BoxShrink: From Bounding Boxes to Segmentation Masks

Michael Gröger$^{(\boxtimes)}$, Vadim Borisov , and Gjergji Kasneci

University of Tübingen, Tübingen, Germany
`michael.groeger@posteo.net`

Abstract. One of the core challenges facing the medical image computing community is fast and efficient data sample labeling. Obtaining fine-grained labels for segmentation is particularly demanding since it is expensive, time-consuming, and requires sophisticated tools. On the contrary, applying bounding boxes is fast and takes significantly less time than fine-grained labeling, but does not produce detailed results. In response, we propose a novel framework for weakly-supervised tasks with the rapid and robust transformation of bounding boxes into segmentation masks without training any machine learning model, coined BoxShrink. The proposed framework comes in two variants – *rapid*-BoxShrink for fast label transformations, and *robust*-BoxShrink for more precise label transformations. An average of four percent improvement in IoU is found across several models when being trained using BoxShrink in a weakly-supervised setting, compared to using only bounding box annotations as inputs on a colonoscopy image data set. We open-sourced the code for the proposed framework and published it online.

Keywords: Weakly-supervised learning · Segmentation · Colonoscopy · Deep neural networks

1 Introduction

Convolutional neural networks (CNNs) have achieved remarkable results across image classification tasks of increasing complexity, from pure image classification to full panoptic segmentation, and have become, as a consequence, the standard method for these tasks in computer vision [19]. However, there are also certain drawbacks associated with these methods. One of them is that in order to achieve satisfactory results, a data set of an appropriate size and high-quality labels are needed [21]. The costs and time associated with labeling increase with the complexity of the task, with image classification being the cheapest and image segmentation being the most expensive one. All of these challenges especially

---

M. Gröger and V. Borisov—equal contribution.

apply to medical artificial intelligence (MAI) applications since they depend on the input and feedback by expensive domain experts [22].

In this work, we present a novel approach for fast segmentation label prepossessing, *which is decoupled from any particular artificial neural network architecture*. The proposed algorithmic framework can serve as a first approach for practitioners to transform a data set with only bounding box annotations into a prelabeled (i.e., semantically segmented) version of the data set. Our framework consists of independent components such as superpixels [23], fully-connected conditional random fields [14] and embeddings. This makes it easy to add our framework to an existing machine learning pipeline.

To evaluate the proposed framework, we select an endoscopic colonoscopy data set [4]. Multiple experiments show that our framework helps to considerably reduce the gap between the segmentation performance and efficiency of a neural network that is trained only on bounding boxes and one trained on fully segmented segmentation masks.

The main contributions of this work are:

– We propose the BoxShrink framework consisting of two methods. One for a time-efficient and one for a more robust transformation of bounding-boxes into segmentation masks. In both methods there is no need to train a model.
– We publish our bounding-box labels for the CVC-Clinic data set for future research in the area of weakly-supervised learning.
– We open-source our code and publish it online.[1]

2   Related Work

In this Section, we further define weakly-supervised learning and separate it from other approaches such as semi-supervised learning. Also, we localize our work among those which use similar components.

