Faster Training of Mask R-CNN by Focusing on Instance Boundaries☆

Roland S. Zimmermann[a,b,1], Julien N. Siems[a,c,2]

[a]BMW Car IT GmbH, Lise-Meitner-Straße 14, 89081 Ulm, Germany
[b]Georg-August University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany
[c]Albert Ludwig University of Freiburg, Fahnenbergplatz, 79085 Freiburg im Breisgau, Germany

Abstract

We present an auxiliary task to Mask R-CNN, an instance segmentation network, which leads to faster training of the mask head. Our addition to Mask R-CNN is a new prediction head, the Edge Agreement Head, which is inspired by the way human annotators perform instance segmentation. Human annotators copy the contour of an object instance and only indirectly the occupied instance area. Hence, the edges of instance masks are particularly useful as they characterize the instance well. The Edge Agreement Head therefore encourages predicted masks to have similar image gradients to the groundtruth mask using edge detection filters. We provide a detailed survey of loss combinations and show improvements on the MS COCO Mask metrics compared to using no additional loss. Our approach marginally increases the model size and adds no additional trainable model variables. While the computational costs are increased slightly, the increment is negligible considering the high computational cost of the Mask R-CNN architecture. As the additional network head is only relevant during training, inference speed remains unchanged compared to Mask R-CNN. In a default Mask R-CNN setup, we achieve a training speed up of 29% and an overall improvement of 8.1% on the MS COCO metrics compared to the baseline.

Keywords: Mask R-CNN, Instance Segmentation, Computer Vision, Auxiliary Task, Edge Detection Filter, Sobel Filter, Laplace Filter, Convolutional Neural Network

1. Introduction

Significant improvements in computer vision techniques have been made possible by the rapid progress of training Deep Convolutional Neural Networks in recent years. Application areas include image classification [1, 2, 3, 4] and object detection [5, 6, 7]. One of the most demanding computer vision tasks is instance segmentation, as it involves localizing and segmenting object instances. Recently, there have been multiple methods [8, 9, 10, 11] proposed to perform this task.

Another beneficial factor to the success of these Deep Learning architectures is the availability of large labelled datasets such as MS COCO [12] and the Cityscapes dataset [13]. Labelling an image dataset for instance segmentation tasks is particularly time consuming, because it requires segmenting all objects in a scene. It is therefore highly desirable to speed up training of an instance segmentation model to be more data efficient. In this work we propose a conceptually straightforward addition to the Mask R-CNN [11] architecture which reduces training time of the mask branch.

The Mask R-CNN architecture is based on Faster R-CNN [14], which introduced an efficient Region Proposal Network (RPN) design to output bounding box proposals. The proposals are computed using a sliding window approach to make them translation invariant. A feature extractor such as ResNet [4], Inception [15] or VGGNet [3] is used as input to the region proposal network. The regions and features are used in the bounding box regression head, that refines the bounding box localization and the softmax classification head, which determines the instance class. This second stage is the architecture as described in Fast R-CNN [5].

Mask R-CNN is a simple but effective addition to the Faster R-CNN architecture that adds a head for instance mask prediction. Using a small Fully Convolutional Neural Network (FCN) [16], it can predict pixel level instance masks. Besides the mask branch, it uses a Feature Pyramid Network (FPN) backbone as proposed by Lin et al. [17]. This addition allows the network to make use of both high-resolution feature maps in the lower layers for accurate localization, as well as semantically more meaningful higher level features, which are of lower resolution. Another contribution is ROI Align which maps region of interests from the input image to the feature map without rounding up to integer dimensions of the feature map using bilinear interpolation. This modification improves the COCO Mask metrics and enables the use of instance masks which require precise localization.

For the mask head a new loss term L_{Mask} has been introduced, which calculates the pixel-wise cross entropy between the pre-

☆Both authors contributed equally to this work and most both be cited. They are both corresponding authors.

[1]R. S. Zimmermann is with the University of Göttingen, Göttingen, Germany. This work was done when he was an intern at BMW Car IT GmbH, Ulm, Germany. Email: roland.zimmermann@stud.uni-goettingen.de

[2]J. N. Siems is with the Albert Ludwig University of Freiburg, Freiburg, Germany. This work was done when he was an intern at BMW Car IT GmbH, Ulm, Germany. Email: julien.siems@jupiter.uni-freiburg.de

dicted and target masks. The Mask R-CNN loss function

$$L_{MRCNN} = L_{Class} + L_{Box} + L_{Mask} \quad (1)$$

is a multi-task loss based on the Faster R-CNN loss.

We propose to attach an Edge Agreement Head to the mask branch of Mask R-CNN which acts as an auxiliary task to Mask R-CNN. This head uses traditional edge detection filters such as Sobel and Laplacian kernels [18, 19] on both the predicted mask and the groundtruth mask to encourage their edges to agree. Instances in natural images are bounded by the edges that annotators use to mark the instance. Therefore, we show that encouraging the edges in the predicted and groundtruth mask to agree leads to faster training of our mask head. We argue that this is a result of the instance boundary being a robust feature to mask prediction, which can be easily propagated from the image to the mask branch.

2. Related Work

Multi Task Learning. The Edge Agreement Head acts as an auxiliary task [20] to the multi-task model Mask R-CNN, which is performing both object detection and instance segmentation. Auxiliary tasks have shown to encourage models to learn robust representations of their input in a variety of applications, such as facial landmark detection [21], natural language processing [22] or steering prediction in autonomous driving [23]. Even seemingly unrelated tasks, e.g. weather prediction to semantic scene segmentation, can improve the models overall performance [24].

