AttentionMask: Attentive, Efficient Object Proposal Generation Focusing on Small Objects

Supplementary Material

Christian Wilms and Simone Frintrop

University of Hamburg
{wilms,frintrop}@informatik.uni-hamburg.de

The supplementary material presents details of AttentionMask as well as further results and evaluation. Sec. 1 gives additional details about training AttentionMask, especially about the selection of ground truth, the loss function used and different strategies for training the system end-to-end. In Sec. 2 the results of the evaluation on the MS COCO dataset using bounding box annotations are given, while in Sec. 3 further quantitative results from the MS COCO dataset are shown.

1 Training Details

In this section, we give further details on training AttentionMask. Primarily, we describe the selection of ground truth for the objectness module, the attentional head, and the segmentation module as well as the loss function used in the system and the strategies for training the system end-to-end. We also detail the hyperparameters as well as the solver used in training.

1.1 Selection of Ground Truth

The selection of the ground truth for the scale-specific objectness attention maps has already been described in the main paper. Therefore, we focus here on the selection of ground truth for the objectness module, the attentional head, and the segmentation module.

Similar to [19], we select for the objectness module up to 64 sampled windows across all scales with an equal distribution of positive and negative samples. A sample window generated from scale $\mathcal{S}_n$ is regarded as positive, if an object is roughly contained in the window, centered in the window and fits to the scale $\mathcal{S}_n$. As described in the main paper for the ground truth of the scale-specific objectness attention maps, an object fits to the scale $\mathcal{S}_n$, if both side lengths of the object are within 40% to 80% of the sampled window side length of $\mathcal{S}_n$ in the original image. An object is roughly contained in a window, if the center of the object is within the image. The criterion of an object being centered in a window is fulfilled, if the distance between the center of the object and the center of the window is no larger than 10% of the window's side length in the original image.

Following [19], sampling negative examples is not applied randomly among all other windows, but focused on hard negative examples. Thus, either the criterion on object size or the object being centered is removed.

For the selection of ground truth for the attentional head and the segmentation module, the subset of up to 32 positive samples selected for the objectness module is used. Thus, the same criteria as above apply. For the attentional head the bounding box of the object mask serves as ground truth, to get a rough location of the object. For the segmentation module the pixel-precise segmentation mask is used.

1.2 Loss Function

As described in Sec. 5.2 of the main paper, training AttentionMask consists of multiple different losses. For training the objectness module ($\mathcal{L}_{objn}$), the attentional head ($\mathcal{L}_{ah}$) as well as the segmentation module ($\mathcal{L}_{seg}$) we use binary cross entropy loss, similar to [19]. $\mathcal{L}_{ah}$ and $\mathcal{L}_{seg}$ are spatially normalized across the result window, as otherwise the gradients of $\mathcal{L}_{ah}$ and $\mathcal{L}_{seg}$ would overrun the gradient of $\mathcal{L}_{objn}$ [19]. Thus, given the binary cross entropy loss function as

$$L(y, y') = y \cdot -\log(\sigma(y')) + (1 - y) \cdot -\log(1 - \sigma(y')), \tag{1}$$

with sigmoid function σ, prediction y' and ground truth y, the three loss functions for one sample are

$$\mathcal{L}_{objn}(o, o') = L(o, o'), \tag{2}$$

$$\mathcal{L}_{ah}(a_{ah}, a'_{ah}) = \frac{1}{W_{ah} H_{ah}} \sum_{x,y}^{W_{ah}, H_{ah}} L(a_{ah_{x,y}}, a'_{ah_{x,y}}), \text{ and} \tag{3}$$

$$\mathcal{L}_{seg}(s, s') = \frac{1}{W_s H_s} \sum_{x,y}^{W_s, H_s} L(s_{x,y}, s'_{x,y}). \tag{4}$$

Here o, a_{ah} and s denote the ground truth for objectness, the attentional head result, and the segmentation mask, while o', a'_{ah} and s' denote the outputs of the different modules accordingly. W_{ah} and H_{ah} as well as W_s and H_s denote the width and the height of the output of the attentional head and the segmentation module respectively.

Training the SOAMs of AttentionMask is different from the other three losses, as the ground truth is significantly imbalanced. For instance, at scale $\mathcal{S}_8$ for one pixel with the label *object* there exist on average 351 pixels with the label *non-object*. As the first row in Tab. 1 indicates, simply applying the binary cross entropy loss function from Eq. 1 leads to suboptimal performance, despite being spatially normalized across the scale-specific objectness attention map similar to $\mathcal{L}_{ah}$ and $\mathcal{L}_{seg}$. Thus, we evaluate two other strategies.

First, we add weights to the binary cross entropy loss function to balance the inequality between classes. This leads to a weighted binary cross entropy loss

Table 1: Comparison of loss functions for $\mathcal{L}_{att}$.