To reduce the need for resources such as time and money, various learning methodologies were introduced such as semi-supervised and weakly-supervised learning [30]. Semi-supervised learning leverages labeled data, e.g. for segmentation tasks correctly and fully segmented images and the availability of a larger amount of unlabeled data [16]. Weakly-supervised learning on the other hand, exploits noisy labels as a weak supervisory signal to generate segmentation masks. These labels can be provided in different forms such as points [3], or image-level labels [27], being the more simpler ones, or more complex ones such as scribbles [15,24], or bounding boxes [6,11]. A similar work [29] to ours also utilizes superpixel embeddings and CRFs, but their method requires an additional construction of a graph of superpixels and a custom deep neural network architecture. Our method, on the other hand, is easier to integrate into existing pipelines. Also, in contrast to many other weakly-supervised approaches [10,28], we do not apply CRFs as a postprocessing step on the output of the model but as a preprocessing step on the input, hence, we leave the downstream model untouched. Furthermore, the proposed framework does not require special hardware such as GPU or TPU for the label preprocessing step.
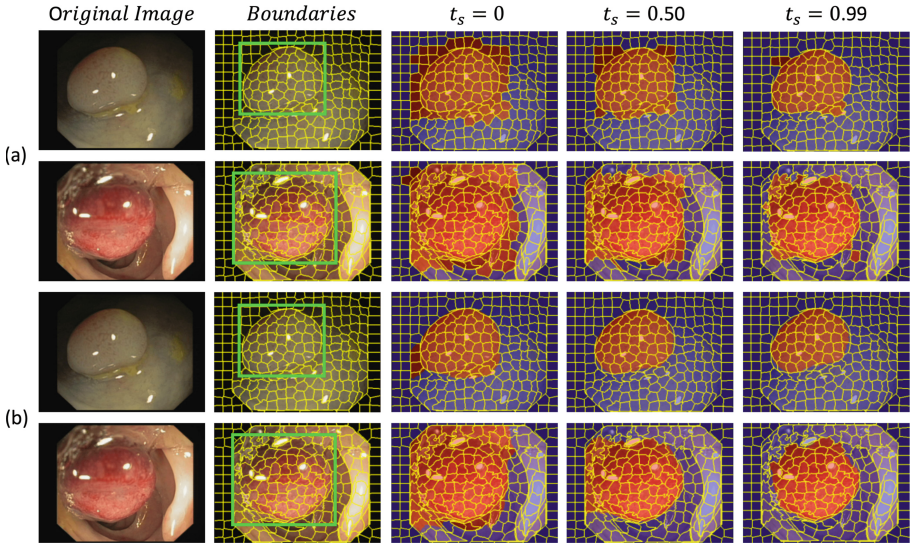
---

[1] https://github.com/michaelgroeger/boxshrink.

Fig. 1. The impact of varying the threshold t_s, i.e., a hyperparameter of the BoxShrink framework for tuning the final segmentation quality, where (a) shows two data samples from the data set after the superpixel assignment step (Sect. 3.2), and (b) demonstrates pseudo-masks after the FCRF postprocessing. As seen from this experiment, having a higher threshold might generate better masks but increases the risk of losing correct foreground pixels.

3   Boxshrink Framework

This section presents our proposed BoxShrink framework. First, we define its main components: superpixel segmentation, fully-connected conditional random fields, and the embedding step. We then explain two different settings of the framework, both having the same goal: to reduce the number of background pixels labeled as foreground contained in the bounding box mask.

3.1   Main Components

Superpixels aim to group pixels into bigger patches based on their color similarity or other characteristics [23]. In our implementation, we utilize the SLIC algorithm proposed by [1] which is a k-means-based algorithm grouping pixels based on their proximity in a 5D space. A crucial hyperparameter of SLIC is the number of segments to be generated which is a upper bound for the algorithm on how many superpixels should be returned for the given image. The relationship between the output of SLIC and the maximum number of segments can be seen in the supplementary material.

Fully-connected-CRFs are an advanced version of conditional random fields (CRFs) which represent pixels as a graph structure. CRFs take into account

a unary potential of each pixel and the dependency structure between that pixel and its neighboring ones using pairwise potentials [25]. Fully-connected-CRFs (FCRFs) address some of the limitations of classic CRFs, such as the lack of capturing long-range dependencies by connecting all pixel pairs. Equation 1 shows the main building block of FCRFs which is the Gibbs-Energy function [13].

$$E(x) = \sum_{i=1}^{N} \psi_u(x_i) + \sum_{i<j}^{N} \psi_p(x_i, x_j), \qquad (1)$$

where the first term $\psi_u(x_i)$ measures the unary potential, that is, the cost if the assigned label disagrees with that of the initial classifier, the second term $\psi_p(x_i, x_j)$ measures the pairwise potential, which is the cost if two similar pixels disagree on their label x. The input is over all pixels N. We use FCRFs to smooth the output pseudo-mask.

Superpixel Embeddings are a key component of the *robust*-BoxShrink variant. The embedding function M produces a numerical representation of every superpixel $k_i \in \boldsymbol{K}$ by returning an embedding vector. Formally, this operation can be depicted $M : \mathbb{R}^m \to \mathbb{R}^n$. Practically, this can be done by feeding each superpixel k_i separately into a CNN model, such as a Resnet-50 [9] pretrained on ImageNet [7]. By doing so, we obtain a 2048-dimensional vector representation for every superpixel. It allows us to get an aggregated representation of the foreground and background, by computing the mean embedding of all foreground and background superpixels in the training data set. These mean vectors are then used to assign superpixels either to the foreground or background class based on their cosine similarity.