Monocular Depth Estimation. Godard et al. [25] use image gradients to encourage consistency between input images and predicted disparity maps. However, the left-right disparity consistency loss does not ensure image gradients of the predicted disparity of the left and right camera to exhibit similar edge detection filter responses.

Scene segmentation. Chen et al. [26] show a two-part model predicting both semantic segmentations and edges. The semantic segmentation model is based on the DeepLab model [27] and the edge detection filter is created using intermediate convolutional filters of the DeepLab model. The task specific edge detection on the input image is used to refine the coarse segmentation using domain transform. Our approach determines edges in fixed size, low dimensional instance mask images, for which traditional edge detection filters have been proven to be effective. Similarly, Marmanis et al. [28] predict both semantic scene segmentation and semantic boundaries. The network responsible for predicting semantic boundaries is trained using an Euclidean loss before each pooling layer to enforce each layer to predict edges at different scales. Our approach uses predefined edge detection filters with well-known properties, which are kept constant during training, leaving us with a significantly lower additional memory footprint and computational costs.

Instance segmentation. Hayder et al. [29] propose a model that predicts the truncated distance transform [30] of the mask, making it more resilient towards non-instance enclosing bounding box proposals. The proposed architecture for the boundary-aware instance segmentation network has many similarities to Mask R-CNN as they are both based on the Faster R-CNN architecture by Ren et al. [14]. However, they achieved lower results on the instance segmentation benchmark on the Cityscapes dataset compared to Mask R-CNN, which we are basing our work on.

Edge detection. The detection of edges has been a research topic for many decades and numerous methods have been proposed [18, 31]. This field has seen large improvements due to deep learning techniques [32, 33, 34]. Our work uses the Sobel image gradient filters proposed in [18], because it keeps the computational overhead to a minimum. Furthermore, our edge detection filters are used on 28×28 sized masks with only one channel depicting a single instance and not high resolution color images. This significantly reduces the complexity of the problem and justifies our choice of simple edge detection filters.

3. Edge Agreement Loss

When training a Mask R-CNN for instance segmentation one often observes incomplete or poor masks, especially during early training steps. Furthermore, the masks often do not follow the real object boundaries. Possible mistakes such as missing parts or oversegmentation are illustrated in Figure 1.

To reduce this problem, we are inspired by how a human would perform instance segmentation: instead of immediately assigning parts of the image to specific objects one often identifies at first the boundaries of the object and fills the enclosed area. To help the network perform the segmentation in an analogous way, i.e. show the importance of edges and boundaries of objects, we have constructed an auxiliary loss called Edge Agreement Loss L_{Edge}. It is defined as the L^p loss between the edges in the predicted mask and the groundtruth mask. The total loss L_{Total} consists of the original Mask R-CNN loss L_{MRNN} (eq. 1) and the new Edge Agreement Loss L_{Edge} which are summed. To compute this new loss, the first step is to identify the edges in the predicted and the groundtruth mask.

3.1. Edge Detection

In detail, we examined edge detection filters which can be described as a convolution with a 3×3 kernel, such as the well-known *Sobel* and *Laplacian* filters.

The Sobel filters [18] are two-dimensional filters to detect edges. As the filters describe a first-order gradient operation they are rotation-dependent. There are two filters

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

which describe the horizontal and the vertical gradient respectively. An edge in the image corresponds to a high absolute

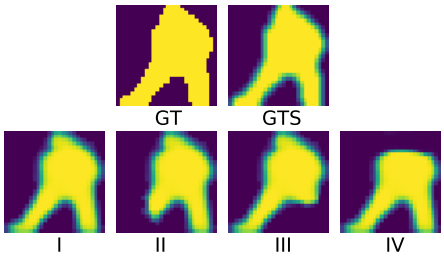


Figure 1: Overview of different example masks to illustrate the effect of the Edge Agreement Loss. *GT* corresponds to the groundtruth, *GTB* to the Gaussian smoothed groundtruth, and *I* to *IV* represent four example mask predictions which demonstrate early-stage predictions of the Mask R-CNN during training.

response along the filter's direction. In the following the concatenation of both filters into a $3 \times 3 \times 2$ dimensional tensor is referenced as the Sobel filter S.

The Laplacian filter is a discretization of the two-dimensional Laplacian operator (i.e. the second derivative). The filter

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3}$$

is the direct result of a finite-difference approximation of the derivative [19]. The operator is known to be rotation invariant which means that it can detect edges in both x and y direction. As it is a second-order operator, an edge in the image corresponds to a zero-crossing, rather than a strong filter response. By including the main- and anti-diagonal elements the filter can be made responsive to 45° angles

$$L = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{4}$$

This is the Laplacian kernel (L) used in all further experiments.

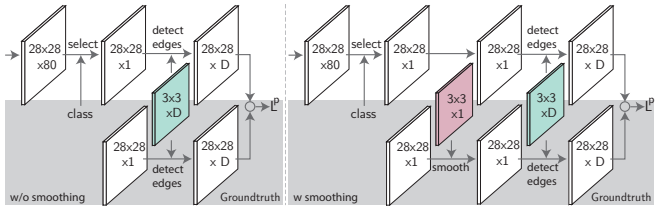


Figure 2: Edge Agreement Head: We extend the existing mask branch architecture. Left/right show the edge agreement head with/-out Gaussian smoothing. Of the $28 \times 28 \times 80$ dimensional output of the mask branch, the mask corresponding to the correct class is selected. The head computes a convolution of the selected mask and the groundtruth mask with the $3 \times 3 \times D$ dimensional edge detection filter (turquoise). Between these a L^p loss is calculated, which results in the term L_{Edge}. If Gaussian smoothing is used (right side), the groundtruth mask will be preprocessed by a 3×3 dimensional Gaussian kernel (red) before the edges are determined (Best viewed in color).

