Diffusion-based generation of Histopathological Whole Slide Images at a Gigapixel scale

Supplementary material

Robert Harb[1,2], Thomas Pock[1], Heimo Müller[2]
[1]Institute of Computer Graphics and Vision, Graz University of Technology, Austria
[2]Diagnostic and Research Institute of Pathology, Medical University of Graz, Austria
{robert.harb, pock}@icg.tugraz.at, heimo.mueller@medunigraz.at

A. Derivation of the super-resolution constraint

In the following, we show how to solve the optimization problem given in Eq. (8) using the method of Lagrangian multipliers. We begin with the problem

$$\bar{\mathbf{u}} = \arg\min_{\bar{\mathbf{u}}} \frac{1}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2 \quad \text{s.t. } \mathbf{A}\bar{\mathbf{u}} = \mathbf{y}. \tag{12}$$

To solve it, we introduce the Lagrangian

$$\mathcal{L}(\bar{\mathbf{u}}, \lambda) = \frac{1}{2}\|\mathbf{u} - \bar{\mathbf{u}}\|^2 + \lambda^T(\mathbf{A}\bar{\mathbf{u}} - \mathbf{y}), \tag{13}$$

with lagrange multipliers λ. The gradient of the lagrangian is given as

$$\nabla_{\bar{\mathbf{u}}}\mathcal{L} = \bar{\mathbf{u}} - \mathbf{u} + \mathbf{A}^T\lambda. \tag{14}$$

Furthermore, setting the gradient to zero, and solving for $\bar{\mathbf{u}}$ results in

$$\bar{\mathbf{u}} = \mathbf{u} - \mathbf{A}^T\lambda. \tag{15}$$

Inserting $\bar{\mathbf{u}}$ from Eq. (15) into the constraint gives

$$\mathbf{y} = \mathbf{A}(\mathbf{u} - \mathbf{A}^T\lambda). \tag{16}$$

By solving Eq. (16) for λ we obtain

$$\lambda = (\mathbf{A}\mathbf{A}^T)^{-1}(\mathbf{A}\mathbf{u} - \mathbf{y}). \tag{17}$$

And inserting λ from Eq. (17) into Eq. (15), gives us a solution for $\bar{\mathbf{u}}$

$$\bar{\mathbf{u}} = \mathbf{u} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}(\mathbf{A}\mathbf{u} - \mathbf{y}), \tag{18}$$

which can be simplified as

$$\begin{aligned}\bar{\mathbf{u}} &= \left(\mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\right)\mathbf{u} + \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} \\ &= \mathbf{u} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\mathbf{u} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{y}.\end{aligned} \tag{19}$$

Using the definition of the pseudoinverse $\mathbf{A}^\dagger$ for full row rank matrices

$$\mathbf{A}^\dagger = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}, \tag{20}$$

we can further simplify Eq. (19) leading to our final solution

$$\bar{\mathbf{u}} = (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{u} + \mathbf{A}^\dagger\mathbf{y}. \tag{21}$$

B. Scaling functions

In the following, we provide the full expressions of the noise level parametrized scaling functions in our diffusion model. Particularly, in our denoiser function Eq. (6)

$$D_\theta(\mathbf{x}; \sigma, s) = c_{\text{skip}}(\sigma)\,\mathbf{x} + c_{\text{out}}(\sigma)\,F_\theta\big(c_{\text{in}}(\sigma)\,\mathbf{x}; \sigma, s\big), \tag{22}$$

and our loss Eq. (7)

$$\mathbb{E}_{s,\tilde{\mathbf{x}},\sigma,\mathbf{n}}\left[\lambda(\sigma)\|D_\theta(\tilde{\mathbf{x}} + \mathbf{n}; \sigma, s) - \tilde{\mathbf{x}}\|_2^2\right], \tag{23}$$

we set

$$c_{\text{skip}}(\sigma) = \sigma_{\text{data}}^2 \,/\, \left(\sigma^2 + \sigma_{\text{data}}^2\right), \tag{24}$$

$$c_{\text{out}}(\sigma) = \sigma \cdot \sigma_{\text{data}} \,/\, \sqrt{\sigma_{\text{data}}^2 + \sigma^2}, \tag{25}$$

$$c_{\text{in}}(\sigma) = 1/\sqrt{\sigma^2 + \sigma_{\text{data}}^2}, \tag{26}$$

and

$$\lambda(\sigma) = \sigma^{-2} + \frac{1}{\sigma_{\text{data}}^2}, \tag{27}$$

where σ_{data} is the standard deviation of our training data. We set $\sigma_{\text{data}} = 0.5$, which is simply done through the normalization of training images. A detailed discussion and derivations of these noise level parametrized scaling functions are provided by Karras *et al*. [18]. In essence, the input scaling $c_{\text{in}}(\sigma)$ is set such that the inputs of F_θ have unit variance. The output scaling $c_{\text{out}}(\sigma)$ is set such that the effective training target of F_θ has unit variance. The skip-connection scaling $c_{\text{skip}}(\sigma)$ is set such that the errors of F_θ are amplified as little as possible. And the loss weighting $\lambda(\sigma)$ weighs loss terms equally across all noise levels σ.

C. Data preprocessing

When sampling patches from WSIs, we only consider patches covering at least 10% tissue area. To segment tissue from the background, we use FESI [4].

