The prediction is 'slid' over the ground-truth to examine the effect on the different losses. The figure on the right shows the behavior of the losses.

when taken together. Therefore, it stands to reason that the optimization of the object detection loss, more specifically the bounding box localization should involve a direct optimization of the IoU. The calculation of the IoU can be seen in Eq. 4. Here $\beta = ((x_1, y_1), (x_2, y_2))$ is the ground-truth bounding box and $\hat{\beta} = ((\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2))$ is the predicted bounding box.

$$IoU = \frac{I_w \times I_h}{Area_1 + \hat{Area_2} - (I_w \times I_h)} \tag{4}$$

The areas for the bounding boxes are calculated as $Area = (x_2 - x_1) \times (y_2 - y_1)$. Converting the IoU measure into a loss function is trivial, since $\mathcal{L}_{IoU} = 1 - IoU$. The intersection term is calculated based on the region of overlap between the two boxes and it is as follows:

$$Intersection = I_w \times I_h \tag{5}$$

Here, $I_h = max(0, min_y - max_y)$ is the intersection height and where $min_y = min(y_2, \hat{y}_2)$ and $max_y = max(y_1, \hat{y}_1)$ are the minimum and maximum y-coordinates, respectively. Similarly, $I_w = max(0, min_x - max_x)$ is the intersection width with $min_x = min(x_2, \hat{x}_2)$ and $max_x = max(x_1, \hat{x}_1)$ as the minimum and maximum x-coordinates for the overlapping region, respectively. The product of I_h and I_w as in Eq. 5, yeilds the intersection. The denominator term for Eq. 4 shows the union term, here $Area_1$ and $Area_2$ are the ground-truth and predicted bounding box areas, respectively.

An examination of the behaviors for the different losses can be seen in Fig. 1. The example presented is designed to show the behavior of the losses as a model predicts bounding boxes that are translated over the ground-truth bounding box. The two boxes are not overlapping up until the point that the predicted bounding box reaches $(20, 40)$. At this point, the overlap starts to occur and increases until the point $(40, 40)$, at which the overlap is maximum and starts to decrease as the box continues moving to the right. The overlap drops to zero at point $(60, 40)$. For the sake of simplicity of the illustration, we are limiting the movement of the box to the x coordinate only in Fig. 1.

For Huber loss, we can see a linear decrease as the predicted box approaches the ground-truth and conversely the loss linearly increases as the box starts to exit from the ground-truth. While inside some threshold δ it behaves quadratically, for areas with no overlap we can see the robustness of the Huber loss as it does not incur a large loss value. The L2 loss shows a similar (increase and decrease) behavior but is not bound by any δ parameter and is, therefore, quadratic throughout. This leads to a very high penalty around the area where the boxes do not overlap. These effects are obvious when we compare the scales of the losses.

Lastly, we can see the IoU loss plateauing outside the region where the two bounding boxes do not overlap. Here it can be seen that for the regions with no overlap between the two boxes the loss plateaus leading to zero gradients thus, effectively making the learning process impossible for all gradient-based learning methods. The areas outside the intersection offer no help in the learning process due to the fact that the IoU is bounded in $[0, 1]$ and the worst case scenario i.e, no overlap always leads to a loss of 1, regardless of how far away the box is. From the loss profile, we can also see that the IoU loss is non-convex (due to the plateauing region violating Eq. 6) and non-differentiable.

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \tag{6}$$

The Huber loss does not suffer from these issues since the regression always returns the distance between the parameters of the bounding boxes. The shortcoming here is the inherent treatment of the parameters of the bounding boxes as independent and unrelated terms.