3.2. Loss Construction

To calculate the final loss L_{Edge} we propose an additional network head, called the Edge Agreement Head. It uses the predicted and the matched groundtruth masks as input, which are

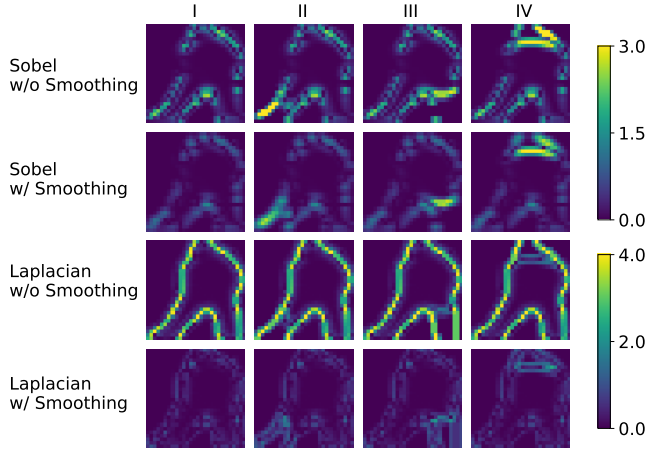


Figure 3: L^2 errors for the four example predictions and the different methods. Each column *I* to *IV* corresponds to one of the examples in Figure 1. The first two rows show the L^2 error based on the Sobel filter magnitude, while for the last two rows the Laplace filter is used. The second and fourth row are calculated on the Gaussian smoothed groundtruth. While the losses calculated on the default groundtruth do not respond strongly to missing areas the losses on the smoothed groundtruth are particularly pronounced in these areas.

then convolved with a selection of edge detection filters. Afterwards, the difference between the predicted and groundtruth edge maps are determined. The entire procedure is illustrated in Figure 2 in the left half. For this task, we choose the set of L^p loss functions. Mathematically they can be expressed as the p-th power of the generalized power mean M_p of the absolute difference between the target $\hat{\mathbf{y}}$ and the prediction $\mathbf{y}$

$$L^p(\mathbf{y}, \hat{\mathbf{y}}) = M_p(|\mathbf{y} - \hat{\mathbf{y}}|)^p. \tag{5}$$

For $p = 2$ this equals the mean square error, commonly used in deep learning.

The application of the previously introduced filters on the exemplary masks of Figure 1 is displayed in Figure 3. The figure illustrates how the L^2 loss between the edge maps, calculated as mentioned above, does not only contain important information but even possibly distracting information: the images in the rows for Sobel without Smoothing and Laplace without Smoothing depict a high error rate which is not limited to areas in which the person has not been segmented but also occurs around the entire boundary. This is mainly caused by the fact, that the groundtruth mask is binary and the mask branch's output is continuous and shows often smooth transitions.

To overcome this problem, we add an additional step in our branch which performs Gaussian smoothing on the groundtruth mask, yielding a smooth version of the binary groundtruth. For this we use an approximate 3×3 Gaussian kernel. The resulting architecture of the head is shown in Figure 2 on the right. The calculated L^p distance using this head proposal is shown in the rows for Sobel with Smoothing and Laplace with Smoothing of the figure mentioned beforehand. Notably, the loss calculated on the smoothed groundtruth focuses particularly on areas with missing parts, while the Edge Agreement Loss on the default groundtruth has a higher value on the overall mask boundary.

Both proposals for the edge agreement head can be calculated with only minimal additional computational and memory

requirements. This means, that the method can be integrated in existing systems for training Mask R-CNN without requiring new or additional hardware.

4. Implementation Details

A mask size of 28×28 pixels and an image resolution of 1024×800 pixels are used. All training images are resized to this size preserving their aspect ratio. As the training images may have different aspect ratios, the remaining space of the image is zero padded. This method differs from the one used in the original Mask R-CNN implementation [11], where resizing is done such that the smallest side is 800 pixels and the largest is trimmed at 1000 pixels.

A similar training strategy to other Mask R-CNN work [11] is followed. We choose to train the network for 160k steps on the MS COCO 2017 train dataset with a batch size of 2 on a single GPU machine, while for Mask R-CNN an effective batch size of 16 was used. The training consists of three stages each lasting for 40k, 80k, 40k steps respectively: in the first stage only the Mask R-CNN branches and not the ResNet backbone are trained. Next, the prediction heads and parts of the backbone (starting at layer 4) are optimized. Finally, in the third stage, the entire model (backbone and heads) is trained together. For the first two training stages we use a learning rate of 0.001 and for the last one a decreased learning rate of $0.001/10$. The optimization is done by SGD with momentum set to 0.9 and weight decay set to 0.0001.

5. Experiments

We perform our experiments using the implementation of Mask R-CNN by matterport [35], based on the Keras framework [36] with a TensorFlow backend [37]. Each training is carried out on a single GPU using either a NVIDIA Titan X or a NVIDIA GeForce GTX 1080 Ti.

We examine three aspects of the proposed Edge Agreement Head. At first, we inspect the influence of the edge detection filters on the training speed (section 5.1). Afterwards (section 5.2), we test whether smoothing the groundtruth is a helpful extension, as suggested in section 3.2. In section 5.3, the different metrics of the L^p family are used to examine the influence of the loss function's steepness on the training speed. Section 5.4 shows the impact weighting the Edge Agreement Loss has on the overall loss. In section 5.5 we show the results on the metrics after longer training. Finally, in section 5.6 we elaborate on modifications to the Edge Agreement Head which didn't have a positive effect on the training.