3.2   *rapid*-BoxShrink

We first split each image into superpixels using the SLIC algorithm for the *rapid*-BoxShrink strategy. We overlap the superpixels with the provided bounding box mask and build a new mask based on those superpixels, which overlap the bounding box mask to a certain threshold. This approach is based on the assumption that the object of interest is always fully contained in the bounding box. The results depend on the number of segments generated which can be seen in the supplementary materials and the chosen threshold shown in Fig. 1. To this end, as shown in the supplementary material in Algorithm 1, to make the final pseudo-mask more smooth, we run a FCRF as described in Sect. 3.1 on the thresholded superpixel mask.

3.3   *robust*-BoxShrink

Leveraging the availability of superpixels, we also explore the use of embeddings to shrink the number of background pixels in the pseudo-mask. We segmented each image in the training data set into superpixels and then assigned them either to the foreground or background group by applying the thresholding approach as we have done it in the *rapid*-BoxShrink variant (Sect. 3.2). To
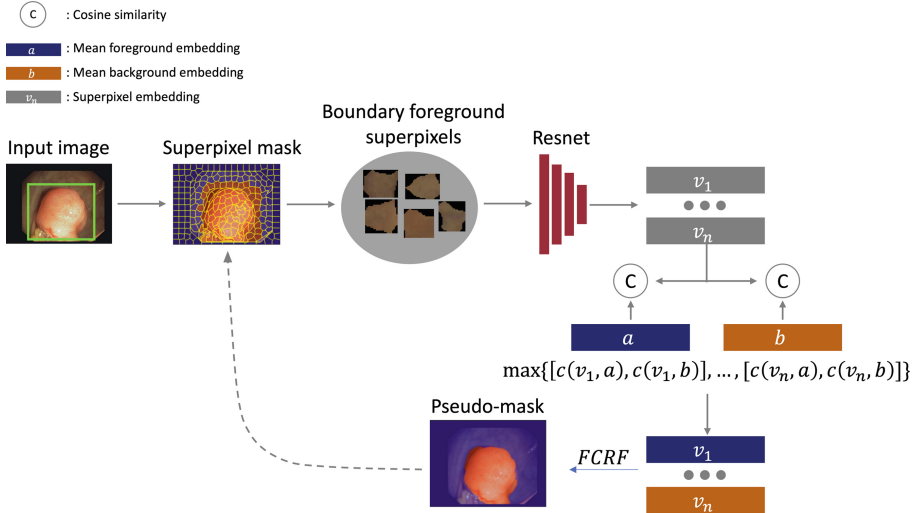
Fig. 2. Overview of the *robust*-BoxShrink method assuming the mean embedding vectors are given. First, we generate a superpixel mask based on the *rapid*-BoxShrink approach but without utilizing the FCRF. Then, we extract each foreground superpixel on the boundary between foreground and background. Feeding each superpixel into a pretrained ResNet model yields one 2048-dimensional embedding vector per superpixel. Next, we calculate the cosine similarity score of each embedding and the mean background and foreground embedding. Based on the highest score we either keep the superpixel as foreground or assign it to the background class. Finally, we apply a FCRF on the resulting superpixel mask. The dashed line indicates that this approach can be run iteratively.

generate the pseudo-masks, we start with the bounding box mask and segment the image using again the thresholding technique. This yields $\mathcal{F}$ superpixels for the foreground and $\mathcal{B}$ superpixels for the background. Then we go along the boundary foreground superpixels $\mathcal{F}_o$ and assign them either to the background or foreground class, depending on their cosine similarity score to the mean background and foreground embedding. The whole process can be seen in Fig. 2. The Algorithm 2, which can be found in the supplementary materials, summarizes the main steps of the *robust*-BoxShrink method.

4 Experiments

This Section presents qualitative and quantitative experiments for both versions of the BoxShrink framework.