In this paper, we present a novel loss that addresses the shortcoming of the standard IoU loss, inherits the advantages of Huber loss, and enables a direct optimization of IoU for two-stage object detection networks. This is done by a proposed relaxation for the IoU loss which mitigates the non-differentiability and non-convexity of the loss without the need to sub-gradient or approximation methods [11]. We propose a dynamic loss, that leverages the gains of Huber loss while directly optimizing for IoU in bounding box localization. The main contributions of this paper are:

a) A robust loss that can be integrated readily into the two-stage models.

b) A performance guarantee that is lower bounded by the state-of-the-art performance.

c) Empirical analysis of the proposed method on standard object detection datasets to show how optimizing for IoU can lead to better bounding boxes (higher IoU).

2 Related Work

The choice of losses in machine learning is dictated heavily by the convergence and convexity of the loss [17]. The Huber loss ensures a stable convergence due to its piece-wise quadratic and convex nature [25]. This enabled the loss to be adapted readily in the bounding box regression for the object detection tasks [5] [16] [9]. Similar to the Huber loss there is also an interest in using the squared loss for the bounding box regression [3] [22]. This is however more susceptible to exploding gradients and is more sensitive to the learning rate and other such hyper-parameters [19]. In Fig. 1 we have demonstrated the behavior of these losses and how they, while suitable for regression, optimize for a proxy loss and a more direct approach for optimizing the IoU is needed. The disadvantage of the IoU loss stems from its non-differentiability and the disappearing gradients outside the regions of intersection. Attempts to addresses this problem by using the IoU loss by looking only at the pixel values inside the predictions and ground truth boxes with a non zero overlap [26]. They convert the IoU loss into a cross-entropy loss since $0 \leq IoU \leq 1$ by wrapping it in the natural log $\mathcal{L} = -ln(IoU)$. This conversion relies on using the IoU information after converting the tuple of four coordinates into a pixel map and then evaluating the IoU pixel-wise. Furthermore, they propose a novel architecture for their loss implementation thus might not be readily compatible with the other architectures used for the object detection tasks. Another related method looks at the complete replacement of the regression loss with their implementation of the IoU loss [14]. They approach the task of image segmentation and how the optimization of the IoU directly can serve to improve the overall performance of the model. They rely on a FCN which is a modified AlexNet [21] and present the work in light of how for the image segmentation task, the discrimination between the background and the foreground serves as an important step. However, optimizing for the overall accuracies could cause a model to encourage larger sized boxes. This can be the case when a larger portion (90%) of the image belongs to the background, in such a situation a naive algorithm can get 90% accuracy simply by predicting everything to be the background [26]. A case like this can be made for using the Huber loss for the bounding box regression which the Huber loss treats four independent and unrelated items during its optimization. The use of Bayesian decision theory has also been attempted by [12] where Conditional Random Field (CRF) is used to maximize the expected-IoU, they also use the pixel values and a greedy heuristic for the optimization of IoU. A pixel-wise approach is inherently slower since it is dictated by the number of pixels in the bounding box. A bounding boxwith size $P \times Q$ where P is the width and Q is the heigth would require $O(PQ)$ operations in order to calculate the IoU. Whereas, by treating the bounding box as a tuple and calculating IoU as in Eq. 4 the number of operations is constant regardless of the size.

3 Methodology

The IoU loss suffers from the plateauing phenomenon because of the unavailability of gradient information since $L_{IoU} \in [0,1]$ where it is constant (1) outside the region of intersection as shown in Fig. 1. This gradient information in a standard Huber loss, for bounding box localization, is available throughout due to the regression between the four points of the ground-truth and predicted bounding box. The vanishing gradients of IoU loss for bounding boxes with no overlap hinder the learning process since two bounding boxes with no overlap present the same constant loss (zero gradients) regardless of how far they are from the ground-truth. A relaxation is needed for the IoU loss that will enable us to bring in the gradient information for the predicted bounding box in terms of the distance, and consequently guide the model in the correct direction. Albeit, this is needed only in the initial learning stage because once the predicted bounding boxes begin to overlap with the ground truth ones, non-zero overlap will address the plateauing behavior of the IoU loss. Standard object detection models treat the regression as an independent and unrelated four-way entity which is not true for a bounding box.

