Mask-aware IoU for Anchor Assignment in Real-time Instance Segmentation

Kemal Oksuz[†], Baris Can Cam[†],
Fehmi Kahraman, Zeynep Sonat Baltaci,
Sinan Kalkan[‡], Emre Akbas[‡]

Dept. of Computer Engineering
Middle East Technical University
Ankara, Turkey

{kemal.oksuz,can.cam,fehmi.kahraman_01,sonat.baltaci,skalkan,eakbas}@metu.edu.tr

Abstract

This paper presents Mask-aware Intersection-over-Union (maIoU) for assigning anchor boxes as positives and negatives during training of instance segmentation methods. Unlike conventional IoU or its variants, which only considers the proximity of two boxes; maIoU consistently measures the proximity of an anchor box with not only a ground truth box but also its associated ground truth mask. Thus, additionally considering the mask, which, in fact, represents the shape of the object, maIoU enables a more accurate supervision during training. We present the effectiveness of maIoU on a state-of-the-art (SOTA) assigner, ATSS, by replacing IoU operation by our maIoU and training YOLACT, a SOTA real-time instance segmentation method. Using ATSS with maIoU consistently outperforms (i) ATSS with IoU by ~ 1 mask AP, (ii) baseline YOLACT with fixed IoU threshold assigner by ~ 2 mask AP over different image sizes and (iii) decreases the inference time by 25% owing to using less anchors. Then, exploiting this efficiency, we devise maYOLACT, a faster and +6 AP more accurate detector than YOLACT. Our best model achieves 37.7 mask AP at 25 fps on COCO test-dev establishing a new state-of-the-art for real-time instance segmentation. Code is available at https://github.com/kemaloksuz/Mask-aware-IoU.

1 Introduction

Instance segmentation is a visual detection problem which aims to classify and locate each object in an image by pixel-level masks. To be able to handle objects of different numbers, locations and scales; SOTA methods [3, 14, 35] employ a dense set of object hypotheses, generally represented by boxes or points, and ensure a maximum coverage of the objects. This coverage necessitates a large number of object hypotheses ($\sim 20k$ per image in YOLACT [3] for images of size 550×550) to be assigned to ground truths boxes; generally known as the *assignment problem* [17, 42].

This assignment problem is commonly tackled by employing heuristic rules. One common rule to assign object hypotheses represented by boxes (i.e. *anchors*) is using a "fixed IoU threshold" [3, 11, 14], in which an anchor, $\hat{B}$, can be assigned with a ground truth (i.e positive), B, when their Intersection-over-Union (IoU), defined as $\mathrm{IoU}(\hat{B}, B) = |\hat{B} \cap B| / |\hat{B} \cup B|$, exceeds a pre-determined threshold, τ. If anchor $\hat{B}$ cannot be assigned to any ground-truth

[†] Equal contribution, [‡] Equal contribution for senior authorship

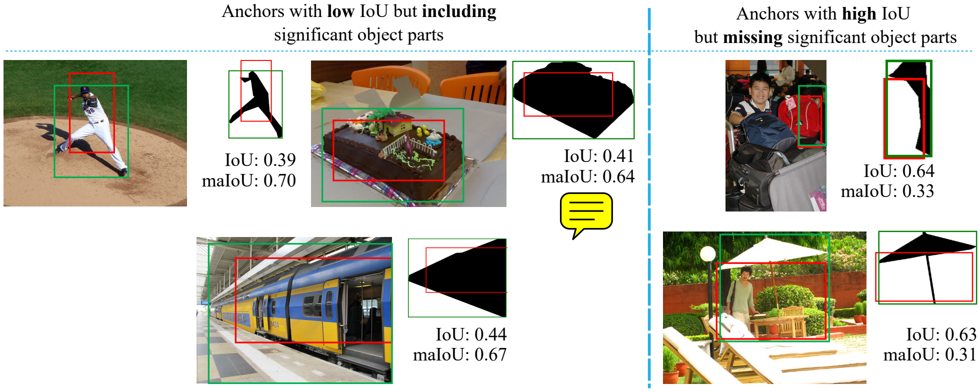

Anchors with **low** IoU but **including** significant object parts
Anchors with **high** IoU but **missing** significant object parts
IoU: 0.39
maIoU: 0.70
IoU: 0.41
maIoU: 0.64
IoU: 0.44
maIoU: 0.67
IoU: 0.64
maIoU: 0.33
IoU: 0.63
maIoU: 0.31

Figure 1: Sample cases illustrating the need for mask-aware IoU (maIoU). Green boxes denote ground truth, red boxes are real anchors produced during the training. Left panel shows cases where the anchor covers a significant part of the object pixels but IoU is low (i.e. less than positive threshold of 0.50 for YOLACT). maIoU is higher than IoU for these cases, potentially correcting the assignment. Right panel shows cases where the anchor covers only a small part of the object pixels but IoU is high (so, anchors are positive). maIoU is lower than IoU, potentially correcting the assignment. Images are from COCO [19].

B (i.e. $\text{IoU}(\hat{B},B) < \tau,\ \forall B$), then $\hat{B}$ is assumed to be a background (i.e. negative) example. A different set of recent methods showed for object detection [17, 18, 42] that assigning the anchors using an “adaptive IoU threshold” determined for each ground truth improves the performance. Still, assignment methods heavily rely on IoU as the de facto proximity measure between ground truths and anchors.