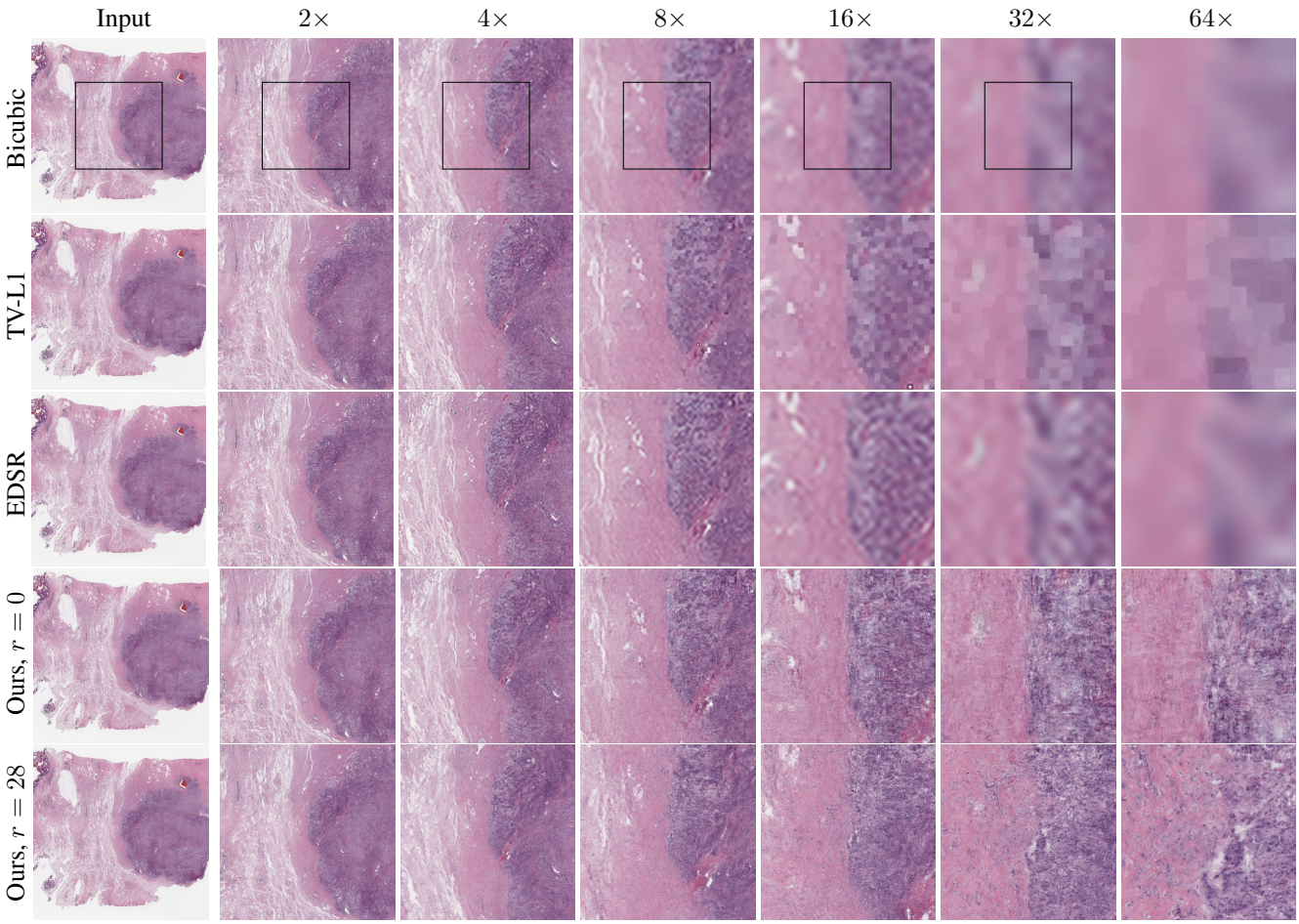

Figure 9. Comparison of our method with multiple super-resolution methods. The first column shows the input image, each subsequent column shows the upscaling result of a patch extracted from the center of the previous column. Best viewed digitally.

D. Comparison with different super-resolution approaches

In this section, we compare upscaling with our approach to established super-resolution methods. To this end, we apply multiple iterations of $2\times$ upscaling on an initial 512×512-sized image. In each iteration, we upscale a 512×512 patch extracted from the centre of the previous iteration's output. We compare with super-resolution approaches that follow multiple paradigms: TV-L1, which is not learning-based; EDSR [22], which is learning-based but not generative; and our method, which is learning-based and generative. For EDSR, we retrained the model using the same data as our method. Additionally, as a baseline, we also show bicubic interpolation.

For our method, we show results with a relaxation parameter $r = 0$ and with $r = 28$. As discussed in the main paper, without relaxation, *i.e.* $r = 0$, our method closely resembles the zero-shot super-resolution approach of DDNM [42], where the super-resolution constraint has to be satisfied strictly. Contrarily, with the relaxation parameter $r > 0$, the model is not strictly bound to the super-resolution constraint, allowing for a trade-off between consistency with the low-resolution input image and introducing new details.

Figure 9 shows the results of our comparison. TV-L1 super-resolution produces sharper results than bicubic interpolation but still gives unsatisfying results for larger magnifications. Similarly, EDSR fails to produce reasonable results for larger magnifications. The results of our method without relaxation are much sharper than TV-L1 and EDSR. However, particularly at larger magnifications, the results no longer retain the structure of histopathological images. Note how individual cells are barely visible at $64\times$ magnification. In contrast, with relaxation, even at large magnifications, results resemble the structure of histopathological images much better, e.g. individual cells are clearly distinguishable.

