Widening the Focus: Biomedical Image Segmentation Challenges and the Underestimated Role of Patch Sampling and Inference Strategies

Frederic Madesta[1], Rüdiger Schmitz[1,2(✉)], Thomas Rösch[2], and René Werner[1]

[1] Department of Computational Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[2] Department for Interdisciplinary Endoscopy, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
{f.madesta,r.schmitz}@uke.de

Abstract. Image analysis challenges have considerably influenced the recent years in natural and biomedical computer vision. With several important architectures and training strategies having emerged from image analysis challenges, they are often interpreted as contests in model design and training, and much effort is put into optimization of these aspects.

This paper is to widen the focus beyond model architecture and training pipeline design by shedding a light on inference efficiency and the underestimated role of patch sampling strategies. A notable influence of the patch overlap on the challenge scores for successful MICCAI challenges of the previous year is found, in contrast to this parameter being systematically reported in rarely any challenge paper. These edge-overlap effects are shown to be etiologically related to varying dataset-specific intra-patch accuracies. Finally, novel strategies for inference-time patch sampling – other than strided cropping and including Monte Carlo - and uncertainty-based strategies – are proposed and examined, where special focus is put on effects that overarch the single-dataset level and, amongst other effects, an improved performance in the low patch number regimen is achieved.

Drawing on these findings, practical guidance is provided to the reader, and potential challenge participant, on how inference strategies can be optimized experimentally. Moreover, implications on the ongoing best practice debate with respect to challenge design and reporting are discussed. In the hope it may stipulate interest in the undervalued topic of optimized sampling strategies, our inference framework and the

F. Madesta and R. Schmitz—Equal contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-59719-1_29) contains supplementary material, which is available to authorized users.

A. L. Martel et al. (Eds.): MICCAI 2020, LNCS 12264, pp. 289–298, 2020.
https://doi.org/10.1007/978-3-030-59719-1_29

source codes for the patch sampling strategies are made publicly available (https://github.com/IPMI-ICNS-UKE/inference-patch-sampling).

Keywords: Inference · Patch sampling · Segmentation · Biomedical image analysis challenges

1 Introduction

1.1 Background

From the 2012 ImageNet challenge won by AlexNet [10] through U-Net [16] at the ISBI 2015 cell tracking challenge to today - image analysis challenges have visibly shaped the fields of computer vision and medical image analysis in the recent years. They have thus acquired a reputation of innovators in terms of model architectures and training strategies and many of them have a continuing influence as benchmarks, lasting much beyond their official end. What is reported and compared in challenges hence also influences the technical literature relying on them as benchmarks.

Two properties (amongst others) are common for many biomedical image analysis challenges: (i) the task or a sub-task of it is segmentation or effectively boils down to it and (ii) a single image, being from the domain of medical images that are typically large, needs to be divided into patches for processing.

1.2 Related Works

From various challenges, benchmarks and technical papers, it is known that patch sampling strategies, such as task-adapted loss functions [3] or specific hard negative mining techniques [4], as well the choice of input patch shapes, e.g. 2.5D versus 3D patches [6–8] or the use of multi-scale patches [9,17] influence model performance. Accordingly, these are commonly reported in the challenge and benchmark papers and topics of ongoing optimization and research.

Given their scientific influence, there logically arose a debate on what shall be reported in challenge tasks and papers and how. Systematic and meta analyses of challenge tasks have brought important stimuli to this debate and helped draw practical consequences [12,15]. For example, the high influence of different ranking schemes or missing cases handling [12] have been contrasted with the low coverage with which these parameters had been reported by the challenge organizers. This let the authors not only generally advocate for the development of common best practice guidelines for challenge design and reporting, but also enabled them to identify specific issues with respect to which design and reporting shall be improved.

Before the widespread introduction of U-Net [16] and Fully-Convolutional Neural Networks [13] and since the Ciresan win at ISBI 2012 [5], the sliding window application of (classification) CNNs was state of the art for biomedical image segmentation. Even after U-Net, sliding window as an idea survived when

using U-Nets, its variants and alike for segmentation of large images. Therefore, the de-facto standard for inference on large images is the use of ordered, strided image crops.