Despite its popularity, IoU has a certain drawback: The IoU between an anchor box and a ground truth box solely depends on their areas, thereby ignoring the shape of the object, e.g. as provided by a segmentation mask. This may give rise to undesirable assignments due to counter-intuitively lower or higher IoU scores. For example, the IoU might be high, implying a positive anchor, but only a small part of the object is included in the anchor; or the IoU may be low, implying a negative anchor, but a large part of the object is included in the anchor. Fig. 1 presents examples for such cases, arising due to objects with unconventional poses, occlusion and objects with articulated or thin parts. We will show (in Section 3.1, Fig. 3) that such examples tend to produce larger loss values and adversely affect training.

In this paper, we introduce mask-aware IoU (maIoU), a novel IoU measure for anchor assignment in instance segmentation by exploiting the ground truth masks of the objects, normally used only for supervision. Specifically, unlike IoU, which equally weights all pixels, maIoU yields a proximity measure between 0 and 1 among an anchor box, a ground truth box and a ground truth mask by promoting the pixels on the masks, thereby providing a more consistent assignment (Fig. 1). Since a naive computation of maIoU is impractical, we present an efficient algorithm with training time similar to the baseline. YOLACT with maIoU-based ATSS assigner consistently improves ATSS assigner with IoU by ~ 1 mask AP and standard YOLACT (i.e. fixed IoU threshold) by ~ 2 mask AP, and also decreases inference time of YOLACT. Finally, utilizing this efficiency gap, we build maYOLACT detector reaching 37.7 mask AP at 25 fps and outperforming all real-time counterparts.

Table 1: IoU Variants, their inputs and primary purposes. IoU variants assign a proximity measure based on the properties (prop.) of two inputs (Input 1 and Input 2). In practice, existing variants compare the inputs wrt. the same properties (i.e. either boxes or masks). Our Mask-aware IoU (maIoU) can uniquely compare a box with a box and a mask. With this, maIoU compares anchors (i.e. only box) with ground truths (box and mask) in order to provide better anchor assignment. *: GIoU is also used as a performance measure.

IoU Variant	Input 1 prop.		Input 2 prop.		Primary purpose as proposed in the paper
	Box	Mask	Box	Mask	
Mask IoU [8, 19]	✗	✓	✗	✓	Performance measure
Boundary IoU [7]	✗	✓	✗	✓	Performance measure
Generalized IoU [29]	✓	✗	✓	✗	Loss function*
Distance IoU [43]	✓	✗	✓	✗	Loss function
Complete IoU [44]	✓	✗	✓	✗	Loss function
Mask-aware IoU (Ours)	✓	✓	✓	✗	*Assigner*

2 Related Work

Deep Instance Segmentation. In general, deep instance segmentation methods have followed object detection literature. The pioneering Mask R-CNN model [14] and its variations [15, 26] extended Faster R-CNN [28] by incorporating a mask prediction branch into the two-stage detection pipeline. Similarly, anchor-based one-stage methods were also adapted for instance segmentation by using an additional mask prediction branch, e.g. YOLACT [3] and YOLACT++ [4] employed a YOLO-like architecture; and PolarMask [39] and PolarMask++ [40], both anchor-free methods, adapted FCOS [30] for instance segmentation. Differently, SOLO variants [34, 35] classify the pixels based on location and size of each instance.

Anchor Assignment in Instance Segmentation. Deep instance segmentation methods using anchors as object hypotheses label anchors based on "fixed IoU threshold" assignment rule: The anchors with IoU larger than τ^+ with a ground truth box are assigned as positive; while the anchors whose maximum IoU with ground truths is less than τ^- are assigned as negatives; and the remaining anchors whose maximum IoU is between τ^- and τ^+ are simply ignored during training. To illustrate, YOLACT variants [3, 4] and RetinaMask [11] use $\tau^- = 0.40$ and $\tau^+ = 0.50$; while the first stage of Mask R-CNN (i.e. region proposal network [28]) sets $\tau^- = 0.30$ and $\tau^+ = 0.70$; and finally its second stage [14] uses $\tau^- = \tau^+ = 0.50$.

Adaptive Anchor Assignment Methods in Object Detection. Recently, adaptive anchor assignment strategies are shown to perform better than fixed IoU threshold in object detection: ATSS [42] uses top-k anchors wrt. IoU to determine an adaptive IoU threshold for each ground truth (Section 3.4 provides more details on ATSS.) and PAA [17] computes a score of each anchor including Generalized IoU and fits the distribution of these scores to a two-dimensional Gaussian Mixture Model to split positives and negatives for each ground truth. Similarly, Ke et al. [16] and Li et al. [18] identify positives and negatives by using different scoring functions of the predictions. However, these methods are devised and tested for object detection, and thus, do not utilize object masks.