E. Sampling

When sampling WSIs, we segment the initial image $\mathbf{z}_0$ into tissue and background areas using FESI [4]. And then run the coarse-to-fine scheme only on patches that cover tissue area. This helps us to reduce the overall sampling time by skipping areas containing background. When stitching patches back together, we fill background patches with the background colour extracted from the segmentation.

F. Network

For the network $F_\theta(\mathbf{x}; \sigma, s)$, we used the U-Net backbone from the implementation of Karras *et al*. [18], which is based on the network of DDPM++ [38]. Tab. 3 shows the parameters we used. We did the additional conditioning with the spatial resolution s, in the same way as the noise conditioning σ is implemented in the network. Hence, we compute a sinusoidal positional encoding of the spatial resolution s in $\mu\mathrm{m/px}$ and push the result through embedding layers. We then simply add the spatial resolution embedding to the embedding of the noise and use the result for following computations instead of the plain noise embedding.

Parameter	Value
Channel multiplier	64
Channel factor per resolution	0.5-1-1-2-2-4-4
Residual blocks per resolution	2
Attention resolutions	{32-16-8}
Attention heads	4
Dropout probability	10%

Table 2. Network parameters

G. Training

Tab. 3 shows the parameters we used for training. Noise σ during training was sampled from a log-normal distribution $\ln(\sigma) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$.

Parameter	Value
Learning rate	1×10^{-4}
Optimizer	Adam
Batch size	64
σ_{min}	0.002
σ_{max}	80
ρ	7
P_{std}	-1.2
P_{max}	1.2

Table 3. Training hyperparameters

H. User study - additional discussion

Figure 11 shows a visualization of the user study results with the respective IDs for each individual WSI. We provide a download to all 20 synthetic WSIs of the user study[1]. To open the downloaded WSIs, make sure to use an appropiate viewer, *e.g*. QuPath[2]. Table 4 maps the IDs in the user study to the respective file IDs in the TCGA-BRCA dataset. Furthermore, Fig. 10 shows a screenshot of the interface we used for the study.

In addition to the discussion in the main paper, we want to add a few remarks about the user study results. Upon examining the results, one can see noticeable differences in the performance of the three pathologists when identifying the synthetic slides. The first pathologist consistently gave ratings with a high degree of uncertainty. In contrast, the other two seemed more confident in their decisions. Notably, while the first pathologist correctly identified all the slides from the TCGA as real, the third pathologist mistakenly classified a few TCGA slides as synthetic with high certainty. Even though there was a tendency for the pathologists to identify the synthetic slides, this suggests that it was not trivial for the pathologists to differentiate the images. Therefore, we conclude that most synthetic WSIs did not contain major, prominent image artefacts. This suggests that grid-shift was effective at preventing stitching artifacts and that our diffusion model did not generate completely pathologically unplausible structures.

User study ID	TCGA-BRCA ID
0	TCGA-A7-A4SD-11A-03-TS3.3781BE68-0CC3-446C-9DA9-35EC6FA954E4
1	TCGA-A7-A6VX-01Z-00-DX1.F74DA243-C65A-4997-BCA0-F1C89675978C
2	TCGA-A8-A09I-01A-02-BS2.ca9aacf2-573b-4af2-bc50-5213526eb3a3
3	TCGA-AN-A0FS-01A-01-TSA.ec030e02-fd7d-4683-803d-830ee80d8173
4	TCGA-AO-A03U-01B-02-BSB.dcb167f4-c3ab-4dcc-8f40-41c4ce453847
5	TCGA-AO-A0J5-01Z-00-DX1.20C14D0C-1A74-4FE9-A5E6-BDDCB8DE7714
6	TCGA-AR-A0TR-01Z-00-DX1.BBCE653F-7DD0-4830-BAD3-C06207A93853
7	TCGA-B6-A0IM-01A-01-BSA.e4fce1ac-0800-4e45-a3bc-f9bcb2ea825f
8	TCGA-B6-A1KC-01Z-00-DX1.4DD3E48B-F434-499F-9FF1-0FFD2883A375
9	TCGA-BH-A0BF-11A-02-TSB.6e4bf881-a29f-4fb4-b38c-5bebe44368ec
10	TCGA-BH-A0DD-11A-01-BSA.e9aae98d-ecf8-4d48-b1ca-f349013f2c42
11	TCGA-C8-A27B-01Z-00-DX1.5A8A14E8-6430-4147-9C71-805024E098CB
12	TCGA-C8-A8HP-01A-01-TSA.C1048607-5CC7-4798-AA55-55C78B31C10D
13	TCGA-E2-A15H-01A-01-TSA.6ba57309-1e15-4a84-98ad-5e8f02688a96
14	TCGA-E2-A15M-01A-01-TSA.41d14b10-8567-4f43-a5a8-b952d859c70f
15	TCGA-E9-A229-01Z-00-DX1.5B448B88-DA0C-44FF-87B3-20649A4A26FE
16	TCGA-EW-A1OX-01A-01-TSA.74283185-7c47-44ce-8904-1a121870104e
17	TCGA-EW-A1P5-01A-01-TSA.0fdc58ed-1cbd-4f60-839e-c12e1450e431
18	TCGA-GM-A2DI-01A-03-TSC.DB9E24D8-2B07-483E-A490-2B64240EFCEE
19	TCGA-OL-A66O-01Z-00-DX1.5F1E4C60-5CE8-41B4-A94D-4AA80D9253F9

Table 4. Mapping between the WSI IDs in the user study and their IDs in the TCGA-BRCA dataset.