	AR@100
simple cross entropy loss (Eq. 1)	0.171
simple cross entropy loss with weight (Eq. 5)	0.253
Eq. 1 with negative sample mining 1 : 3 [23]	**0.258**

function

$$L_{r,w_r}(y, y') = y \cdot -\log(\sigma(y')) \cdot r \cdot w_r + (1 - y) \cdot -\log(1 - \sigma(y')) \tag{5}$$

that assigns a higher loss in cases where the ground truth y equals 1 while the prediction y' is different from 1. r in Eq. 5 denotes the ratio of pixels with label *non-object* and label *object* in the ground truth of a scale-specific objectness attention map and thus of negative and positive samples. w_r is a weight factor, which we set to 0.5 for best results.

Second, following [23] and as described in the main paper, we use the negative sample mining strategy and randomly sample 3 *non-object* pixels for each *object* pixel resulting in a set of positive and negative locations S. As loss function in this case, we use the standard binary cross entropy loss function from Eq. 1 with the spatial normalization. The results of evaluating the two strategies are presented in Tab. 1 and show the superiority of the negative sample mining strategy. Thus, the loss function $\mathcal{L}_{att}(a, a')$ for a SOAM is

$$\mathcal{L}_{att}(a, a') = \frac{1}{|S|} \sum_{(x,y) \in S} L(a_{x,y}, a'_{x,y}), \tag{6}$$

with a denoting the ground truth scale-specific objectness attention map and a' the prediction.

Overall, we use a weighted sum of the different loss functions as the overall loss function from one image with N samples

$$\mathcal{L}(a, o, a_{ah}, s, a', o', a'_{ah}, s') = w_{att} \sum_{m=0}^{M} \mathcal{L}_{att_m}(a_m, a'_m) + \tag{7}$$

$$\frac{1}{N} \sum_{n=0}^{N} \Big(w_{objn} \mathcal{L}_{objn}(o_n, o'_n) + \mathbf{1}(o_n) \big(w_{ah} \mathcal{L}_{ah}(a_{ah,n}, a'_{ah,n}) + w_{seg} \mathcal{L}_{seg}(s_n, s'_n) \big) \Big),$$

with M denoting the number of scales used and thus the number of scale-specific objectness attention maps generated. $\mathbf{1}$ denotes the indicator function, implying that only positive samples for the losses $\mathcal{L}_{ah}$ and $\mathcal{L}_{seg}$ add to the overall loss. In our experiments, we set $w_{objn} = 0.5, w_{ah} = 1.25, w_{seg} = 1.25$ and, $w_{att} = 0.25$, as it gave the best results.

Table 2: Approaches for sampling windows during training.

	AR@100
sample windows using SOAM output	0.252
sample windows using ground truth	**0.258**

1.3 Training Strategies

For training the system end-to-end, we evaluate two different strategies regarding the connection between the SOAMs and the selective window sampling module. In the main paper and in contrast to [23], we do not use the output of the SOAMs to sample windows in training. Instead, we use the scale-specific objectness attention ground truth as input for the selective window sampling module. This gives the advantage of immediately training the back of the system ($\mathcal{L}_{objn}$, $\mathcal{L}_{ah}$, and $\mathcal{L}_{seg}$) with useful examples. Tab. 2 indicates that using this strategy produces superior results compared to sampling based on the calculated attention in training. Note that in both cases there is no flow of information backwards from the selective sliding window module to the SOAMs or the ground truth. The extra supervision is not needed, as there are no trained parameters between those modules and the SOAMs have their own ground truth.

1.4 Hyperparameters and Solver

To optimize the combined loss $\mathcal{L}$, we use stochastic gradient descent with an initial learning rate of 0.0001, however multiplying the learning rate for the layers learned from scratch with the factor 10. The momentum equals 0.9, the weight decay 0.00005 and the batch size is 1 since the sampled windows technically form a batch within an image. We train AttentionMask for 17 epochs.

2 Bounding Box Evaluation

In the main paper, we already presented the evaluation results of AttentionMask compared to state-of-the-art class-agnostic object proposal systems based on pixel-precise masks on the MS COCO dataset. Here, we show the results using the smallest bounding boxes of the pixel-precise segmentation masks for evaluation, except for BING and EdgeBoxes that directly regress bounding boxes. Tab. 3 presents the results of BING [6], EdgeBoxes [44], MCG [31], DeepMaskZoom [30], SharpMask [30], and FastMask [19] as well as the proposed $\text{AttentionMask}^{8}_{128}$, $\text{AttentionMask}^{8}_{192}$, and $\text{AttentionMask}^{16}_{192}$ on the first 5000 validation images of MS COCO using bounding boxes. The results for BING [6] are taken from [17].