Other IoU Variants. Over the years, many IoU variants have been proposed – see Table 1 for a comparative summary. One of the most related IoU variants is Mask IoU, which is used to measure the detection mask quality with respect to (wrt.) the ground truth mask during evaluation [8, 19]. Similarly, Boundary IoU [7] evaluates detection masks by giving higher weights to the pixels closer to the boundaries. Note that these IoU variants compare

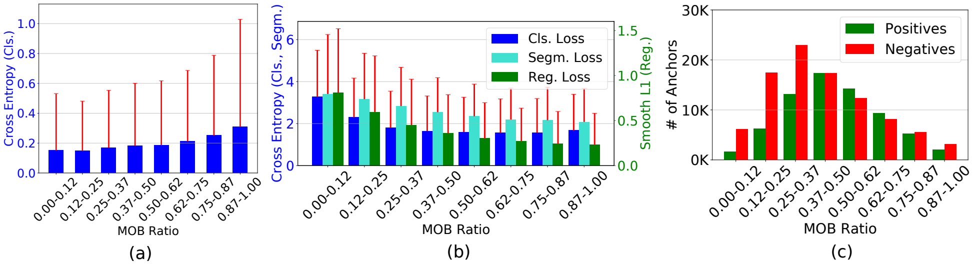


Figure 2: **(a,b)** Mean and standard deviation of loss values of negative anchors (a) and positive (b) anchors with **similar IoUs** (IoU between $[0.30-0.50]$ for negatives and $[0.50-0.70]$ for positives) over different MOB ratios for a **trained** YOLACT on COCO *minival*. Red lines denote the standard deviation. Note that when the MOB ratio increases, the loss values increase for negatives; however, the loss values of all three sub-tasks (Cls.: classification, Segm.: segmentation, Reg.: regression) tend to decrease for positives. **(c)** The number of anchors for each MOB ratio is in the order of thousands.

only two masks, and unlike our maIoU, they cannot compare a box with another box and its associated mask. The other IoU variants are all devised to measure the proximity of two boxes: Generalized IoU (GIoU) [29] uses the minimum enclosing box in order to measure the proximity of boxes when boxes do not intersect (i.e. their IoU is 0); Distance IoU [43] adds a penalty parameter based on the minimum enclosing box and the distance between the centers of boxes; Complete IoU [44] additionally considers aspect ratio differences of the boxes. These IoU-variants compute the overlap at the box level and neglect object shape; and also, they are mainly used as loss functions, not as a positive-negative assignment criterion.

Comparative Summary. By measuring the proximity of an anchor box with a ground truth, our maIoU is designed for anchor-based models, which have been using a fixed IoU threshold for assigning anchors as the dominant approach, thereby ignoring object shape. We first show that ATSS [42], an adaptive anchor assigner, yields better performance on YOLACT [3]. Then, we propose maIoU, as the first IoU variant that can measure the proximity of an anchor box with a ground truth box and ground truth mask (Table 1). Replacing IoU of ATSS by our maIoU improves the performance of this strong baseline. We also investigate GIoU and DIoU for anchor assignment. Since they rely only on boxes, our maIoU provides more discriminative information then these IoU variants. Finally, besides our maIoU, adopting recently proposed improvements into YOLACT detector, we build maYOLACT detector, which outperforms its counterparts while being more efficient as well (Section 4).

3 Methodology

This section first presents an analysis on fixed-threshold IoU assigner in Section 3.1. Then, Section 3.2 defines maIoU and Section 3.3 provides an efficient algorithm to compute maIoU. Finally, Section 3.4 incorporates our maIoU into the SOTA ATSS assigner [42] to label anchors during the training of instance segmentation methods.

3.1 The Mask-over-box Ratio

Our analysis is based on an intuitive measure to represent the rate of the ground truth mask in a box, defined as the Mask-over-box (MOB) ratio as follows.

Definition 1. *Mask-over-box (MOB) ratio of a box (i.e. anchor or ground truth), $\bar{B}$, on a ground truth mask, M, is the ratio of (i) the area of the intersection of the mask and the box, and (ii) the area of $\bar{B}$ itself:*

$$\mathrm{MOB}(\bar{B},M) = \frac{|\bar{B} \cap M|}{|\bar{B}|}. \tag{1}$$

$\mathrm{MOB}(\bar{B},M) \in [0,1]$ and less object pixels from M in $\bar{B}$ implies a lower MOB ratio. When $\bar{B}$ is the ground truth box of M (i.e. $\bar{B} = B$), $|M \cap B| = |M|$, and thus $\mathrm{MOB}(B,M) = |M|/|B|$.

We now analyse how losses on a trained standard YOLACT model (with ResNet-50 backbone), a SOTA instance segmentation method, and the number of anchors change wrt. MOB ratio (Fig. 2) and make the following crucial observations:

(1) The loss value (i.e. error) of a trained YOLACT for an anchor is related to the amount of mask pixels covered by that anchor (i.e. MOB ratio), which is ignored by the standard fixed IoU threshold assigner. Fig. 2(a,b) present that the average and standard deviation of the loss values of anchors that are close to the IoU assignment threshold (i.e. the anchors with IoU between $[0.30-0.50]$ for negatives and $[0.50-0.70]$ for positives; hence, anchors with similar IoUs) *increase in all tasks* (for negatives, it is only classification; for positives, we look at all tasks – i.e. classification, box regression and segmentation) as MOB ratio decreases/increases (i.e. implying covering less/more on-mask pixels) for positive/negative anchors. Also, the numbers of anchors with larger losses are in the order of thousands in all cases (Fig. 2(c)).