For all experiments we follow the same scheme: every network configuration examined is trained and evaluated three times. The resulting training curves and metrics displayed are the averaged values. Furthermore, to be able to compare all runs and to reduce the time required for all experiments we do not train the networks until they have converged, but only for a fixed and limited number of training steps. The only data augmentation used in all three steps are random horizontal flips.

We present the COCO metrics for our experiments and compare them with the results obtained using a Mask R-CNN model without modifications (baseline) for every experiment conducted. A significant disadvantage of using the COCO Mask metrics is that they do not compare the groundtruth and predicted mask pixel per pixel since they only consider the area and the instance enclosing bounding box. As a result, the COCO Mask metrics are unsuitable to compare improvement in the details of instance masks which are located on the inside of the mask and do not affect the extremities. We use the COCO metrics because of their dominance in other publications. The mask loss however, can be regarded as a better metric to compare the quality of the instance mask, because it is computed using a cross entropy between matched groundtruth and predicted mask.

Furthermore, we present the training speed-up factor (SU), which is measured by comparing the required number of update steps for the configuration at hand to reach the same Mask R-CNN Loss L_{MRCNN} as the baseline model after 160k steps.

5.1. Influence of Filters

In the first experiment the choice of edge detection filters on the training speed and the mask quality is analyzed. The Edge Agreement Loss is computed using the L^2 loss. The mask loss L_{Mask} and the original Mask R-CNN loss L_{MRCNN} are displayed in Figure 4a. The results on the COCO Mask metrics and the speed-up factor are shown in Table 1. Using the Sobel edge detection filter gave a 7% improvement on the AP score. Notably the Sobel filter resulted in a 12% improvement on the AP_{75} and a 10% improvement on the AP_S compared to the respective baseline scores. On average, the COCO metrics have been improved by 8.1%. The combination of Sobel and Laplacian (S & L) was the second-best configuration. Using the Laplacian kernel showed only marginal improvements over the baseline.

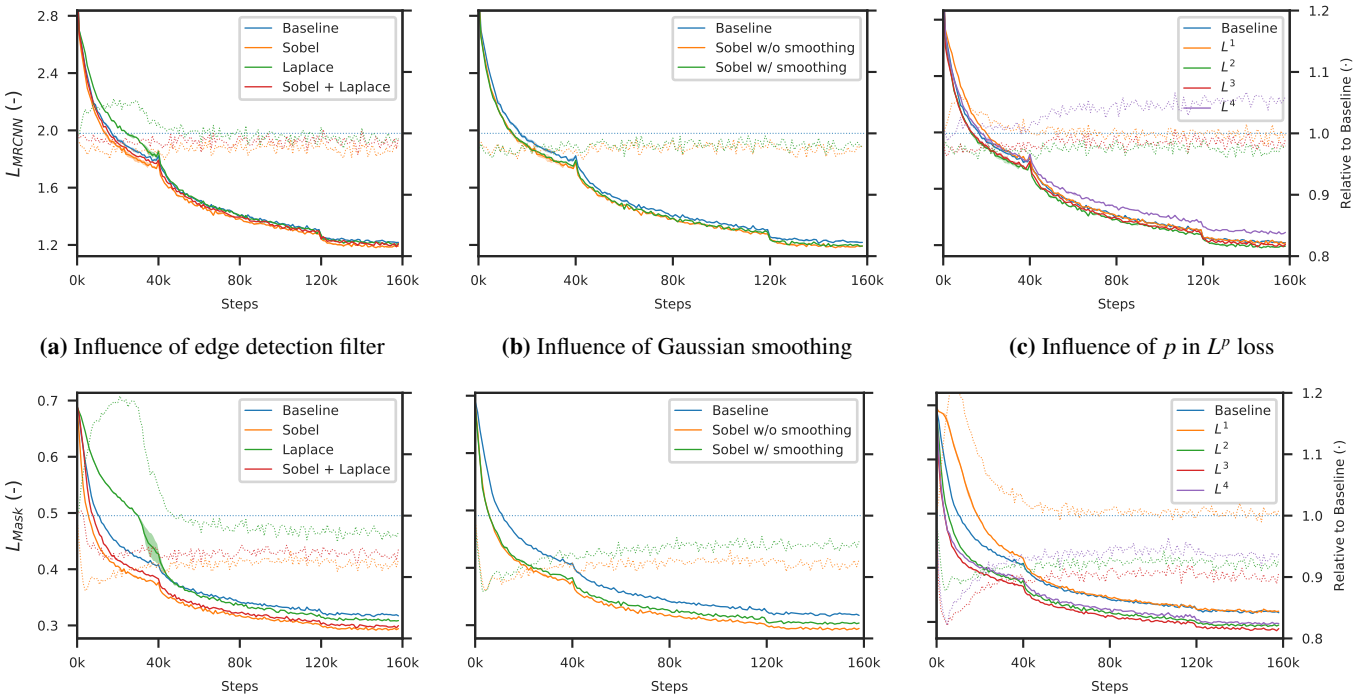
A possible explanation for the superiority of the Sobel filter is its structure: as it consists of two filters, not only the strength of an edge along the x and y axis but also the edge's orientation can be used during the gradient descent to minimize the total loss. This additional information accelerates the training

A qualitative comparison between computed masks is shown in Figure 5. We observe that the models trained with Edge Agreement Loss tend to be less likely to propose bounding boxes which do not contain any object, therefore reducing false positives. This indicates that the features needed to minimize the Edge Agreement Loss are also useful to the Region Proposal Network.