In order to optimize the true goal of object detection, we need to directly optimize for IoU in the bounding box localization loss. Our method proposes to morph the Huber loss in order to include the IoU information.

3.1 Smooth IoU Loss

A smooth stiched loss, named Smooth IoU is presented as an improvement on the Huber loss to enhances the localization of the bounding boxes while also overcoming the non-convexity issues (stemming from the IoU loss being bounded in $[0,1]$) of the vanilla IoU loss, and is presented in Eq. 7.

$$\mathcal{L}_{SmoothIoU} = \lambda \mathcal{L}_{IoU} + (1-\lambda)\mathcal{L}_{HuberLoss} \tag{7}$$

The first term of Eq. 7 is the IoU element which directly incorporates the IoU in the optimization process. The second term is the state-of-the-art Huber loss. The purpose of having the Huber loss is to make sure that the positional guidance can be made use when there is no overlap between the ground-truth and predicted bounding boxes, thus making gradient information available throughout the learning process. The two terms of the loss are linked by a scaling parameter λ. Naively, this term can be treated as a hyper-parameter that can be tuned for the best performance, however, doing so will be computationally expensive as well as time-consuming. Additionally, treating λ as a hyper-parameter will lead to having one λ for the entire retraining which was found to be detrimental to the overall performance. A mini-batch could have poor predictions and thus lead to bounding boxes with no overlap in which case the fixed value for λ would still try to make use of the non-existent gradients coming from the IoU element of Eq. 7. In order to prevent such outcomes, we propose to treat λ not as a hyper-parameter but rather scale it dynamically during training. λ is calculated

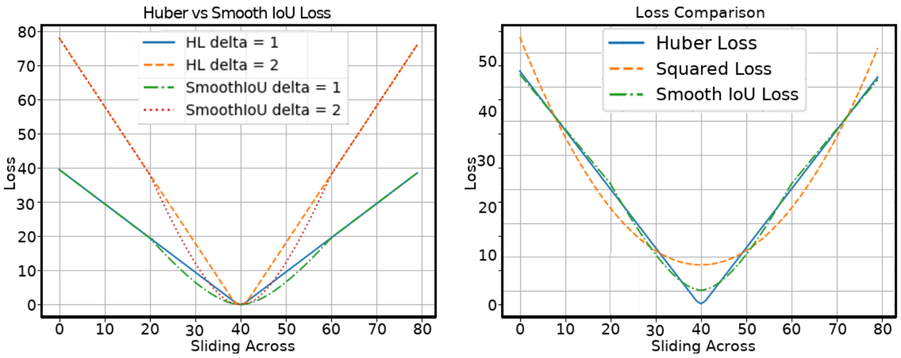

Fig. 2. (Right) Comparison of the Smooth IoU with Huber and Squared Loss, the losses have been scaled to highlight the profiles as they relates to the others. (Left) Behavior of Huber and Smooth IoU losses for varying values of δ. (see Eq. 2).

based on the mean IoU of the minibatch under evaluation and used in scaling the loss between IoU and state-of-the-art Huber loss. This dynamic scaling enables us to remove the need to tune λ and allows the model to learn End-to-End. This enables the model to be trained faster and without the need for a problem specific set of hyper-parameters.

The loss profile for the Smooth IoU loss is presented in Fig. 2 (left). In order to distinguish our contribution from that state-of-the-art Huber loss, we are presenting the behavior of the losses for the same example as seen in Fig. 1 where a predicted bounding box is translated over the ground truth bounding box. The δ term is the cutoff threshold from Eq. 2 and is varied from 1-2 in order to show that the behavior of the Smooth IoU loss is not just a scaled variation of the inherent Huber loss cutoff behavior. Fig. 2 (left) shows the behavior of the Smooth IoU loss and highlights how it is purely a function of the overlap of the boxes. From the figure, it can be seen that the Smooth IoU loss introduces a quadratic behavior to the loss as soon as the overlap starts to occur at the point $(20, 40)$ (see the description of the example in the introduction). This quadratic behavior appears much later in the state-of-the-art Huber loss, which is governed by the δ and is not dynamic but rather fixed for each run. Outside of the areas of intersection, we are still maintaining the Huber loss which allows the loss to overcome the inherent shortcomings of the IoU loss, and localize the boxes better.