I. Downscaling operator

In the following, we show how the average-pooling operator $\mathbf{A}$ and its pseudoinverse $\mathbf{A}^\dagger$ from Eq. (9) can be implemented in PyTorch [42].

[1] https://drive.google.com/file/d/1VpNFGgcw2iEYY4cbHrskQsjjwHMy47A9/view?usp=sharing

[2] https://qupath.github.io/

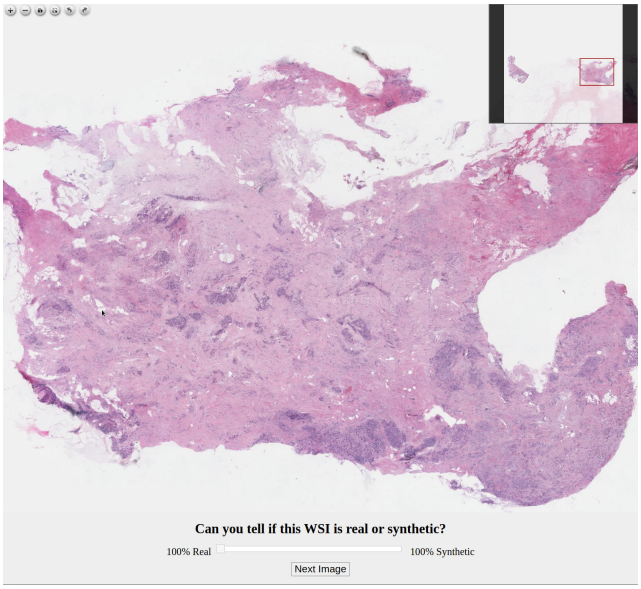

Figure 10. Screenshot of our user study interface. The participants could freely navigate the shown WSIs through their full magnification range.

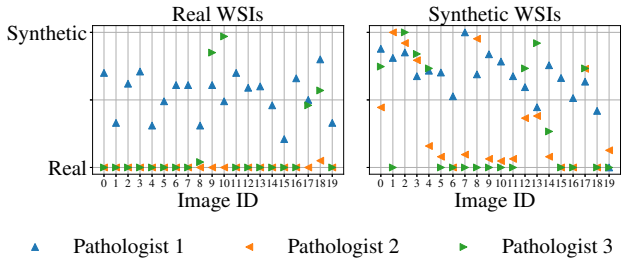

Figure 11. User study results with Image IDs. The IDs of the synthetic WSIs correspond to the filenames in the provided download, and Tab. 4 maps the IDs of the real WSIs to the respective IDs in the TCGA-BRCA dataset.

```
def PatchUpsample(x, scale):
    n,c,h,w = x.shape
    x = torch.zeros(n,c,h,scale,w,scale) + x.view(n,c,h,l,w,l)
    return x.view(n,c,scale*h,scale*w)

A = torch.nn.AdaptiveAvgPool2d(())
Ap = lambda z: PatchUpsample(z, scale)
```

J. Additional examples

In the following, we show additional WSIs generated by our method. The shown patches are resized to 1024×1024. To get a full impression about the quality of the generated WSIs, download the full-resolution WSIs from the user study.