(2) Similar to anchors, MOB ratios of the ground truth boxes also vary significantly. We observe in Fig. 3(a) that there exists a significant amount of ground truth boxes with low MOB ratios (i.e. for 30 % of the ground truths, MOB ratio is less than 0.50.

Our observations suggest that IoU does not discriminate anchors with more on-mask pixels from those with less on-mask pixels, which appears naturally due to varying MOB ratios of the ground truths; and thus using IoU for the assignment of the anchors may not be the best method.

3.2 Mask-aware Intersection-over-Union (maIoU)

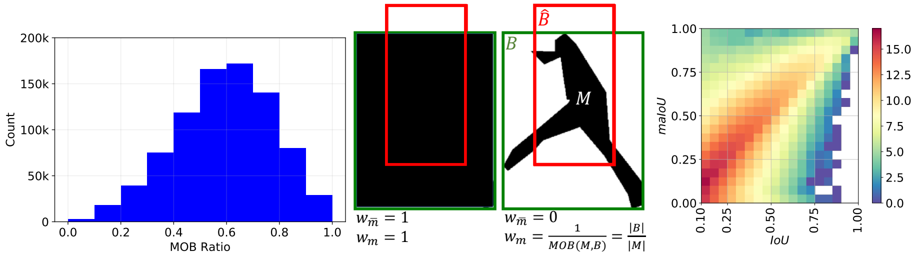
Intuition. The main intuition behind mask-aware IoU (maIoU) is to reweigh the pixels within the ground truth box, B, such that on-mask pixels are promoted (in a way, the contribution of off-mask pixels are reduced) by preserving the total *energy* of B (i.e. $|B|$). We simply achieve this by distributing the contributions of the off-mask pixels uniformly over the on-mask pixels in B. Finally, maIoU is computed as an intersection-over-union between $\hat{B}$ (i.e. anchor box) and B with the new pixel weights in B (Fig. 3(b)).

Derivation. To facilitate derivation, we first reformulate intersection $\mathcal{I}$ between B and $\hat{B}$ in a weighted form as follows ($w_m, w_{\overline{m}}$: the contributions of on-mask and off-mask pixels respectively):

$$\mathcal{I}(B,\hat{B}) = \sum_{i \,\in\, B\cap\hat{B}} w = \sum_{i \,\in\, \hat{B}\cap M} w_m + \sum_{i \,\in\, \left(\hat{B}\cap B - \hat{B}\cap M\right)} w_{\overline{m}}, \tag{2}$$

which effectively does not make use of the mask M since $w = w_m = w_{\overline{m}} = 1$ for IoU.

In maIoU, we discard the contribution of an off-mask pixel: $w_{\overline{m}} = 0$, and in order to preserve the total energy, the reduced contribution from all off-mask pixels, which equals $|B| - |M|$, is distributed to the on-mask pixels uniformly. This will increase w_m by $(|B| - |M|)/|M|$: $w_m = 1 + (|B| - |M|)/|M| = 1 + |B|/|M| - |M|/|M| = |B|/|M| = 1/\mathrm{MOB}(B,M)$.



(a) (b) (c)

Figure 3: (a) The distribution of MOB ratios of the ground truths on COCO training set. (b) How IoU and maIoU weights the pixels in ground truth box (see Fig. 1 top-left example for the image of this example). While IoU does not differentiate among on-mask (w_m) and off-mask ($w_{\overline{m}}$) pixels, our maIoU sets $w_{\overline{m}} = 0$ and weights each on-mask pixel by inverse MOB ratio considering object mask M. (c) Anchor count distribution (in log-scale) of IoU vs maIoU. While IoU & high-maIoU positively correlate, there are quite a few examples with low-IoU & high-maIoU and vice versa.

With these weights, the mask-aware intersection, ma$\mathcal{I}$ is defined by extending Eq. 2:

$$\mathrm{ma}\mathcal{I}(\hat{B},B,M) = \sum_{i \,\in\, \hat{B}\cap M} w_m + \sum_{\cancel{i \,\in\, (\hat{B}\cap B - \hat{B}\cap M)}} \cancelto{0}{w_{\overline{m}}} = w_m|\hat{B}\cap M| = \frac{1}{\mathrm{MOB}(B,M)}|\hat{B}\cap M|. \quad (3)$$

Extending the definition of union (i.e. $|B|+|\hat{B}|-|B\cap\hat{B}|$) with this intersection concept:

$$\mathrm{ma}\mathcal{U}(\hat{B},B,M) = |B| + (|\hat{B}| - |\hat{B}\cap B| + \mathrm{ma}\mathcal{I}(\hat{B},B,M)) - \mathrm{ma}\mathcal{I}(\hat{B},B,M) = |\hat{B}\cup B|, \quad (4)$$

which is equal to the conventional union. This is not surprising since our formulation preserves ground truth area (i.e. $|B|$). With the updated definitions, mask-aware IoU is simply mask-aware intersection over mask-aware union:

$$\mathrm{maIoU}(\hat{B},B,M) = \frac{\mathrm{ma}\mathcal{I}(\hat{B},B,M)}{\mathrm{ma}\mathcal{U}(\hat{B},B,M)} = \frac{1}{\mathrm{MOB}(B,M)} \frac{|\hat{B}\cap M|}{|\hat{B}\cup B|}, \quad (5)$$

which, in effect, is the ratio of covered on-mask pixels by the anchor ($|\hat{B}\cap M|$) in the union of boxes ($|\hat{B}\cup B|$), normalized by on-mask pixel density in B (i.e. $\mathrm{MOB}(B,M)$).