The Smooth IoU loss comes with a performance guarantee of the state-of-the-art in the worst case scenario i-e. the loss converges to the state-of-the-art performance if the modifications suggested in this method do not improve the bounding box localization. This can be shown as:

$$\lim_{\lambda \to 0} \mathcal{L}_{SmoothIoU} = \mathcal{L}_{HuberLoss} \tag{8}$$

$$\lim_{\lambda \to 1} \mathcal{L}_{SmoothIoU} = \mathcal{L}_{IoU} \tag{9}$$

```
Algorithm 1: Smooth IoU Loss
   Data: predicted_boxes $P := \{p \in \mathbb{R}^{K\times 4} \mid p = \{y_1, x_1, h, w\}\}$ , target_boxes
         $T := \{t \in \mathbb{R}^{K\times 4} \mid t = \{\bar{y}_1, \bar{x}_1, \bar{h}, \bar{w}\}\}$
   Result: Smooth IoU Loss
 1 for $k$ = 1, . . . ,$K$ do
 2     P′= Transform ($P_k$);
 3     T′= Transform ($T_k$);
 4     $\mathcal{L}_{Huber_k}$ = Huberloss ($P_k{}', T_k{}'$);
 5     $IoU_k$ = Calculate_IoU($P_k{}', T_k{}'$);
 6     $\mathcal{L}_{IoU_k}$ = 1 - $IoU_k$;
 7 end
 8 $\lambda$ = mean($IoU_{1:K}$);
 9 for $k$ = 1, . . . , $K$ do
10     $loss_{SmoothIoU_k}$ = $\lambda \times \mathcal{L}IoU_k$ + (1- $\lambda$)$\mathcal{L}_{Huber_k}$;
11 end
```

This further highlights that the loss presented here is guaranteed to perform at the state-of-the-art level in the worst case scenario, making it a robust version of the Huber loss while introducing the IoU information into the optimization process. Additionally, this loss can be readily substituted into the current two-stage models without any architectural changes or case-specific modifications, making it modular.

Algorithm 1. shows the implemenetation of the Smooth IoU loss, lines 2-3 are used to transform the incoming points from $\{y_1, x_1, h, w\}$ representation to a $\{x_{min}, y_{min}, x_{max}, y_{max}\}$ representation. This is done in order to calculate the IoU of the bounding boxes. Line 10 shows the implementation of Eq. 7.

For the sake of completeness, the same example as seen in Fig. 1 is reproduced but this time with an introduced size mismatch between the prediction and

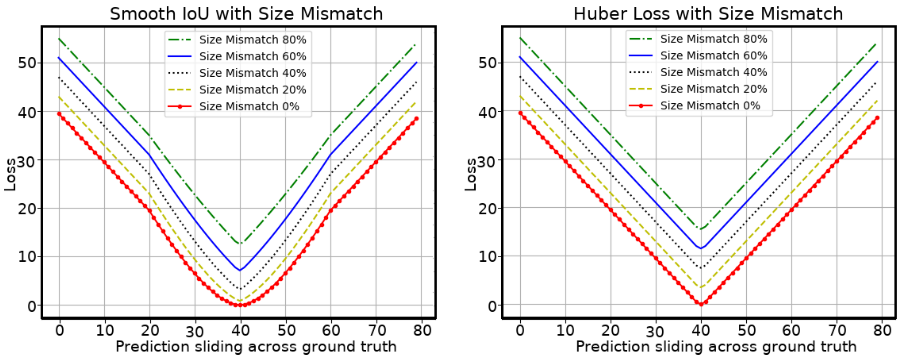

Fig. 3. Comparison of the losses with a size mismatch between the predicted and ground truth bounding boxes.