5.2. Influence of Smoothing

The effectiveness of smoothing the groundtruth mask before applying an edge detection filter is evaluated using the Sobel, Laplace and the combination Sobel & Laplace filter. The resulting COCO metrics are listed in Table 2, while the losses for the Sobel Edge Detection Filter are displayed in Figure 4b.

Contrary to expectation, the Sobel filter was more effective when used without smoothing the groundtruth, as the Mask



(a) Influence of edge detection filter **(b)** Influence of Gaussian smoothing **(c)** Influence of p in L^p loss

Figure 4: Comparison of different Edge Agreement Head configurations. The left y-axis corresponds to the absolute loss values (solid lines) and the right y-axis corresponds to the relative improvement compared to the baseline (dotted lines). The first row shows the original Mask R-CNN Loss L_{MRCNN} while the second row shows the Mask Loss L_{Mask}. The first column illustrates the influence of different edge detectors used in the Edge Agreement Head, while the second demonstrates the influence of Gaussian smoothing when using a Sobel edge detection filter. The last column compares the performance of different L^p loss functions for the Edge Agreement Loss (Best viewed in color).

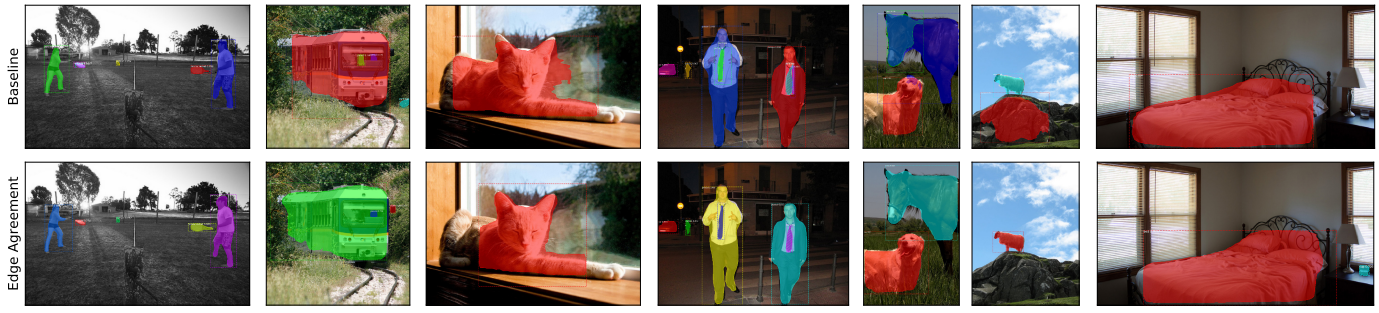


Figure 5: Comparison between masks predicted by Baseline Mask R-CNN and Mask R-CNN with Edge Agreement Head using a Sobel edge detection filter and no groundtruth smoothing after 160k steps on images taken from the MS COCO 2017 dataset [12] (Best viewed in color).

R-CNN training loss fell faster and was lower in this case, as shown in Figure 4b. Particularly the Mask Loss during training is lower. However, overall the Edge Agreement Loss (both Sobel with and without smoothed groundtruth) is beneficial and their training loss stayed below the baseline for the majority of the time.

The Laplacian filter was evaluated with and without smoothing the groundtruth as well. Smoothing slowed training even further than without smoothing the groundtruth. We observe that if instead of the groundtruth mask, the predicted mask is smoothed the training is significantly more stable. This is likely a result of the Laplacian kernel's sensitivity to noise, which is reduced when first smoothing the predicted mask before computing the Edge Agreement Loss.

The results obtained contradict the theoretical considerations of the possible benefit to smoothing the groundtruth. The smoothing of the groundtruth was designed to ignore minor mistakes at the boundary of an almost perfectly predicted mask, but only focusing on major mistakes. Apparently, the network does not only profit from highlighting the most crucial mistakes in the predicted masks, but rather from all mistakes done.

5.3. Influence of the Choice p in L^p loss

Next, the influence of the exponent p in L^p chosen for the Edge Agreement Loss is analyzed. For all previous experiments an L^2 loss was used; now different values of $p \in \{1, 2, 3, 4\}$ are applied. As an increasing value of p increases the steepness of the loss, falsely detected edges are penalized stronger. For this evaluation we used the Sobel edge detection filter without smoothing the Groundtruth.

The two losses L_{MRCNN} and L_{Mask} are displayed in Figure 4c. While a higher value for p causes the mask loss to decrease, it also increases the overall loss. The metrics obtained in these experiments are listed in Table 3. Overall, choosing the L^2 loss appears to be the best choice, as it yields the best results on the COCO metrics.

Table 1: Influence of the choice of edge detection filters on the instance segmentation mask COCO AP metrics after 160k steps. Higher is better. The last column shows the relative speedup compared to the baseline.

	AP	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$	**SU**
Sobel	**20.2 ± 0.17**	**37.5 ± 0.37**	**20.0 ± 0.07**	**8.8 ± 0.27**	**21.9 ± 0.18**	**28.9 ± 0.30**	**29%**
Laplace	19.4 ± 0.12	36.5 ± 0.19	18.9 ± 0.18	8.0 ± 0.22	21.0 ± 0.10	27.8 ± 0.11	21%
S & L	20.0 ± 0.25	37.0 ± 0.36	19.6 ± 0.24	8.3 ± 0.03	21.7 ± 0.30	28.4 ± 0.47	23%
Baseline	18.8 ± 0.14	36.5 ± 0.24	17.8 ± 0.13	8.0 ± 0.21	20.4 ± 0.29	26.6 ± 0.24	–

Table 2: Influence of Gaussian smoothing on the instance segmentation mask COCO AP metrics after 160k steps. Higher is better. The last column shows the relative speedup compared to the baseline.