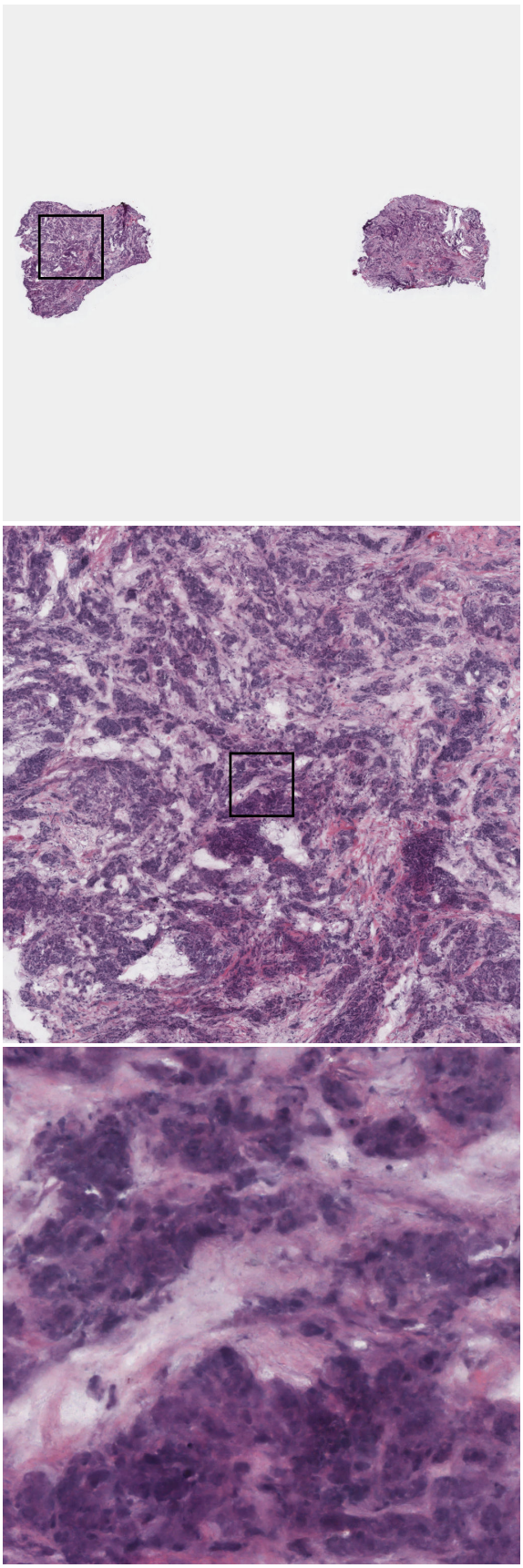

Figure 12. Synthetic WSI

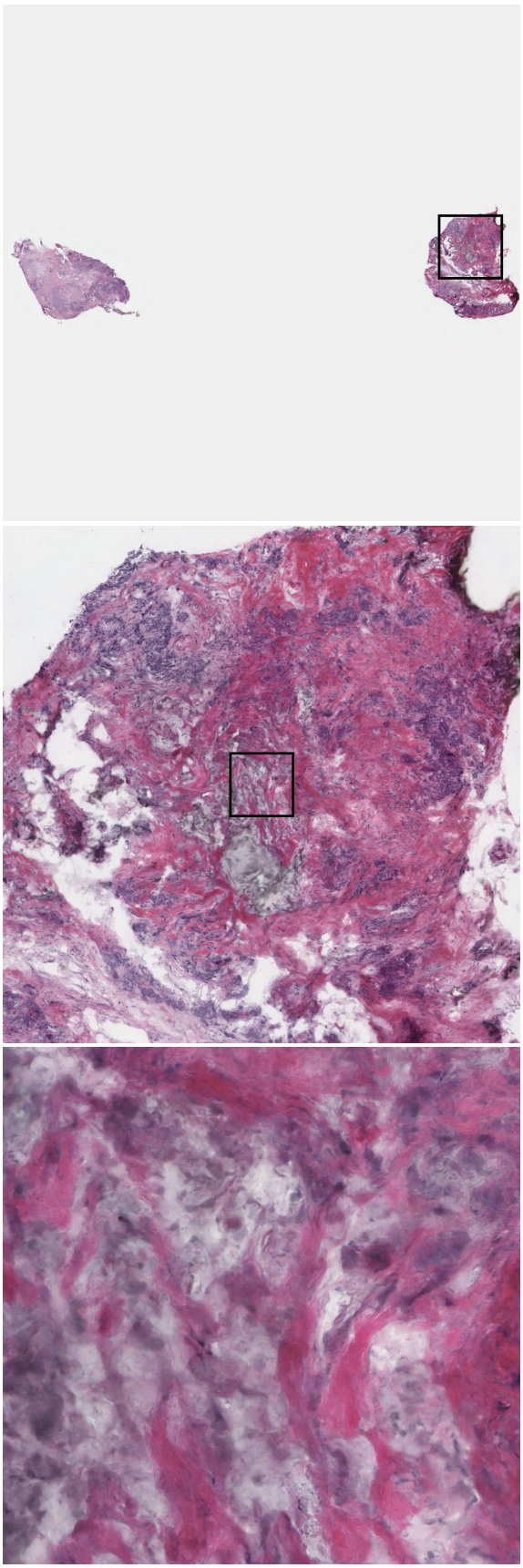

Figure 13. Synthetic WSI