Interpretation. Similar to IoU, $\mathrm{maIoU}(\hat{B},B,M) \in [0,1]$ and a larger value implies a better localisation considering not only the boxes but also the ground truth mask (Fig. 3(b)). We visualize the anchor count distribution (in log-scale) of YOLACT on COCO minival on the space spanned by IoU and maIoU (Fig. 3(c)): While IoU and maIoU are positively correlated, there are quite a number of examples with low-IoU & high-maIoU and vice versa, hence the assignment rules based on maIoU is quite different than those based on IoU.

3.3 Computation of maIoU

Compared to IoU, computing maIoU involves two additional terms (Eq. 5): (i) $|M|$, the total number of mask-pixels (since calculating $|B|$ for $\mathrm{MOB}(B,M)$=$|M|/|B|$ is trivial), and (ii) $|\hat{B}\cap M|$, the number of mask-pixels in the intersection. While the masks are included in the datasets and computing these two quantities is straightforward; it is impractical to compute them naively (i.e. brute force, see Table 3) considering the large number of anchors covering the image. For this reason, we employ *integral images* [9] on the binary ground truth masks, M. More specifically, for M covering an image of size $m \times n$, we compute its integral image Θ^M, an $(m+1) \times (n+1)$ matrix that encodes the total number of mask pixels above and to the left of each pixel. Accordingly, denoting the $(i,j)^{th}$ element of Θ^M by $\Theta^M_{i,j}$, the last element of Θ^M stores $|M|$, i.e. $|M| = \Theta^M_{m+1,n+1}$. As for the second term, assuming $\hat{B}$ is represented by a top-left point (x_1,y_1) and a bottom-right point (x_2,y_2) such that $x_2 > x_1$ and $y_2 > y_1$, $|\hat{B} \cap M|$ involves only four look-up operations, i.e. $|\hat{B} \cap M| = \Theta^M_{x_2+1,y_2+1} + \Theta^M_{x_1,y_1} - \Theta^M_{x_2+1,y_1} - \Theta^M_{x_1,y_2+1}$. The overall algorithm to compute maIoU$(\hat{B},B,M)$ is presented in Alg. 1.

Algorithm 1 The algorithm for efficiently calculating mask-Aware IoU.

```
procedure MaskAwareIoU($\hat{B},B,M$)
    Compute $|B|$, $|\hat{B} \cup B|$ and $\Theta^M$ as integral image of $M$ such that $\Theta^M_{i,j}$ is $(i,j)^{th}$ element of $\Theta^M$
    Set $|M| = \Theta^M_{m+1,n+1}$ and $|\hat{B} \cap M| = \Theta^M_{x_2+1,y_2+1} + \Theta^M_{x_1,y_1} - \Theta^M_{x_2+1,y_1} - \Theta^M_{x_1,y_2+1}$.
    Compute MOB$(B,M) = |M|/|B|$ for ground truth $B$
    return maIoU$(\hat{B},B,M)$ (Eq. 5)
end procedure
```

3.4 Incorporating maIoU into ATSS Assigner

ATSS assigner [42] is a SOTA assignment method used for object detection, yielding better performance than a fixed-threshold IoU assigner and simplifying the anchor design by using a single anchor per pixel unlike its predecessors with up to nine anchors per pixel [21]. ATSS assigner comprises three steps: (i) selecting top-k (i.e. conventionally $k=9$) anchors ($\hat{B}$) wrt. the distance of the centers between B and $\hat{B}$ for each ground truth (B) on each FPN level as "candidates", (ii) filtering out the candidates using an adaptive IoU threshold, computed based on the statistics of these candidates for each B, and (iii) filtering out the candidates, whose centers lie out of B. The surviving candidates after steps (ii) and (iii) are the positive examples and the remaining examples are the negatives. Using our maIoU with ATSS (or any IoU-based assigner) is straightforward: In step (ii), we just replace IoU-based adaptive thresholding by maIoU-based adaptive thresholding.

4 Experiments

Dataset. We train all models on the COCO *trainval* set [19] (115K images), test them on the COCO *minival* set (5k images) unless otherwise stated.

Performance Measures. We mainly report AP-based performance metrics: COCO-style AP (AP, in short), APs where true positives are validated from IoUs of 0.50 and 0.75 (AP$_{50}$ and AP$_{75}$), and APs for small, medium and large objects (AP$_{\text{S}}$, AP$_{\text{M}}$ and AP$_{\text{L}}$ respectively). Furthermore, we also exploit the recent optimal Localisation Recall Precision (oLRP) Error [24, 25]. While AP is a higher-is-better measure, oLRP is a lower-is-better metric.