	AP	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$	**SU**
S w/ smoothing	19.7 ± 0.01	37.3 ± 0.15	19.1 ± 0.16	8.7 ± 0.07	21.3 ± 0.03	27.9 ± 0.14	25%
S w/o smoothing	**20.2 ± 0.17**	**37.5 ± 0.37**	**20.0 ± 0.07**	**8.8 ± 0.27**	21.9 ± 0.18	**28.9 ± 0.30**	**29%**
L w/ smoothing	18.5 ± 0.37	36.2 ± 0.49	17.3 ± 0.43	7.9 ± 0.18	20.1 ± 0.44	26.2 ± 0.50	4.6%
L w/o smoothing	19.4 ± 0.12	36.5 ± 0.19	18.9 ± 0.18	8.0 ± 0.22	21.0 ± 0.10	27.8 ± 0.11	21.4%
S & L w/ smoothing	20.1 ± 0.14	37.3 ± 0.32	**20.0 ± 0.27**	8.4 ± 0.11	**22.1 ± 0.29**	28.5 ± 0.19	24%
S & L w/o smoothing	20.0 ± 0.25	37.0 ± 0.36	19.6 ± 0.24	8.3 ± 0.03	21.7 ± 0.30	28.4 ± 0.47	23%
Baseline	18.8 ± 0.14	36.5 ± 0.24	17.8 ± 0.13	8.0 ± 0.21	20.4 ± 0.29	26.6 ± 0.24	–

5.4. Influence of Weighting Factor on Edge Agreement Loss

By choosing a higher value of p for the L^p Edge Agreement Loss, the loss becomes steeper and yields higher values for wrongly predicted masks. This increment also implies a higher relative importance of the Edge Agreement Loss compared to the other loss functions in the sum of the total loss which usually stay in the range $[0, 1]$.

In these trainings we used the Sobel edge detection filter without smoothing the groundtruth.

To examine the influence of the relative importance of the new Edge Agreement Loss to the other losses, we include a factor α which scales the Edge Agreement Loss. We test its influence on the usage of the L^2 and L^4 losses to investigate the impact of the Mask Edge Loss on the total loss. For this comparison all trainings are performed only once and up to 120k steps instead of 160k steps, as already after 120k steps a clear trend has been recognizable.

Figure 6a shows the Mask R-CNN loss, while the Edge Agreement Loss is scaled by $\alpha \in \{0.5, 1, 8, 16\}$. The Mask R-CNN loss increases with higher weight factor, despite faster decreasing Mask Loss, the other loss terms remain higher. In fact, the L^2 loss with weight factor 1.0 already appears to be a good trade-off between enforcing better predicted masks and optimizing the other objectives of the network.

The L^4 loss yields high values compared to the L^2 loss. Therefore, we scale it by $\alpha = 1/16$, which is approximately the ratio of Edge Agreement Loss between using L^2 and L^4 in the first few steps. The training progression for the Mask R-CNN loss is shown in Figure 6b. Reducing the Edge Agreement Loss improves training significantly, making the Loss stay below the Baseline for most of the steps.

5.5. Longer training

To measure the effect of the Edge Agreement Head after longer training, we increased the number of steps previously used (320k and 640k steps rather than 160k). In this case the last step of the training schedule, in which all layers are trained, was extended from 40k steps to 200k and 520k steps (a total of 320k and 640k training steps respectively). The results are shown in Table 4.

All metrics improved as expected when trained for additional steps. Interestingly, in most metrics the Mask R-CNN model trained with Edge Agreement Head trained for 160k steps was not only superior to the baseline trained for 160k steps but also to the one trained 320k steps. No significant influence of the Edge Agreement Loss on losses other than the Mask Loss is observed. We notice that the difference in the Mask Loss between a baseline Mask R-CNN and one trained with Edge Agreement Loss remains constant with later training steps. This was contrary to our own intuition that the Edge Agreement Loss would primarily be helpful early in training. It was expected that the Mask Loss of the two models would approach each other, but this was not found to be the case. We conclude that the Edge Agreement Head is not only useful early on in training, but can guide the training even in later training steps and change the point of convergence.

It should be noted that our results on the MS COCO dataset [12] are significantly lower than the results reported by He et al. [11]. Firstly, we are not using the official implementation of Mask R-CNN made available in the Detectron [38], but an independent implementation which reported lower results of their pretrained models [35]. Secondly, we use a batch size of 2, while Mask R-CNN used an effective batch size of 16. We argue that this does not hurt the generality of our method.