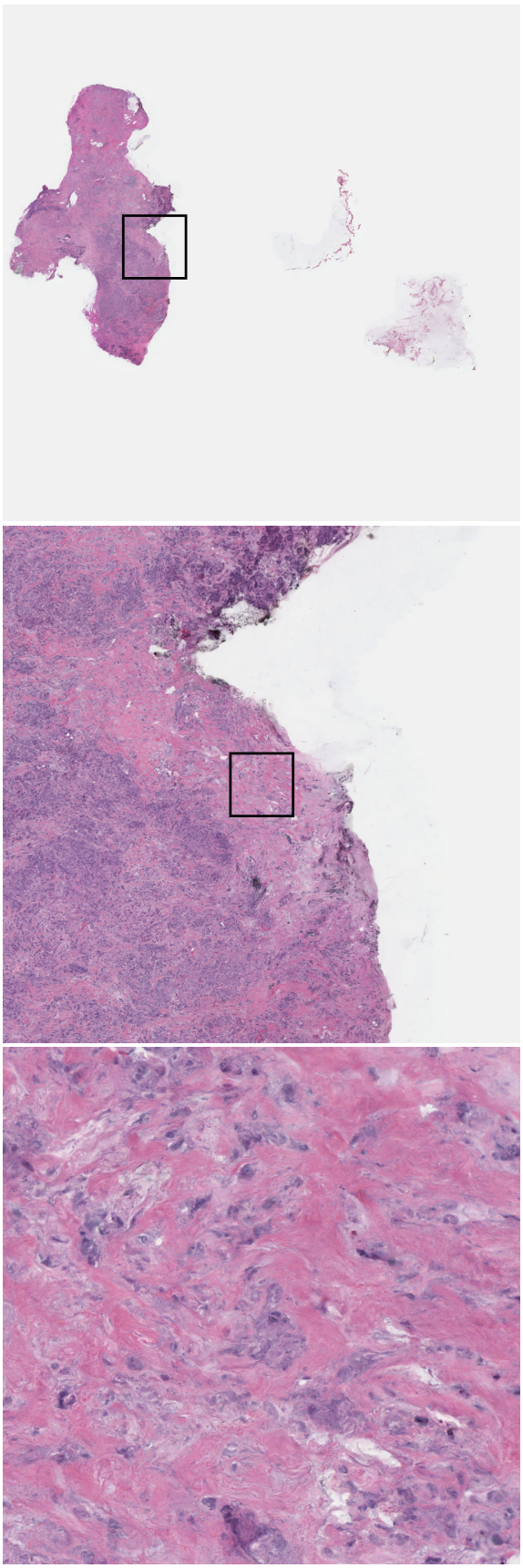

Figure 14. Synthetic WSI

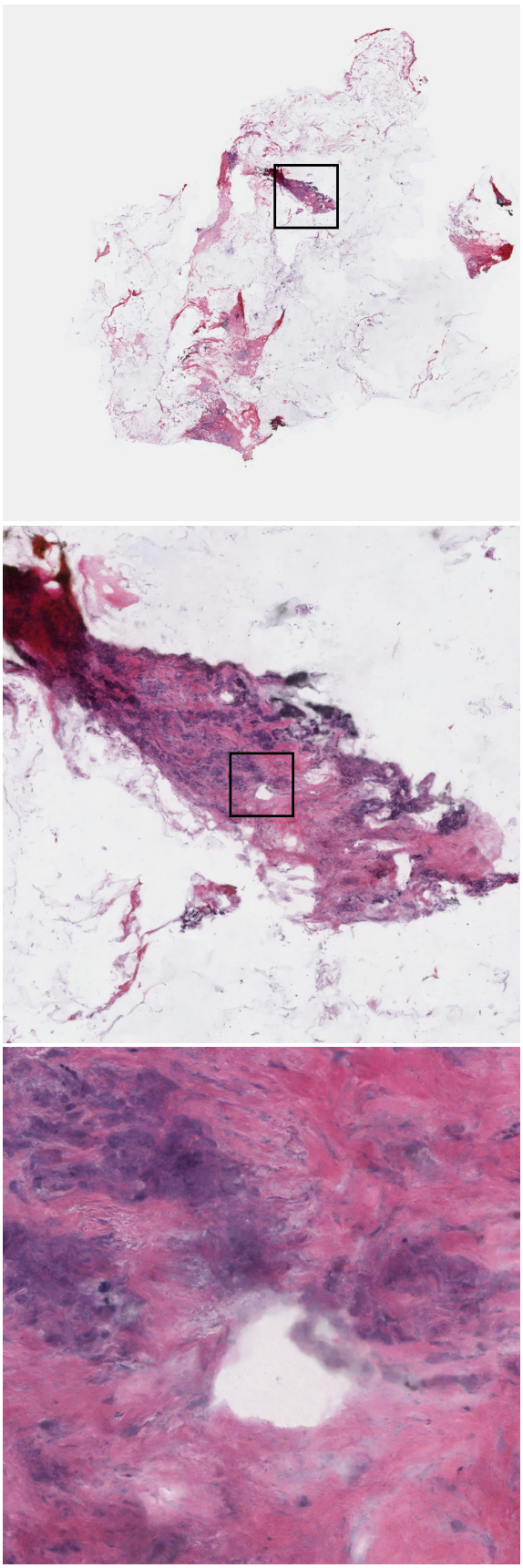

Figure 15. Synthetic WSI

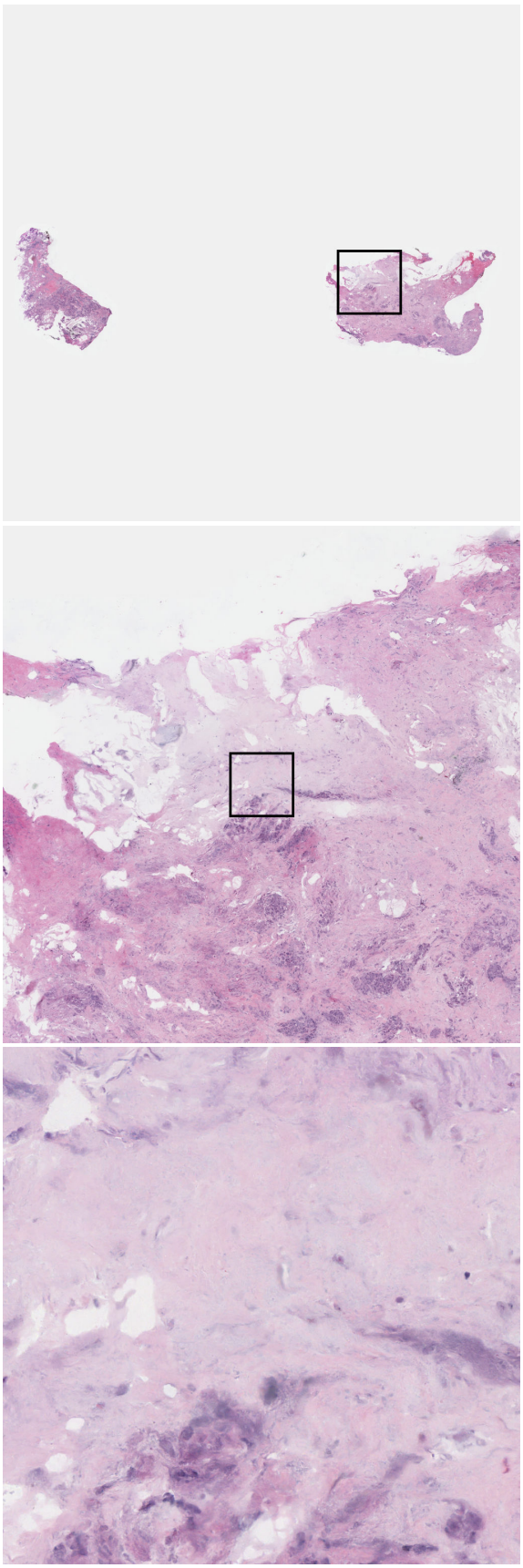