Implementation Details. We conduct our experiments on YOLACT [3], an anchor-based real-time instance segmentation method, using the mmdetection framework [8]. Following

Table 2: Comparison of different assigners and IoU variants on YOLACT. Considering the shapes of the objects, our ATSS w. our maIoU consistently outperforms its counterparts.

Scale	Assigner	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_S ↑	AP_M ↑	AP_L ↑	oLRP ↓
400	fixed IoU threshold	24.8	42.4	25.0	**7.3**	26.0	42.0	78.3
	ATSS w. IoU	25.3	43.5	25.5	6.8	27.3	43.8	77.7
	ATSS w. DIoU	25.4	43.6	25.2	7.2	27.1	43.4	77.7
	ATSS w. GIoU	25.1	42.7	25.3	7.0	26.8	41.8	78.0
	ATSS w. maIoU (Ours)	**26.1**	**44.3**	**26.3**	7.2	**28.0**	**44.3**	**77.1**
550	fixed IoU threshold	28.5	47.9	29.4	11.7	31.8	43.0	75.2
	ATSS w. IoU	29.3	49.2	30.2	11.1	33.0	44.5	74.5
	ATSS w. DIoU	29.5	49.5	30.1	11.7	33.2	44.9	74.4
	ATSS w. GIoU	29.1	48.6	30.0	**12.0**	32.2	43.3	74.7
	ATSS w. maIoU (Ours)	**30.4**	**50.3**	**31.4**	11.5	**33.9**	**46.3**	**73.7**
700	fixed IoU threshold	29.7	50.0	30.4	14.2	32.8	43.7	74.3
	ATSS w. IoU	30.8	51.8	31.2	14.1	35.0	44.0	73.3
	ATSS w. DIoU	30.9	51.9	31.7	14.0	35.4	44.0	73.3
	ATSS w. GIoU	30.1	50.7	31.0	14.0	33.8	43.1	74.0
	ATSS w. maIoU (Ours)	**31.8**	**52.8**	**32.8**	**14.7**	**35.6**	**45.7**	**72.5**

Zhang et al. [42], when we use ATSS assigner, we keep $k = 9$ (Section 3.4) and simplify the anchor configuration by placing a single anchor on each pixel with an aspect ratio of $1:1$ and a base scale of 4 unless otherwise stated. Also, with ATSS (with IoU, DIoU [43], GIoU [29] or our maIoU), we keep classification and box regression loss weights as they are (1.0 and 1.5 respectively), and increase mask prediction loss weight from 6.125 to 8. When we replace the assigner, note that it affects the examples in all branches (i.e. classification, box regression, semantic and instance mask predictions). We train all models with 32 images distributed on 4 GPUs (8 images/GPU). The remaining design choices of YOLACT [3] are kept. We adopt ResNet-50 [13] as the backbone and resize the images during training and inference to $S \times S$ where S can be either 400, 550 or 700 following Bolya et al. [3].

4.1 Ablation Experiments

In this section, we demonstrate that our maIoU improves upon other assigners based on IoU variants consistently and our Alg. 1 makes computation of maIoU feasible during training.

Using ATSS with IoU. We first replace the fixed IoU threshold assigner of YOLACT by ATSS with (w.) IoU and have a stronger baseline for our maIoU: ATSS w. IoU improves fixed IoU assigner by $0.5 - 1.1$ mask AP over different scales (Table 2).

Replacing IoU of ATSS with maIoU. Replacing IoU in ATSS by our maIoU (Table 2) (i) improves fixed IoU assigner by 1.3, 1.9 and 2.1 mask APs for 400, 550 and 700 scales respectively, (ii) outperforms ATSS w. IoU by ~ 1.0 mask AP in all scales, (iii) performs also better than other IoU variants (i.e. ATSS w. GIoU and ATSS w. DIoU). We note that the contribution of maIoU (i) on models trained by images with larger scales (700 vs. 400 in Table 2) and (ii) on larger objects (AP_L vs. AP_S) are more significant. This is intuitive since the shape of the object gets more precise as the object gets larger.

Computing maIoU Efficiently. Computing maIoU for every anchor-ground truth pair during training by brute force is infeasible, i.e. it would take ~ 3 months to train a single model with 41.89 sec/iteration (Table 3). Using Alg. 1, we reduce the average iteration time by

$\sim 70\times$ to 0.59 sec/iteration, which is similar to other standard assigners (Table 3).

Table 3: Avg. iteration time (t) of assigners. While brute force maIoU computation is inefficient (Alg. 1 is ✗); our Alg. 1 decreases t by $\sim 70\times$ and has similar t with Fixed IoU Thr. and ATSS w. IoU.

Assigner	Alg. 1	t (sec.)
Fixed IoU Thr.	N/A	0.51
ATSS w. IoU	N/A	0.57
ATSS w. maIoU	✗	41.89
ATSS w. maIoU	✓	0.59

Table 4: ATSS w. maIoU (underlined) makes YOLACT more accurate and $\sim 25\%$ faster mainly owing to less anchors. Thanks to this efficiency, we build maYOLACT-550 with 34.8 AP and still larger fps than YOLACT.