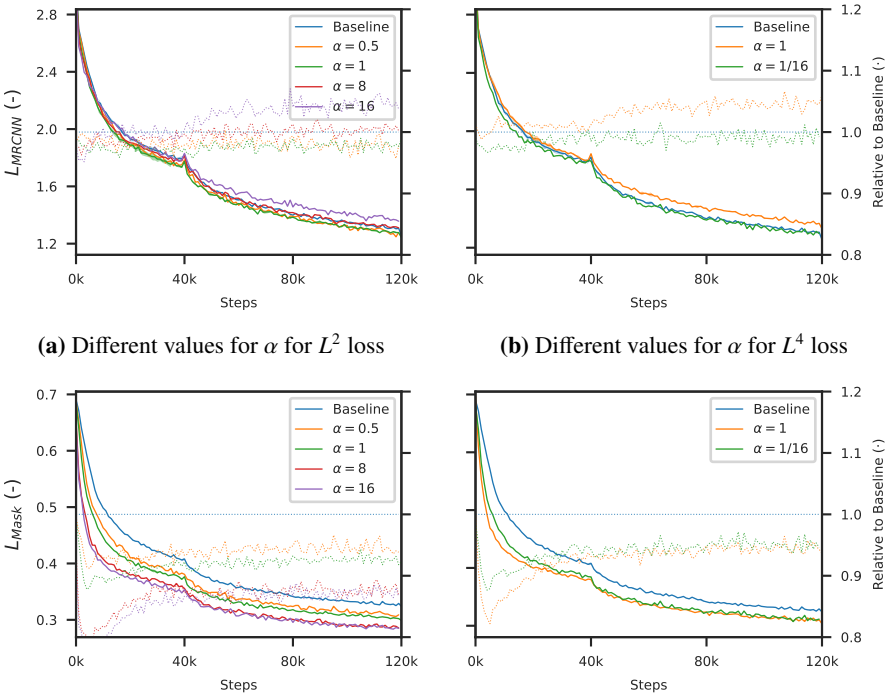
5.6. Other experiments

The configuration of the Edge Agreement Loss we describe above was found to have the optimal impact on the training. We tried a variety of modifications which showed either no effect or had a negative impact on the training.

As discussed in section 5.4, the magnitude of the Edge Agreement Loss appears to have a high influence on the L_{MRCNN} loss. In an attempt to balance the loss terms we tried homoscedastic task uncertainty as proposed by Kendall et al. [39]. Our approach was to weigh all the loss terms including

Table 3: Influence of the chosen L^p loss on the instance segmentation mask AP COCO metrics after 160k steps. Higher is better. The last columns shows the relative speedup compared to the baseline.

	AP	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$	**SU**
L^1	19.5 ± 0.28	36.6 ± 0.41	18.9 ± 0.41	8.2 ± 0.30	21.0 ± 0.32	27.7 ± 0.5	11%
L^2	**20.2 ± 0.17**	**37.5 ± 0.37**	20.0 ± 0.07	**8.8 ± 0.27**	**21.9 ± 0.18**	**28.9 ± 0.30**	**29%**
L^3	**20.2 ± 0.20**	37.0 ± 0.41	**20.1 ± 0.22**	8.6 ± 0.14	21.8 ± 0.24	28.5 ± 0.53	22%
L^4	17.8 ± 0.13	33.5 ± 0.12	17.4 ± 0.13	7.6 ± 0.12	19.3 ± 0.23	24.7 ± 0.24	0%
Baseline	18.8 ± 0.14	36.5 ± 0.24	17.8 ± 0.13	8.0 ± 0.21	20.4 ± 0.29	26.6 ± 0.24	–



(a) Different values for α for L^2 loss

(b) Different values for α for L^4 loss

Figure 6: Influence of the weighting factor α on the behavior of the Edge Agreement Loss. The first row displays again the original total loss of Mask R-CNN L_{MRCNN} while the second row displays only the mask loss L_{Mask}. The first column shows the loss trajectory for different alpha values using the L^2 loss whereas the second column shows the influence on the L^4 loss. Best viewed in color

the Edge Agreement Loss. However, the results were consistently worse than the baseline and therefore not included in this paper.

Furthermore, we tried to weigh the cross entropy mask loss L_{Mask} with the Edge Agreement Loss. Two different formulations for this weighted cross entropy loss were tried out, which can be expressed as

$$L_{Edge} = L_{Mask-PW} \cdot L_{Edge-PW}$$

or

$$L_{Edge} = L_{Mask-PW} \cdot \exp\left(L_{Edge-PW}/4\right),$$

using $L_{Mask-pw}$ and $L_{Edge-pw}$ to denote the pixel-wise Mask Loss and pixel-wise Edge Agreement Loss respectively. For both formulations the results were identical with the more concise formulation of the Edge Agreement Loss that we used in the rest of the paper.

In addition, when using the Sobel filter, we did not solely consider the horizontal and vertical image gradient but also the gradient's magnitude for calculating the L^p Edge Agreement Loss. No improvement compared to not including the magnitude was found. Therefore, it was not used in the rest of the paper.

6. Conclusion

In this paper we have analyzed the behavior of Mask R-CNN networks during early training steps. By inspecting the predicted masks of the mask branch, we recognized that these often have blurry boundaries which do not follow sharp and fine contours of the original masks. To reduce this symptom, we successfully introduced a parameter free network head, the Edge Agreement Head. This head uses classical edge detection filters applied on the instance masks to calculate a L^p loss between the predicted and groundtruth mask contours.

By including the new Edge Agreement Loss in the training, we achieved a performance increment of 8.1% averaged over all the MS COCO metrics after a fixed number of 160k training steps.

The ablation studies performed showed that the Sobel filter yields a better performance than the Laplace filter. Beyond expectations, the proposed smoothing of the groundtruth mask did not improve but hinder the performance. Out of all losses examined the often-used L^2 loss performs the best.

When trained longer, the difference in Mask Loss between a baseline Mask R-CNN and one with Edge Agreement Head persists, demonstrating the effectiveness of the additional loss not only early during training but also during later steps.

Table 4: Comparison of the instance segmentation mask AP COCO metrics of our best performing model with the baseline after an extended training duration. The best performing model uses the Edge Agreement Head with Sobel edge detection filter, no groundtruth smoothing and L^2 Edge Agreement Loss.