1.3 Contributions

However, patches need not only to be sampled at training time, but also when predicting testing data (and, notably, when practically applying the trained model for clinical use).

Taking recent challenges arching over a diverse set of different imaging modalities as examples, this work shows that

- when using strided crops for sliding window inference, the stride width can have a decisive influence on the challenge metrics.
- However, taking challenge reports of the years 2016–2019 as examples, this is contrasted by how patch sampling strategies are usually reported: not. We argue that challenge reports by best practices should include the systematic reporting of evaluation time patch sampling.
- Having said that stride width *can* decisively influence challenge results, we also show that is does not have to. The differences between the scenarios are tracked back to fundamental properties of the model and dataset.
- Based on this, we propose a scheme of how to infer experimentally information on the optimal inference strategy 'on the go' when training the model.

In addition, we propose novel strategies for evaluation-time patch sampling, namely

- guided random sampling (Monte Carlo) as well as
- generalized entropy - and
- uncertainty-based inference patch sampling

and show that these can beat standard sliding window techniques in certain scenarios.

We therefore conclude that design of patch sampling strategies beyond the de-facto standard of ordered crops opens up a rich potential for inference step optimization. The focus should be widened to include it into biomedical challenge participation, analysis and beyond.

2 Materials and Methods

2.1 Datasets

We perform our experiments on three different challenge datasets that are selected in order to span a wide range of imaging modalities. These are: (i) the StructSeg 2019 challenge on organ and tumor segmentation in CT scans, (ii) the DRIVE challenge for vessel segmentation in fundoscopy images, and (iii) the PAIP 2019 challenge focusing on liver cancer segmentation in histopathology images. A detailed description is provided in the Supplementary Materials.

2.2 Segmentation Models

For our experiments in this work, we employ pre-trained segmentation models for the datasets described above. The models have either competed directly in the respective challenges and reached top-10 results or have achieved superb results when being evaluated against the challenge dataset used as a benchmark later. Details can be found in the Supplementary Materials.

2.3 Challenge Analysis

In order to assess how evaluation strategies are reported in segmentation challenge papers, we searched and filtered recent years' challenges as described in the Supplementary Materials. This procedure resulted in $N = 27$ segmentation challenges with a total of $N = 170$ different models for analysis.

2.4 Inference Framework and Experimental Setup

The evaluated segmentation models are implemented in different frameworks, namely Tensorflow 2.1.0 [2] and PyTorch 1.2.0 [14]. Using Python 3.7, we have implemented a generic evaluation framework which can load models from both frameworks and handles data loading, patch sampling, prediction stitching and metric evaluations in a framework-independent manner. Performance scores reported in this paper are based on the Dice coefficient (DC). Complementary results for the 95% Hausdorff distance (95HD) are provided in the Supplementary Materials.

2.5 Patch Sampling Strategies

Let $I : \Omega \to V^C$ be a D-dimensional image with $\Omega = [1, N_i] \times \cdots \times [1, N_D]$ and $V \subset \mathbb{R}^C$, where $C \in \mathbb{N}_+$ denotes the number of channels. An image patch P_φ of size $(s_1, \ldots, s_D)^\top$ centered at $\boldsymbol{c} \in \Omega$ can then be described as the restriction of I to $\varphi(\boldsymbol{c}) = [c_1 - s_1/2, c_1 + s_1/2] \times \cdots \times [c_D - s_D/2, c_D + s_D/2]$, i.e., $P_\varphi = I|_{\varphi(\boldsymbol{c})}$. The corresponding D'-class prediction for this patch can be analogously defined as $P_\varphi^{\text{pred}} : \varphi(\boldsymbol{c}) \to [0, 1]^{D'}$. While sampling patches by any of the strategies below, both the areas φ that have already been sampled as well as corresponding predictions P_φ^{pred} are tracked using the multisets Φ_{area} and Φ_{pred}. In contrast to sets, multisets allow for multiple instances for each element thus enable stitching potentially overlapping and duplicate patches back together.