	Method	AP	AP^{box}	fps	Anchor #
maYOLACT-550	YOLACT-550	28.5	30.7	27	$\sim 19.2K$
	+ ATSS w. maIoU	30.4	32.5	**33**	$\sim$ **6.4K**
	+ Carafe FPN [32]	31.4	33.3	32	$\sim$ **6.4K**
	+ DCNv2 [46]	33.2	35.8	31	$\sim$ **6.4K**
	+ more anchors	33.5	36.3	30	$\sim 12.8K$
	+ cosine annealing [22]	34.8	37.9	30	$\sim 12.8K$

4.2 maYOLACT Detector: Faster and Stronger

Thanks to using fewer number of anchors, YOLACT trained with our ATSS w. maIoU assigner (underlined in Table 4) is $\sim 25\%$ faster than baseline YOLACT (33 vs. 27 fps[1]), pointing out the importance of anchor design for the efficiency of real-time systems as well[2]. Exploiting this run-time gap; our aim in this section is to extend the standard YOLACT using our maIoU and the recent improvements in order to make it competitive with the recent methods also by keeping the resulting detector to process images in real-time[3]. To achieve that, we use (i) carafe [32] as the upsampling operation of FPN [20], (ii) deformable convolutions [46] in the backbone, (ii) two anchors with base scales 4 and 8 on each pixel, and (iv) cosine annealing with an initial learning rate of 0.008 by replacing the step learning rate decay. Effect of these improvements are presented in Table 4 and the resulting detector with these improvements is coined as *maYOLACT*. Note that our maYOLACT-550 detector is still faster than baseline YOLACT-550 and improves it by +6.3 mask AP and +7.2 box AP reaching 34.8 mask AP and 37.9 box AP (Table 4).

4.3 Comparison with State-of-the-art (SOTA)

Table 5 compares our maYOLACT with state-of-the-art methods on COCO *test-dev*.
Comparison with YOLACT variants. Achieving 35.2 mask AP, our maYOLACT-550 outperforms all YOLACT variants including the ones with larger backbones (e.g. YOLACT-550++ with ResNet-101) and larger scales (e.g. YOLACT-700). Besides, different from YOLACT++ [4], which is $\sim 25\%$ slower than YOLACT (see Table 6 in Bolya et al. [4]), our maYOLACT-550 is faster than YOLACT-550 (Table 4), and still keep 7 mask AP gain also on COCO *test-dev* reaching 35.2 mask AP (Table 5).
Comparison with real-time methods. Without multi-scale training as in Solov2 [35] or specially designed backbone as in CenterMask [36]; our maYOLACT-700 reaches 37.7 mask AP at 25fps and outperforms existing real-time counterparts. Besides, our best model

[1] For all models, we follow and report the results on the mmdetection framework [5] on a single Tesla V100 GPU. Mmdetection's YOLACT is slower than the official implementation by Bolya et al. [3], who reported 45fps.

[2] Note that more efficient models can be obtained by using better anchor design methods [12, 23, 41, 45]

[3] We use 25fps as the cut-off for "real-time" following the common video signal standards (e.g. PAL [37] and SECAM [38]) and existing methods [10, 27, 31, 33].

Table 5: Comparison with SOTA on COCO *test-dev*. Our maYOLACT-700 establishes a new SOTA for real-time instance segmentation. * implies our implementation for YOLACT with ATSS w.IoU. When a paper does not report a performance measure, N/A is assigned and we reproduce the performance using its repository for completeness (shown by †).

	Methods	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Reference
fps < 25	YOLACT-700 [3]	ResNet-101	31.2	50.6	32.8	12.1	33.3	47.1	ICCV 19
	PolarMask [39]	ResNet-101	32.1	53.7	33.1	14.7	33.8	45.3	CVPR 20
	PolarMask++ [40]	ResNet-101	33.8	57.5	34.6	16.6	35.8	46.2	TPAMI 21
	RetinaMask [11]	ResNet-101	34.7	55.4	36.9	14.3	36.7	50.5	Preprint
	Mask R-CNN [6]	ResNet-50	36.8	59.2	39.3	17.1	38.7	52.1	ICCV 17
	TensorMask [6]	ResNet-101	37.1	59.3	39.4	17.4	39.1	51.6	ICCV 19
fps ≥ 25	YOLACT-550 [3]	ResNet-50	28.2	46.6	29.2	9.2	29.3	44.8	ICCV 19
	YOLACT-550*	ResNet-50	29.7	49.9	30.7	11.9	32.4	42.7	Baseline
	Solov2-448 [35]	ResNet-50	34.0	54.0	36.1	N/A	N/A	N/A	NeurIPS 20
	Solov2-448† [2]	ResNet-50	34.0	54.0	36.0	10.3	36.3	53.8	NeurIPS 20
	YOLACT-550++ [4]	ResNet-50	34.1	53.3	36.2	11.7	36.1	53.6	TPAMI 20
	YOLACT-550++ [4]	ResNet-101	34.6	53.8	36.9	11.9	36.8	55.1	TPAMI 20
	CenterMask-Lite† [1]	VoVNetV2-39	35.7	56.7	37.9	**18.4**	37.8	47.3	CVPR 20
	CenterMask-Lite [36]	VoVNetV2-39	36.3	N/A	N/A	15.6	38.1	53.1	CVPR 20
	Solov2-512† [2]	ResNet-50	36.9	57.5	39.4	12.8	39.7	**57.1**	NeurIPS 20
	Solov2-512 [35]	ResNet-50	37.1	57.7	39.7	N/A	N/A	N/A	NeurIPS 20
	maYOLACT-550 (Ours)	ResNet-50	35.2	56.2	37.1	14.7	38.0	51.4	
	maYOLACT-700 (Ours)	ResNet-50	**37.7**	**59.4**	**39.9**	18.1	**40.8**	52.5	

achieves 59.4 wrt. common AP_{50} metric with a gap of 1.7 AP_{50} points compared to its closest real time counterpart (i.e. SOLOv2-512).