	AP	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$
Ours 160k steps	20.2 ± 0.17	37.5 ± 0.37	20.0 ± 0.07	8.8 ± 0.27	21.9 ± 0.18	28.9 ± 0.30
Ours 320k steps	21.3	38.7	21.1	8.9	23.2	30.0
Ours 640k steps	**22.7**	**41.0**	**23.1**	**10.2**	**24.6**	**32.0**
Baseline 160k steps	18.8 ± 0.14	36.5 ± 0.24	17.8 ± 0.13	8.0 ± 0.21	20.4 ± 0.29	26.6 ± 0.24
Baseline 320k steps	20.0	38.5	19.1	8.6	21.6	28.2
Baseline 640k steps	21.5	40.5	20.8	9.0	22.9	30.7

7. Future work

The idea to enforce edge agreement in predicted semantic segmentation could be applied to scene segmentation for example on the DeepLab architecture [27]. Monocular depth estimation could also potentially be enhanced by encouraging the predicted depth map to have comparable gradients to the groundtruth depth map image gradients.

Furthermore, balancing the different individual losses contained in the total loss by introducing new scaling variables might be a necessary step to further increase the training speed. Instead of introducing new static hyperparameters for the multi-task loss one could modify the gradients like Chen et al. [40].

As the Edge Agreement Loss accelerates the training of the Mask Head, it enables Mask R-CNN to be used more easily with sparse labels for object instance masks. This allows new training strategies for new datasets, e.g. one could mix a few hand segmented frames with datasets containing only object bounding boxes, such as PASCAL VOC [41] or the more recent Open Images dataset [42].

Acknowledgments

This research has been conducted during an internship at BMW Car IT GmbH in Ulm. The researchers would like to thank the company for the financial support and for the resources offered. Furthermore, the authors sincerely thank Krassimir Valev and Stefan Müller from BMW Car IT GmbH for their help in organizing this publication.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105. doi:10.1145/3065386.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi:10.1109/CVPR.2015.7298594.

[3] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR), 2015.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.90.

[5] R. Girshick, Fast r-cnn, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169.

[6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.91.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision (ECCV), Springer, 2016, pp. 21–37. doi:10.1007/978-3-319-46448-0_2.

[8] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.472.

[9] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.305.

[10] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[11] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask r-cnn, in: The IEEE International Conference on Computer Vision (ICCV), 2017. doi:10.1109/ICCV.2017.322.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.350.

[14] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031.

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: AAAI Conference on Artificial Intelligence, 2017.

[16] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi:10.1109/TPAMI.2016.2572683.

[17] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.106.

[18] I. Sobel, G. Feldman, A 3x3 isotropic gradient operator for image processing., in Hart, P. E. & Duda R. O. Pattern Classification and Scene Analysis.

[19] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall Professional Technical Reference, 2002.

[20] S. Ruder, An overview of multi-task learning in deep neural networks, CoRR abs/1706.05098.

[21] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: European Conference on Computer Vision (ECCV), Springer, 2014, pp. 94–108. doi:10.1007/978-3-319-10599-4_7.

[22] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167. doi:10.1145/1390156.1390177.

[23] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.
[24] L. Liebel, M. Körner, Auxiliary tasks in multi-task learning, arXiv preprint arXiv:1805.06334.
[25] C. Godard, O. Mac Aodha, G. J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.699.
[26] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, A. L. Yuille, Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi:10.1109/CVPR.2016.492.
[27] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2018) 834–848. doi:10.1109/TPAMI.2017.2699184.
[28] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, U. Stilla, Classification with an edge: Improving semantic image segmentation with boundary detection, ISPRS Journal of Photogrammetry and Remote Sensing 135 (2018) 158–172. doi:https://doi.org/10.1016/j.isprsjprs.2017.11.009.
[29] Z. Hayder, X. He, M. Salzmann, Boundary-aware instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi:10.1109/CVPR.2017.70.
[30] G. Borgefors, Distance transformations in digital images, Computer vision, graphics, and image processing 34 (3) (1986) 344–371. doi:10.1016/S0734-189X(86)80047-0.
[31] S. Konishi, A. L. Yuille, J. M. Coughlan, S. C. Zhu, Statistical edge detection: Learning and evaluating edge cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 57–74. doi:10.1109/TPAMI.2003.1159946.
[32] G. Bertasius, J. Shi, L. Torresani, Deepedge: A multi-scale bifurcated deep network for top-down contour detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi:10.1109/CVPR.2015.7299067.
[33] W. Shen, X. Wang, Y. Wang, X. Bai, Z. Zhang, Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi:10.1109/CVPR.2015.7299024.
[34] S. Xie, Z. Tu, Holistically-nested edge detection, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1395–1403. doi:10.1007/s11263-017-1004-z.
[35] W. Abdulla, Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow (2017).
[36] F. Chollet, et al., Keras, https://keras.io (2015).
[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/
[38] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, K. He, Detectron, https://github.com/facebookresearch/detectron (2018).
[39] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[40] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: ICML, 2018.
[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338. doi:10.1007/s11263-009-0275-4.
URL https://doi.org/10.1007/s11263-009-0275-4
[42] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, K. Murphy, Openimages: A public dataset for large-scale multi-label and multi-class image classification., Dataset available from https://storage.googleapis.com/openimages/web/index.html.

Authors

Roland S. Zimmermann received his B.Sc. degree with distinction in Physics from Georg-August University of Göttingen, Göttingen, Germany in 2017 where he is currently pursuing his M.Sc. His research interests lie in the areas of non-linear physics, computer vision and recurrent neural networks.

Julien N. Siems received his B.Sc. degree in Computer Science from TU Dresden, Dresden, Germany in 2017. He is currently pursuing his M.Sc. at the Albert Ludwig University of Freiburg, Freiburg, Germany. His research interests lie in the areas of object detection, segmentation and probabilistic graphical models.