Table 3: Results on MS COCO with bounding box proposals. S, M, L denote small, medium, large objects.

Method	AR@10	AR@100	AR@1k	AR^S@100	AR^M@100	AR^L@100	Time
BING [6]	0.037	0.084	0.163	-	-	-	**0.20s**
EdgeBoxes [44]	0.074	0.178	0.338	0.017	0.138	0.505	0.31s
MCG [31]	0.101	0.246	0.398	-	-	-	45s
DeepMaskZoom [30]	0.191	0.378	0.511	0.141	0.493	0.617	1.35s
SharpMask [30]	0.198	0.367	0.490	0.063	0.514	0.674	1.03s
SharpMaskZoom [30]	0.202	0.397	0.533	0.147	0.519	0.648	2.02s
FastMask [19]	**0.227**	0.430	0.568	0.175	**0.549**	0.692	0.33s
$\text{AttentionMask}^{8}_{128}$	0.214	0.426	0.570	**0.210**	0.508	0.673	0.22s
$\text{AttentionMask}^{8}_{192}$	0.221	**0.435**	**0.576**	0.206	0.512	0.710	0.22s
$\text{AttentionMask}^{16}_{192}$	0.219	0.425	0.554	0.148	0.542	**0.726**	0.21s

3 Additional Qualitative Results

Fig. 1, Fig. 2, and Fig. 3 show additional qualitative results from the first 5000 validation images of MS COCO.

Fig. 1: Qualitative results of SharpMaskZoom [30], FastMask [19] and $\text{AttentionMask}^{8}_{128}$ on the MS COCO dataset. The filled colored contours denote found objects, while not filled red contours denote missed objects.