		Initial LR	**Proposals**	**Drop-out**
Pets	Smooth	0.0002	300	[0.2, ... , 0.8]
	Huber	0.0002	300	[0.2, ... , 0.8]
Udacity	Smooth	0.002	300	[0.2, ... , 0.8]
	Huber	0.002	300	[0.2, ... , 0.8]
VWFS	Smooth	0.0002	300	[0.2, ... , 0.8]
	Huber	0.0002	300	[0.2, ... , 0.8]
PASCAL	Smooth	0.0002	300	[0.2, ... , 0.8]
	Huber	0.0002	300	[0.2, ... , 0.8]

Table 1. Hyper-parameters used for the different datasets

ground truth bounding boxes. The loss profiles for both the Smooth IoU loss and Huber loss are shown in Fig. 3. The overall effect visible here is that even for the size mismatch between the two boxes the Smooth IoU loss tends to converge to a smaller loss value while leveraging the advantages of the Huber loss for non-overlapping regions. This smaller loss value will prevent the possibility of exploding gradients thus stabilizing the learning process.

4 Experiments

Having laid out the design of our new loss and its characteristics. We evaluated the performance on the object detection task using the Oxford-IIIT Pet Dataset [13], Udacity Self-Driving Car Dataset [24], PASCAL VOC [4], and Volkswagon Financial Services (VWFS) Damage Assessment dataset (propriety). The VWFS dataset is made up of images taken during an end-of-leasing damage assessment. The areas of interest in these images are the damaged parts that have been loosely annotated. The aim of this data is to serve as a foundation for training a model which is able to detect these damages automatically and estimate the damage costs at the end of the car lease period. The data has a very high variance in the number of images per class and suffers and from a long tail distribution. All experiments were conducted using Nvidia GTX 1080Ti, and Tesla P100 GPUs.

The experiments conducted herewith focused on Faster-RCNN from Tensorflow Object Detection API [6] but Smooth IoU loss can be used with any two-stage model. The reported results are with the hyper-parameters reported in Tab. 1 tuned to the best performance for the respective loss and to reproduce the numbers reported in the original paper [16]. For the minimization of the loss we have used RMSProp [23], with a learning rate reduced by 10^{-1} every 50K steps and a momentum term of 0.9. The underlying feature extractor was Inception˙V2 [7], pre-trained on the COCO dataset [8]. This warm start enabled us to speed up training by leveraging the advantages offered by transfer learning and as it has been shown to be an effective method for initializing the network [5] [16] [6]. The baseline implementation [16] used VGG-16 [20] pre-trained on ImageNet [18] as the feature extractor. In all of our experiments, the models were retrained for 200k iterations and showed a smooth convergence behavior.

	Pets		**UD**		**VW**		**2007**		**2012**		**VOC++**	
$\mathcal{L}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$	$\mathcal{L}_{HL}$	$\mathcal{L}_{S\_IoU}$
IoU	0.143	**0.162**	0.417	**0.425**	0.383	**0.388**	0.231	**0.263**	**0.275**	0.270	0.240	**0.244**

Table 2. Localization Metrics: These highlight the results for the datasets under test. **VW** = VWFS, **UD** = Udacity, **2007** = VOC2007, **2012** = VOC2012, **VOC++** = VOC++Train. The IoU metric directly is reported here to show the quality of the bounding boxes. $\mathcal{L}_{S\_IoU}$ is optimized for the IoU directly. $\mathcal{L}_{HL}$ is the baseline Huber loss.

For the Udacity, and Pets datasets we have used a standard train/test split. For the VWFS dataset, a 80-20 split was created while for the PASCAL VOC dataset we have used the VOC2007 Train split for training the models and its performance is presented on the VOC2007 val split. VOC2012 was treated the same way. We have also used a merged PASCAL VOC dataset where the VOC++Train was created by merging the VOC2007 and VOC2012 train splits for training and we are presenting the results on the VOC2012 val dataset. Since the ground truth bounding box information was not available on the test set of PASCAL VOC datasets, we present the results on the validation set.