Figure 16. Synthetic WSI

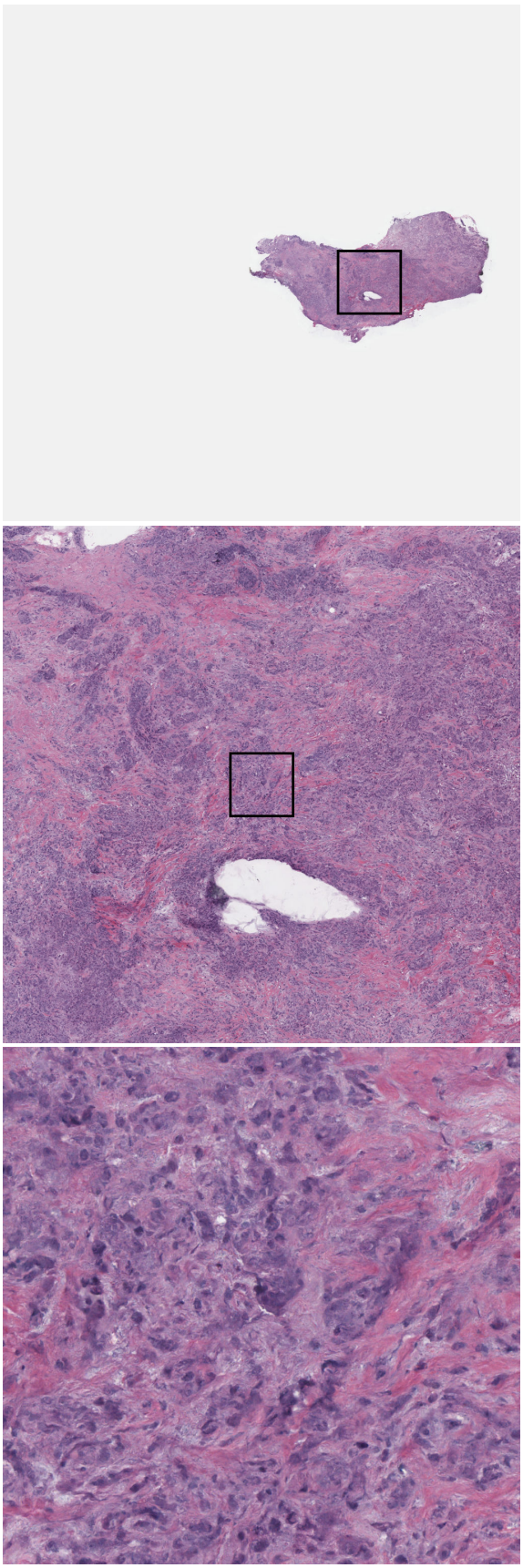

Figure 17. Synthetic WSI

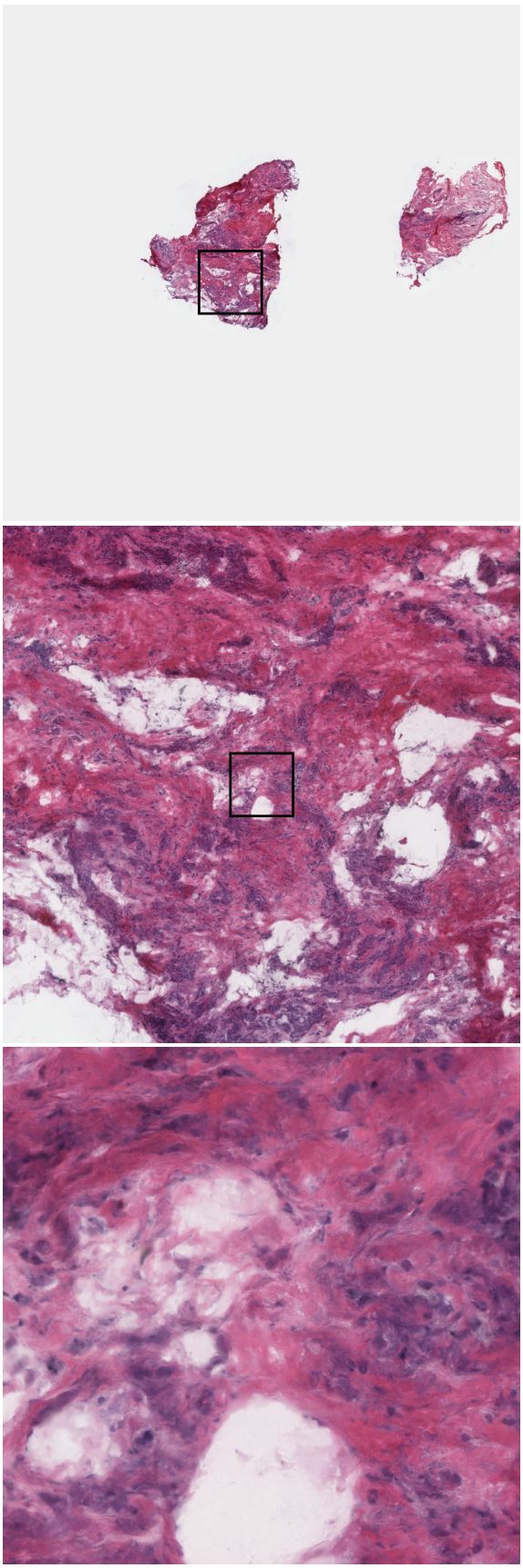