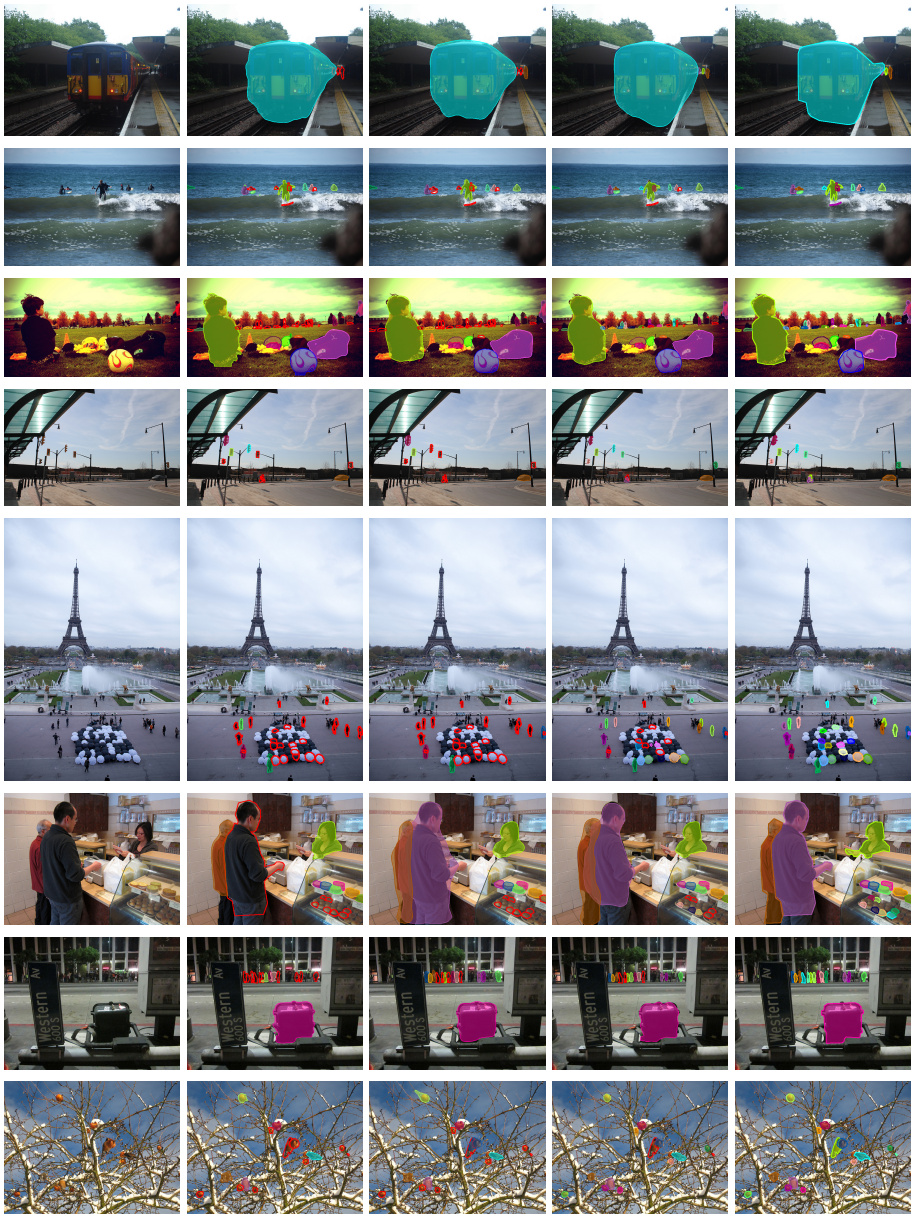

Fig. 2: Qualitative results of SharpMaskZoom [30], FastMask [19] and AttentionMask$^{8}_{128}$ on the MS COCO dataset. The filled colored contours denote found objects, while not filled red contours denote missed objects.

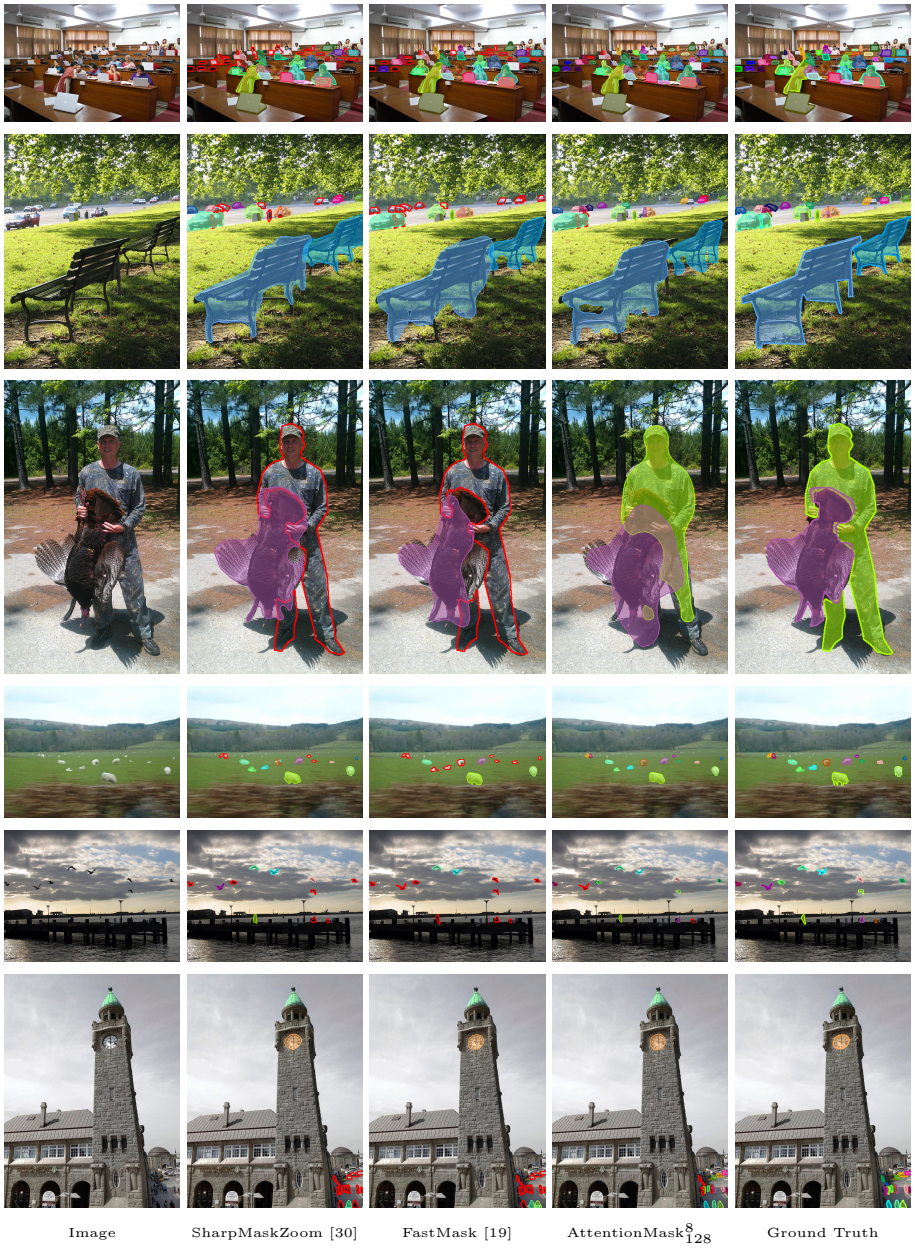

Fig. 3: Qualitative results of SharpMaskZoom [30], FastMask [19] and AttentionMask$^{8}_{128}$ on the MS COCO dataset. The filled colored contours denote found objects, while not filled red contours denote missed objects.

References

For bibliography see the main paper.