Data Set. For all our experiments we utilize the endoscopic colonoscopy frames for polyp detection data set (CVC-Clinic DB) [4], it consists of 612 endoscopy images, each having a size of $288 \times 384 \times 3$. The data set comes along with binary ground truth segmentation masks, which we utilize for the evaluation of
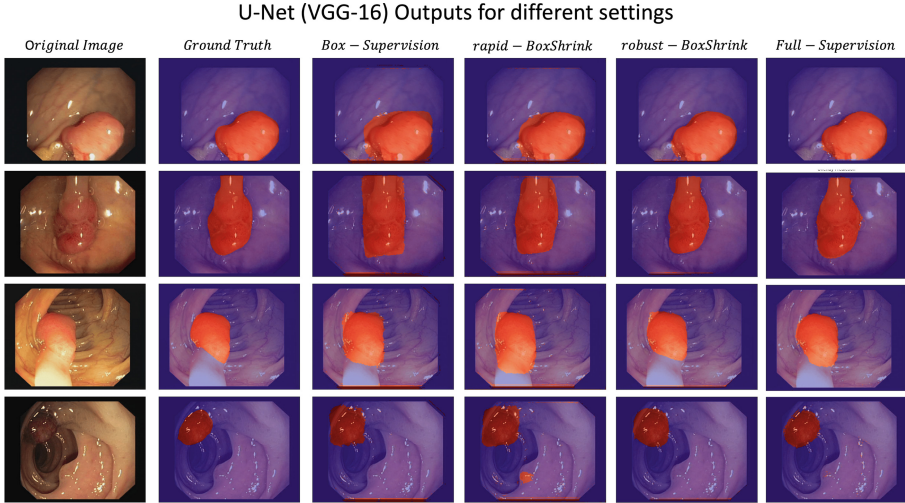
U-Net (VGG-16) Outputs for different settings

| *Original Image* | *Ground Truth* | *Box − Supervision* | *rapid − BoxShrink* | *robust − BoxShrink* | *Full − Supervision* |

Fig. 3. Qualitative model prediction masks on four random samples from the CVC-Clinic test set. The setting on which the model was trained on is indicated on top.

our weakly-supervised framework and to infer the bounding boxes. This data set was featured in multiple studies [2,8].

4.1 Qualitative and Quantitative Experiments

For our experiments, we utilize two popular deep learning architectures for segmentation tasks - U-Net [20] and DeepLabV3+ [5].

Settings. We have four settings, using: (1) Bounding boxes as labels which serves as our lower baseline, (2) labels generated with the *rapid*-BoxShrink label transformation strategy, (3) labels generated with the *robust*-BoxShrink label transformation strategy, and (4) a fully-supervised upper baseline with segmentation masks as labels.

Quality Measure. We use the Intersection over Union (IoU) score as an evaluation measure. The IoU, also called Jaccard similarity J between two sets $\boldsymbol{A}$ and $\boldsymbol{B}$, is a commonly used measure of how well the prediction aligns with the ground truth in image segmentation [18]. As the equation below shows, the IoU is computed by dividing the intersection of two masks by their union.

$$J(\boldsymbol{A}, \boldsymbol{B}) = \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|}. \tag{2}$$

Results. We present the quantitative results in Table 1. In line with other publications, we also share situations where our presented Framework fails. Figure 5, which can be found in the supplementary materials shows some examples.

Figure 3 shows some good prediction masks from the test set made by models trained on the aforementioned four different settings.

4.2   Reproducibility Details

We split the CVC-Clinic DB data set into 80% training data, 10% validation data and 10% test data. For splitting, we use the implementation from sklearn [17] with a random state of 1. To generate the superpixel masks, we set the maximum number of segments s to 200, a threshold t_s of 0.6 for all training images and use the implementation from skimage [26]. To get the embeddings, we use a maximum number s of 250 segments and a threshold t_s of 0.1 to not loose too much of the foreground. To smooth the superpixel masks we use the FCRF implementation provided by the pydensecrf package.[2] Note that we do not train the FCRF (similar to [10]) and set the FCRF hyperparmeters of the x/y-standard deviation for the pairwise Gaussian to 5 and for the pairwise bilateral to 25. We set the rgb-standard deviation to 10. To determine the best performing model, we use the intersection over union (IoU) during training on the validation set. After the training, the best performing model is kept and evaluated once on the test set. Both, the test and validation set consist of ground truth masks. We generate all models using the segmentation-models PyTorch library.[3]

For our experiments we select ResNet-18, ResNet-50, and VGG-16 backbones pretrained on the ImageNet data set paired with U-Net and DeepLabV3+ as a decoder. We use the Sigmoid function as an activation function and the Adam [12] optimizer with a learning rate of 0.0001. As the loss function we utilize the Cross-Entropy Loss. During training, we apply step-wise learning rate scheduling where we decay the learning rate by 0.5 each 5 epochs. We train the ResNet-18 & VGG-16 architecture for 25 epochs and the ResNet-50 architecture for 15 epochs. The training is being done on a 16 GB Nvidia Tesla P-100. We use a batch size of 64 when using the ResNet-18, 32 for the VGG-16 architecture and 16 when using ResNet-50. For both methods, *rapid*-BoxShrink and *robust*-BoxShrink, we return the initial bounding box mask if the total mask occupancy, that is the ratio of the bounding box and the total image is less than 0.1 or the IoU between the pseudo mask and the bounding box mask is less than 0.1.