Ordered Crops: Starting from $\boldsymbol{c} = (s_1/2, \ldots, s_D/2)^\top$, the central coordinate $\boldsymbol{c}$ is shifted by some stride $\mathbf{d} = (d_1, \cdots, d_D)^\top$. Without loss of generality, the iteration is started with the first dimension. After a full loop regarding this dimension, one step is taken into the direction of the second dimension, et cetera, until the whole image is covered.

Monte Carlo Sampling: Random Cropping with Guidance: For this strategy, the central coordinate $\boldsymbol{c}$ of a patch to be cropped is taken as a random variable X. Using a well-defined probability mass function $p_X(\boldsymbol{c}) = \mathbb{P}(X = \boldsymbol{c})$, random center points and thus random patches can be sampled. Initially, p_X is set to a uniform distribution $p_X(\boldsymbol{c}) = |\Omega|^{-1}$. Employing Φ, the so-called coverage can be defined by counting how often each pixel/voxel has already been covered by randomly sampled patches

$$N(\boldsymbol{r}) = \sum_{\varphi \in \Phi_{\text{area}}} \mathbb{1}_{\varphi}(\boldsymbol{r}), \quad \boldsymbol{r} \in \Omega. \tag{1}$$

While sampling, p_X gets updated according to

$$p_X(\boldsymbol{c}) = \frac{1}{\mathcal{N}} \frac{|\Omega|^{-1}}{N(\boldsymbol{c}) + \delta}, \quad \delta \in \mathbb{R}_+. \tag{2}$$

We refer to this term as *coverage-guidance* as it encourages sampling of patches that have a relatively low coverage. The resulting strategy is called *coverage-guided Monte Carlo sampling*. In order to obtain a valid probability mass function, we employ a variable normalization factor $\mathcal{N}^{-1} \in \mathbb{R}_+$, so that p_X sums to one.

Dynamic Uncertainty and Entropy-Inspired Sampling: The above formalism (and so the corresponding implementation) can easily be adopted in order to accommodate other ideas than coverage-guidance, just by alteration of the probability map. Exemplarily, this can be used to host dynamic, online uncertainty-dependent strategies or to make use of a generalized entropy for putting emphasis on "interesting" image regions:

To focus the sampling on more relevant areas, we base the calculation of p_X on the probability map

$$E(\boldsymbol{r}) = \frac{1}{\sum_{\boldsymbol{r} \in \Omega} \|\nabla I(\boldsymbol{r})\|_2^2} \|\nabla I(\boldsymbol{r})\|_2^2, \tag{3}$$

inspired by the gradient-based extension of Shannon's entropy for multi-dimensional images [11]. The gradient operation is discretized and approximated by second-order central differences. Additionally, we increase the re-sampling likelihood in areas where overlapping predictions differ significantly (in the sense of an overlap uncertainty). To this end, we calculate the variance for every pixel/voxel with non-zero coverage

$$\sigma^2(\boldsymbol{r}) = \frac{\sum_{P_\varphi^{\text{pred}} \in \Phi_{\text{pred}}} \left(P_\varphi^{\text{pred}}(\boldsymbol{r})\right)^2 - \frac{1}{N(\boldsymbol{r})} \left(\sum_{P_\varphi^{\text{pred}} \in \Phi_{\text{pred}}} P_\varphi^{\text{pred}}(\boldsymbol{r})\right)^2}{N(\boldsymbol{r})}. \tag{4}$$