Figure 18. Synthetic WSI

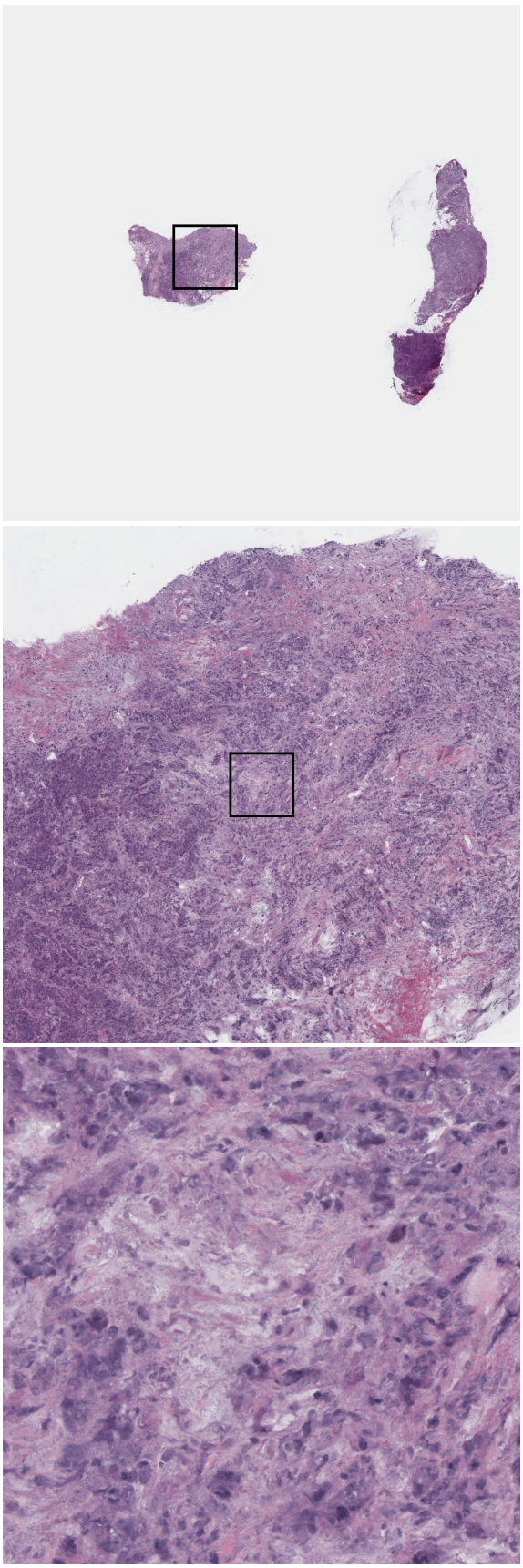

Figure 19. Synthetic WSI

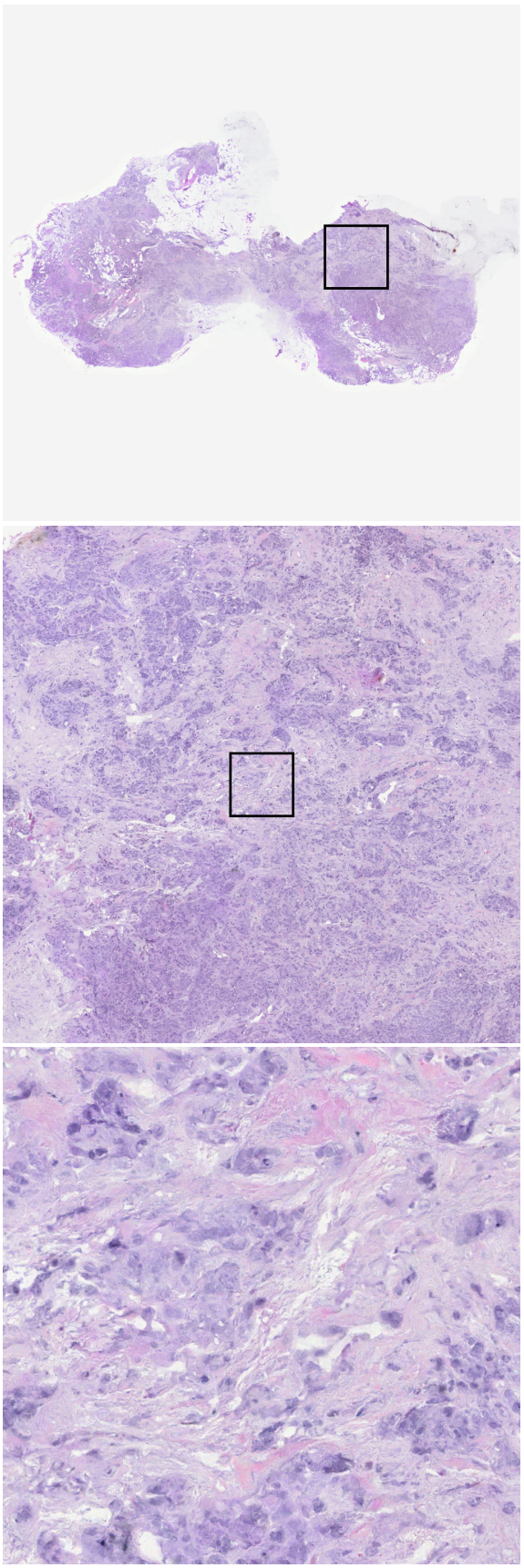

Figure 20. Synthetic WSI