5   Discussion

In this Section, we further discuss the application and future work of the proposed weakly-supervised framework.

The choice between *rapid*-BoxShrink and *robust*-BoxShrink depends on multiple factors - the time budget and expected label transformation quality. In our experiments, we observe that *rapid*-BoxShrink takes on average only 0.5 seconds to transform the labels for a singe data sample, where *robust*-BoxShrink

---

[2] https://github.com/lucasb-eyer/pydensecrf.
[3] https://github.com/qubvel/segmentation_models.pytorch.

Table 1. Experimental results on the CVC-Clinic data set. All models are evaluated on the *ground truth segmentation mask* in the validation and test set. The label format indicates the initial input label on which the model was either trained or our proposed frameworks were applied to. The results are averages of six runs; we also report the corresponding standard deviation for each setting. This is being done to deliver a more consistent picture because of the random initialization of the decoder part and the stochasticity of the optimizer. The best performing results for our proposed methods are marked in bold. Higher IoU is better.

Segmentation model	Label format	Backbone	Validation (IoU)	Test (IoU)
U-Net	Bounding Boxes	VGG-16	0.749 ± 0.023	0.772 ± 0.030
U-Net (*rapid*-BoxShrink)	Bounding Boxes	VGG-16	0.769 ± 0.026	0.807 ± 0.028
U-Net (*robust*-BoxShrink)	Bounding Boxes	VGG-16	$\mathbf{0.775} \pm 0.013$	$\mathbf{0.824} \pm 0.010$
U-Net	Segment. Masks	VGG-16	0.796 ± 0.025	0.829 ± 0.025
U-Net	Bounding Boxes	ResNet-18	0.691 ± 0.051	0.729 ± 0.060
U-Net (*rapid*-BoxShrink)	Bounding Boxes	ResNet-18	0.730 ± 0.021	0.781 ± 0.024
U-Net (*robust*-BoxShrink)	Bounding Boxes	ResNet-18	$\mathbf{0.755} \pm 0.021$	$\mathbf{0.808} \pm 0.021$
U-Net	Segment. Masks	ResNet-18	0.800 ± 0.032	0.859 ± 0.044
U-Net	Bounding Boxes	ResNet-50	0.785 ± 0.010	0.810 ± 0.010
U-Net (*rapid*-BoxShrink)	Bounding Boxes	ResNet-50	0.807 ± 0.018	0.851 ± 0.019
U-Net (*robust*-BoxShrink)	Bounding Boxes	ResNet-50	$\mathbf{0.813} \pm 0.015$	$\mathbf{0.852} \pm 0.012$
U-Net	Segment. Masks	ResNet-50	0.889 ± 0.012	0.920 ± 0.016
DeepLabV3+	Bounding Boxes	VGG-16	0.746 ± 0.033	0.766 ± 0.034
DeepLabV3+ (*rapid*-BoxShrink)	Bounding Boxes	VGG-16	$\mathbf{0.779} \pm 0.023$	$\mathbf{0.817} \pm 0.0201$
DeepLabV3+ (*robust*-BoxShrink)	Bounding Boxes	VGG-16	0.767 ± 0.0187	0.809 ± 0.024
DeepLabV3+	Segment. Masks	VGG-16	0.832 ± 0.049	0.858 ± 0.051
DeepLabV3+	Bounding Boxes	ResNet-18	0.723 ± 0.025	0.758 ± 0.021
DeepLabV3+ (*rapid*-BoxShrink)	Bounding Boxes	ResNet-18	0.743 ± 0.021	0.787 ± 0.026
DeepLabV3+ (*robust*-BoxShrink)	Bounding Boxes	ResNet-18	$\mathbf{0.759} \pm 0.005$	$\mathbf{0.806} \pm 0.002$
DeepLabV3+	Segment. Masks	ResNet-18	0.808 ± 0.010	0.844 ± 0.012

needs on average 3 seconds to complete the label transformation, the processing time can be further optimized in future versions. However, from our extensive experiments (Sect. 4.1), we can conclude that *robust*-BoxShrink tends to outperform *rapid*-BoxShrink in the weakly-supervised setting. The difference between the two variants is smaller for bigger models with *rapid*-BoxShrink being once better than *robust*-BoxShrink for the VGG-16 architecture. One explanation could be that bigger models are more robust to the label noise than smaller ones. We want to point out however, that the margin between the two is still overlapped by the standard deviations of both methods.