In order to keep σ^2 independent of the number of classes D' and make the sampling method applicable to arbitrary segmentation tasks, σ^2 is divided by the

maximum possible variance $\max\{\sigma^2\} = 1/4$ and averaged over all classes. This results in the normalized uncertainty measure σ_{n}^2. Using the guidance technique (c.f. Eq. 2), the final probability mass function for patch sampling is computed by

$$p_X(\boldsymbol{c}) = \frac{1}{\mathcal{N}} \left(\frac{E(\boldsymbol{c})}{N(\boldsymbol{c}) + \delta} + \sigma_{\mathrm{n}}^2(\boldsymbol{c}) \right), \quad \delta \in \mathbb{R}_+. \tag{5}$$

2.6 Measuring Intra-patch Performance on the Go

In order to get an estimate of how the model performance varies within a patch, we calculate a pixel/voxel-wise cross entropy loss for each prediction/ground truth patch tuple in our test sets. The resulting accumulated loss map can be interpreted as the global spatially resolved performance of the predicted patches. It can, as we show below, be employed to set an optimal stride width for a sliding window evaluation strategy. In addition, it signals when other strategies might be beneficial.

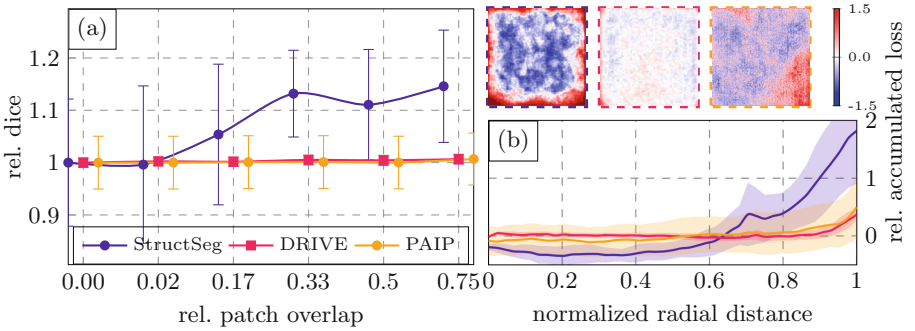

Fig. 1. (a): relative increase of the Dice score with increasing patch overlap for sliding window evaluation (mean ±95% confidence intervals). The insets give the accumulated intra-patch performances for the respective models (normalized by their means). (b) depicts the mean relative patch performances as functions of the radial distance. Lower values correspond to a better performance.

3 Results

3.1 The Stride-Width May Determine Your Challenge Performance - and Will Most Probably Not Be Reported in the Challenge Paper

Our first experiments seeks to examine whether and how variations in the stride width influence the segmentation quality as measured by the Dice coefficient (cf. Supplementary Materials for a complementary analysis using the 95HD).

In order to compare how different models and datasets benefit from increasing patch overlap, we analyse the DC relative to its value for 0 overlap (stride width equal to patch width). As depicted in Fig. 1(a), DRIVE and PAIP models do not benefit from increasing overlap. This is opposed to StructSeg, for which a relative patch overlap of 33% achieves a rise of 13.2% (in absolute terms and for our model: Dice 0.715 at zero overlap vs. 0.809 at 33% overlap). Taking the original DC scores for this challenge task as an example [1] and assuming equal sampling strategies, that relative increase would have made the 4th (0.4857) beat the original winner (0.5447) of this task without any changes to the model or training pipeline.

Figure 1(a) shows that variations in the stride width may have an enormous influence on segmentation performance in certain cases. Irrespective of this, neither sampling strategies nor stride widths are systematically reported in the challenge literature: The patch sampling strategies for all models can be inferred in 30% of the cases (6/20). The stride width, however, is not given for more then 4.1% of the models (7/170). We have not found a single challenge that reports all stride widths for all competing models.

3.2 Experimentally Determining the Optimal Patch Overlap

With the strikingly different patch-overlap effects in PAIP and DRIVE on the one hand and StructSeg on the the hand, we asked for the underlying reasons for this. The insets of Fig. 1 show the accumulated intra-patch performances for StructSeg, DRIVE and PAIP (left to right). When viewed as functions of the radial distance (Fig. 1(b)) the relative intra-patch accumulated loss reflects the behaviour that we had seen in panel (a). In fact, the StructSeg model is characterised by a relatively poor patch performance at its margin versus at its center. As compared to DRIVE and PAIP, the lateral performance slip is stronger and occurs at much smaller radii for StructSeg. The radial coordinate at which the relative patch performance deteriorates for StructSeg corresponds to a relative patch overlap of 33–50%, again suggesting an etiological relation between intra-patch performance and stride effects.