4.1 Evaluation and Discussion

We propose a new loss for the bounding box localization that takes into account the direct optimization of the IoU in order to improve the quality of the predicted bounding boxes. This localization loss ties in closely with the overall performance of the two-stage networks. Therefore, it is important to evaluate the IoU quality of the model as well as the accuracies in order to showcase the effectiveness of our method. Detection accuracies (mean average precision and average recall) measure the correct classification of the objects and do not directly take into account the quality of the predicted bounding boxes. This is because mAP scores are calculated for a subset of the predicted bounding boxes (that fall above a threshold of IoU). We want to demonstrate how the optimization of the IoU directly for bounding box localization helps in the overall learning process while simultaneously improving the accuracy over the state-of-the-art by proposing better bounding boxes. Furthermore, we would also like to showcase the robustness of the loss over varying levels of difficulty of the object detection tasks, hence the choice of datasets that range from low difficult (Pets) to high difficulty (PASCAL). As stated earlier, this loss comes with a performance guarantee of the state-of-the-art performance and that is shown in the results that follow. Faster RCNN with the standard Huber loss (for bounding box localization) becomes the baseline.

We break down the evaluation of the model into localization and classification performance. For the classification performance, we are using COCO detection metrics as they are readily available in the API and are also a favored metric for the object detection task. Localization is the primary focus here since we are optimizing it directly for the IoU. Tab. 2 presents the comparison of our

Dataset		mAP @.50IoU*	mAP @.75IoU**	mAP***	AR@1	AR@10
Wins	$\mathcal{L}_{HL}$	0	3	2	2	1
	$\mathcal{L}_{S\_IoU}$	**6**	3	**4**	**4**	**5**
Pets	$\mathcal{L}_{HL}$	89.94	80.46	66.03	75.59	76.63
	$\mathcal{L}_{S\_IoU}$	**93.91**	**87.19**	**72.09**	**79.68**	**80.35**
VW	$\mathcal{L}_{HL}$	64.3	37.01	37.03	**21.0**	45.42
	$\mathcal{L}_{S\_IoU}$	**64.79**	**37.33**	**37.4**	20.94	**45.63**
UD	$\mathcal{L}_{HL}$	78.23	**27.67**	**36.22**	37.14	50.68
	$\mathcal{L}_{S\_IoU}$	**78.36,**	27.65	35.7	**37.15**	**51.13**
2007	$\mathcal{L}_{HL}$	65.94	**43.40**	**40.68**	**37.77**	**57.14**
	$\mathcal{L}_{S\_IoU}$	**66.29**	43.01	40.67	37.32	56.79
2012	$\mathcal{L}_{HL}$	69.14	**49.09**	44.20	39.63	59.82
	$\mathcal{L}_{S\_IoU}$	**69.31**	48.29	**44.41**	**40.26**	**60.04**
VOC++	$\mathcal{L}_{HL}$	70.47	49.68	44.19	39.81	59.39
	$\mathcal{L}_{S\_IoU}$	**70.52**	**50.08**	**45.17**	**39.96**	**59.58**

Table 3. Classification Metrics: *mAP@.50IoU is the PASCAL metric as set out in COCO Detections metrics, takes into account bounding boxeswith a 50% overlap with the ground-truth. Similarly, **mAP@.75IoU takes into account 75% overlap. ***mAP takes into account overlap of 50% and higher. Average Recall (@1 and @10).

proposed method against Huber loss for the different datasets. We are reporting the value of $IoU \in [0, 1]$, the values in bold show where our method is better than the baseline. We are evaluating the quality of the IoU against the ground-truth bounding boxes (higher value is better). The reported numbers here are all proposed boxes by the model, we are not discounting any boxes through post-processing, hence the values appear to be small. This is done to see the raw behavior of the model for the baseline and the proposed method. For the localization, it can be seen in Tab. 2 that our method outperforms the baseline in five out of six datasets that are under consideration. The results show that by optimizing for the IoU directly leads to better bounding boxes. Furthermore, the robustness of the loss is also verified by looking at the results for VOC2012 in Tab. 3. We underperform the baseline on the VOC2012 dataset in terms of the IoU however when we look at the overall performance for the classification Tab. 3, we can see that for the VOC2012 dataset our method is still better than the baseline. This indicates that the Smooth IoU loss can be used for directly optimizing the IoU and will not harm the overall performance in cases where it does not directly improve the IoU.