Comparison with other methods. Our maYOLACT is also competitive against slower methods (Table 5): It outperforms PolarMask++ [40], RetinaMask [11], Mask R-CNN [14] and TensorMask [6] while being faster. To illustrate, on Tesla V100 GPU, our maYOLACT-700 (i) has $\sim 2\times$ more throughput with 25fps and nearly 4 mask AP gain (37.7 AP - Table 5) compared to PolarMask++ on ResNet-101 with 14 fps test time; and (ii) is $\sim 8\times$ faster than TensorMask on ResNet-101 (i.e. $\sim$ 3fps) with similar performance.

5 Conclusion

We presented maIoU to assign a proximity value for an anchor compared to both a ground truth box and its mask. Using maIoU to assign anchors as positive or negative for training instance segmentation methods, we utilised the shape of objects as provided by the ground-truth segmentation masks. We showed that ATSS with our maIoU also improves throughput of the model. Exploiting this efficiency, we improved the performance further and reached SOTA results in real-time.

Acknowledgements: This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) (under grants 117E054 and 120E494). We also gratefully acknowledge the computational resources kindly provided by TÜBİTAK ULAKBIM High Performance and Grid Computing Center (TRUBA) and Roketsan Missiles Inc. Dr. Kalkan is supported by the BAGEP Award of the Science Academy, Turkey.

References

[1] Official repository of centermask. https://github.com/youngwanLEE/CenterMask. (Last accessed: 01 September 2021).

[2] Official repository of solo variants. https://github.com/WXinlong/SOLO. (Last accessed: 01 September 2021).

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv*, 1906.07155, 2019.

[6] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[7] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary IoU: Improving object-centric image segmentation evaluation. In *CVPR*, 2021.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, 1984.

[10] Wei Fang, Lin Wang, and Peiming Ren. Tinier-yolo: A real-time object detection method for constrained environments. *IEEE Access*, 8:1935–1944, 2020. doi: 10.1109/ACCESS.2019.2961959.

[11] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C. Berg. RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv*, 1901.03353, 2019.

[12] Mingyu Gao, Yujie Du, Yuxiang Yang, and Jing Zhang. Adaptive anchor box mechanism to improve the accuracy in the object detection system. *Multimedia Tools and Applications*, 78:27383–27402, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[15] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *CVPR*, 2020.

[17] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In *The European Conference on Computer Vision (ECCV)*, 2020.

[18] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S. Davis. Learning from noisy anchors for one-stage object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *The European Conference on Computer Vision (ECCV)*, 2014.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(2):318–327, 2020.

[22] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[23] Wenshuo Ma, Tingzhong Tian, Hang Xu, Yimin Huang, and Zhenguo Li. Aabo: Adaptive anchor box optimization for object detection via bayesian sub-sampling. In *The European Conference on Computer Vision (ECCV)*, 2020.

[24] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Localization recall precision (LRP): A new performance metric for object detection. In *The European Conference on Computer Vision (ECCV)*, 2018.

[25] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. One metric to measure them all: Localisation recall precision (lrp) for evaluating visual detection tasks. *arXiv*, 2011.10772, 2020.

[26] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Rank & sort loss for object detection and instance segmentation. In *The International Conference on Computer Vision (ICCV)*, 2021.

[27] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6):1137–1149, 2017.

[29] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[31] Jianqiang Wan, Yang Liu, Donglai Wei, Xiang Bai, and Yongchao Xu. Super-bpd: Super boundary-to-pixel direction for fast image segmentation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[33] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven C. H. Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1448–1457, 2019. doi: 10.1109/CVPR.2019.00154.

[34] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *The European Conference on Computer Vision (ECCV)*, 2020.

[35] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[36] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: Single shot instance segmentation with point representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] Martin H. Weik. *Phase Alternation by Line*, pages 1255–1255. Springer US, Boston, MA, 2001. ISBN 978-1-4020-0613-5. doi: 10.1007/1-4020-0613-6_13881. URL https://doi.org/10.1007/1-4020-0613-6_13881.

[38] Martin H. Weik. *système electronique couleur avec memoire*, pages 1718–1718. Springer US, Boston, MA, 2001. ISBN 978-1-4020-0613-5. doi: 10.1007/1-4020-0613-6_18872. URL https://doi.org/10.1007/1-4020-0613-6_18872.

[39] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Enze Xie, Wenhai Wang, Mingyu Ding, Ruimao Zhang, and Ping Luo. Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[41] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[42] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Ye Rongguang, and Ren Dongwei. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI Conference on Artificial Intelligence*, 2020.

[44] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, pages 1–13, 2021.

[45] Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Anchor box optimization for object detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[46] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.