The intra-patch performance can be computed on the go via training. Therefore, it allows us to infer the necessity for a strong evaluation patch overlap already at training time. Likewise, it allows the challenge organizers to estimate the importance of stride width effects in their dataset.

It is worth noting that, by the nature of fully-convolutional neural networks, one can always find a center sub-region of the input patch for which no edge-effects occur because individual pixels are processed equally. Hence, one can theoretically compute a stride width which safely prevents edge-effects. This center region, however, is usually very small (particularly modern architectures that are designed for leveraging context information). In contrast to this conservative estimate, intra-patch performance provides a framework to experimentally determine the *necessary*, hence *optimal* overlap.

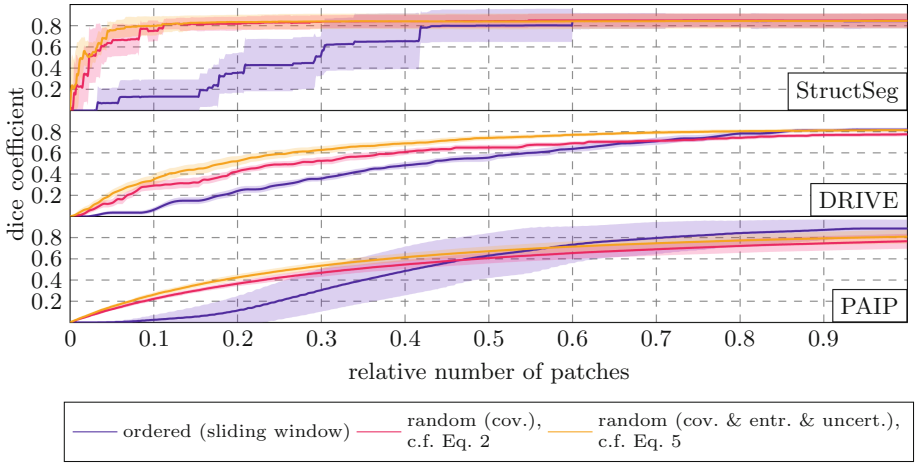

Fig. 2. Dice convergence curves for all applied patch sampling strategies and datasets/models (mean ± standard deviation).

3.3 An Attempt at Beating Sliding Window Patch-Sampling

We next evaluate the alternative patch sampling strategies (cf. Sect. 2.5) against the sliding window technique as the baseline. For the latter, we choose an optimal stride width as described in Sect. 3.1, resulting in 67% of the patch size for StructSeg. For DRIVE and PAIP, where patch overlap is of lesser influence, we chose 33% and 16.7% as examples. Figure 2 shows the Dice scores for the strategies as depending on the number of predicted patches. To allow for an inter-dataset comparison despite their different image sizes, the number of patches is given as a relative number (with one being the number of strided patches needed to cover the whole image).

First, it is evident that our proposed strategies achieve a much faster performance increase in the early patches, independent of the dataset. This has, of course, to be weighed against the computational overhead by sampling patches from the corresponding probability distribution and by updating the probability distribution itself, which we observed to be very moderate, however. On the PAIP dataset, for instance, our costliest strategy including a coverage-dependent plus and uncertainty term results in 3% overhead, when updating the probability distribution every 1% of the image. Second, the convergence curves for these strategies are stable between cases, as opposed to the sliding window approach. The latter is stable only for DRIVE, where the blood vessels as the segmented class are spread over the entire image. For localized classes, as in PAIP or StructSeg, the sliding window technique results in a much more unstable convergence curve as judged by the relatively large standard deviations. Third, entropy guidance leads to a particularly steep increase at the very first patches. The fact that this effect is independent of the specific dataset suggests that entropy guidance can robustly select the most meaningful image regions. This is opposed to

alternative, 'handcrafted' techniques for thresholding the most relevant image regions, which are specific for a single dataset. Lastly, it should be noted that the proposed strategies are more flexible: in that one can proceed arbitrarily if time permits and in that one can easily adopt further strategies. For example, the uncertainty, being an overlap uncertainty in this case, can easily be replaced by e.g. an uncertainty term from Bayesian inference.