For the classification metrics in Tab. 3, the first row provides the total win/loss count for our proposed method and the baseline. As can be seen, our method outperforms the baseline for mAP@.50IOU, mAP, AR@1, and AR@10. We tie with the baseline for mAP@.75IOU. It should be noted here that we are not modifying the baseline classification optimization and show how directly optimizing the IoU can lead to gains in the mAP and recall as well. This lends credence to our proposed method shows it's usefulness. All the results highlight that there is a signal available in the IoU information of the bounding boxes and this information should be used during training. We experimentally showcase that the IoU driven loss variant proposed herewith can outperform the standard loss and it is lower-bounded to be at the state of the art level.

5 Conclusion

In this paper, we have presented a novel loss for the bounding box localization of two-stage models. The loss optimizes the IoU directly by treating the parameters of a bounding box as a single highly correlated item. Our loss is lower-bounded to perform at the state-of-the-art level. We demonstrate the efficacy of our model by replacing the Huber loss in Faster RCNN to show that optimizing for IoU directly in bounding box localization can lead to better bounding boxes and also improve the classification accuracy. Our method has shown to outperform the baseline in both localization and classification metrics. The modular and robust nature of the proposed loss makes it readily compatible with all two-stage models.

Acknowledgments

This work is co-funded by the industry project ”Data-driven Mobility Services” of ISMLL and Volkswagen Financial Services.(`https://www.ismll.uni-hildesheim.de/projekte/dna_en.html`)

References

1. Box, G., Hunter, J.: Annals of mathematical statistics. The Annals of Mathematical Statistics **27**(4), 1144–1151 (1956)
2. Caruana, R.: Multitask learning. Machine learning **28**(1), 41–75 (1997)
3. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2147–2154 (2014)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
5. Girshick, R.: Fast r-cnn. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
6. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7310–7311 (2017)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
8. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)

9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
11. Nedic, A., Bertsekas, D.P.: Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization **12**(1), 109–138 (2001)
12. Nowozin, S.: Optimal decisions from probabilistic models: the intersection-over-union case. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 548–555 (2014)
13. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
14. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: International symposium on visual computing. pp. 234–244. Springer (2016)
15. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
17. Rosasco, L., Vito, E.D., Caponnetto, A., Piana, M., Verri, A.: Are loss functions all the same? Neural Computation **16**(5), 1063–1076 (2004)
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
19. Schaul, T., Zhang, S., LeCun, Y.: No more pesky learning rates. In: International Conference on Machine Learning. pp. 343–351 (2013)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Sutskever, I., Hinton, G.E., Krizhevsky, A.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems pp. 1097–1105 (2012)
22. Szegedy, C., Reed, S., Erhan, D., Anguelov, D., Ioffe, S.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 (2014)
23. Tieleman, T., Hinton, G.: Rmsprop gradient optimization. URL http://www. cs. toronto. edu/tijmen/csc321/slides/lecture_slides_lec6. pdf (2014)
24. Udacity: Self driving car. `https://github.com/udacity/self-driving-car/tree/master/annotations` (2017)
25. Xu, Y., Lin, Q., Yang, T.: Stochastic convex optimization: Faster local growth implies faster global convergence. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3821–3830. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017), `http://proceedings.mlr.press/v70/xu17a.html`
26. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.S.: Unitbox: An advanced object detection network. CoRR **abs/1608.01471** (2016), `http://arxiv.org/abs/1608.01471`