Future Work. We want to further integrate the framework into the training pipeline by, e.g., adjusting the mean foreground and background embeddings as the model gets better. Also, we have evaluated our approach on a medium-sized data set with binary class segmentation. For a more detailed quality evaluation, an analysis of BoxShrink's performance on multi-class problems and bigger

data sets is required. Lastly, starting with BoxShrink pseudo-masks instead of bounding box annotations directly could also improve existing state-of-the-art weakly-supervised learning algorithms.

6   Conclusion

In this work, we presented BoxShrink, a weakly-supervised learning framework for segmentation tasks. We successfully demonstrate the effectiveness of the BoxShrink framework in the weakly-supervised setting on a colonoscopy medical image data set, where we employ bounding-box labeling and output the segmentation masks. Compared to the fully-supervised setting, our weakly-supervised framework shows nearly the same results. Finally, we open-sourced and published the code and bounding boxes for the CVC-Clinic data set .

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Akbari, M., et al.: Polyp segmentation in colonoscopy images using fully convolutional network. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 69–72. IEEE (2018)
3. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: semantic segmentation with point supervision. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 549–565. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_34
4. Bernal, J., et al.: Comparative validation of polyp detection methods in video colonoscopy: results from the MICCAI 2015 endoscopic vision challenge. IEEE Trans. Med. Imaging **36**(6), 1231–1249 (2017)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
6. Dai, J., He, K., Sun, J.: BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1635–1643 (2015)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
8. Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: PraNet: parallel reverse attention network for polyp segmentation. In: Martel, A.L., Abolmaesumi, P., Stoyanov, D., Mateus, D., Zuluaga, M.A., Zhou, S.K., Racoceanu, D., Joskowicz, L. (eds.) MICCAI 2020. LNCS, vol. 12266, pp. 263–273. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59725-2_26
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

10. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7014–7023 (2018)
11. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: weakly supervised instance and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 876–885 (2017)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in neural Information Processing Systems, vol. 24 (2011)
14. Krähenbühl, P., Koltun, V.: Parameter learning and convergent inference for dense random fields. In: International Conference on Machine Learning, pp. 513–521. PMLR (2013)
15. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3159–3167 (2016)
16. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12674–12684 (2020)
17. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
18. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: Bebis, B., et al. (eds.) ISVC 2016. LNCS, vol. 10072, pp. 234–244. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50835-1_22
19. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. **29**(9), 2352–2449 (2017)
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
21. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: a survey. IEEE Trans. Neural Netw. Learn. Syst. (2022)
22. Sourati, J., Gholipour, A., Dy, J.G., Tomas-Fernandez, X., Kurugol, S., Warfield, S.K.: Intelligent labeling based on fisher information for medical image segmentation using deep learning. IEEE Trans. Med. Imag. **38**(11), 2642–2653 (2019)
23. Stutz, D., Hermans, A., Leibe, B.: Superpixels: an evaluation of the state-of-the-art. Comput. Vis. Image Underst. **166**, 1–27 (2018)
24. Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C.: Normalized cut loss for weakly-supervised CNN segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1818–1827 (2018)
25. Triggs, B., Verbeek, J.: Scene segmentation with CRFs learned from partially labeled images. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
26. Van der Walt, S., et al.: scikit-image: image processing in python. PeerJ **2**, e453 (2014)
27. Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H., Yang, R.: Salient object detection in the deep learning era: An in-depth survey. IEEE Trans. Pattern Anal. Mach. Intell. (2021)

28. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: a simple classification to semantic segmentation approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1568–1576 (2017)
29. Xing, F.Z., Cambria, E., Huang, W.B., Xu, Y.: Weakly supervised semantic segmentation with superpixel embedding. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 1269–1273. IEEE (2016)
30. Yang, X., Song, Z., King, I., Xu, Z.: A survey on deep semi-supervised learning. arXiv preprint arXiv:2103.00550 (2021)