4 Conclusions

Patch sampling strategies play an important role not only at training, but also at inference time. Therefore, reporting of evaluation time patch sampling strategies in challenge reports needs to be systematized and improved.

Patch sampling other than by strided crops holds potential in a variety of scenarios: If fast convergence is needed, strided crop strategies can easily be outperformed by a variety of strategies. Other than that, non-strided strategies allow for combination with uncertainty and Bayesian evaluation methods, and a flexible stopping or continuation of inference runs.

Generalized entropy can be used for additional guidance in order to sample informative regions. It achieves this in a dataset-independent manner, as opposed to "handcrafted" pre-segmentation by thresholds and alike.

Intra-patch performance should be monitored at training time. It provides clues about the ideal patch sampling strategy as well as the optimal stride width and can be computed on the go while training, hence *before* any evaluations and without any extra cost.

Acknowledgments. This work was supported by DFG grant WE 6197/2-1, the European Fund for Regional Development (ERDF), the Free and Hanseatic City of Hamburg, the Forschungszentrum Medizintechnik Hamburg (02fmthh2017), and Olympus Co. Hamburg. Furthermore, RS gratefully acknowledges funding by the Studienstiftung des deutschen Volkes and the Günther Elin Krempel foundation. The authors would like to thank NVIDIA for the donation of graphics cards under the GPU Grant Program.

References

1. StructSeg 2019: Automatic Structure Segmentation for Radiotherapy Planning Challenge 2019, August 2019. https://structseg2019.grand-challenge.org/Home/
2. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems, March 2016. arXiv:1603.04467
3. Abraham, N., Khan, N.M.: A novel focal Tversky loss function with improved attention U-net for lesion segmentation. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 683–687. IEEE, Venice, April 2019
4. Bian, C., et al.: Pyramid network with online hard example mining for accurate left atrium segmentation. In: Pop, M., et al. (eds.) STACOM 2018. LNCS, vol. 11395, pp. 237–245. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12029-0_26

5. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 2843–2851. Curran Associates, Inc. (2012)
6. Han, X.: Automatic liver lesion segmentation using a deep convolutional neural network method. Med. Phys. **44**(4), 1408–1419 (2017)
7. Isensee, F., Jaeger, P., Full, P.M., Wolf, I., Engelhardt, S., Maier-Hein, K.H.: Automatic cardiac disease assessment on cine-MRI via time-series segmentation and domain specific features. arXiv:1707.00587 (2018)
8. Isensee, F., Maier-Hein, K.H.: An attempt at beating the 3D U-Net. arXiv:1908.02182, October 2019
9. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. **36**, 61–78 (2017)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)
11. Larkin, K.G.: Reflections on shannon information: In search of a natural information-entropy for images. arXiv:1609.01117 (2016)
12. Maier-Hein, L., et al.: Why rankings of biomedical image analysis competitions should be interpreted with care. Nat. Commun. **9**(1), 1–13 (2018)
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
14. Paszke, A., et al.: Automatic differentiation in PyTorch. NIPS 2017. https://openreview.net/pdf?id=BJJsrmfCZ
15. Reinke, A., et al.: How to exploit weaknesses in biomedical challenge design and organization. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 388–395. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_45
16. 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
17. Schmitz, R., Madesta, F., Nielsen, M., Werner, R., Rösch, T.: Multi-scale fully convolutional neural networks for histopathology image segmentation: from nuclear aberrations to the global tissue architecture. arXiv:1909.10726, September 2019