CellViT: Vision Transformers for Precise Cell Segmentation and Classification

Fabian Hörst[1,2*], Moritz Rempe[1,2], Lukas Heine[1,2], Constantin Seibold[1,3], Julius Keyl[1,4], Giulia Baldini[1,5], Selma Ugurel[6,7], Jens Siveke[8,9,10,11], Barbara Grünwald[12,13], Jan Egger[1,2], and Jens Kleesiek[1,2,7,14]

[1]Institute for AI in Medicine (IKIM), University Hospital Essen (AöR), Essen, Germany
[2]Cancer Research Center Cologne Essen (CCCE), West German Cancer Center Essen, University Hospital Essen (AöR), Essen, Germany
[3]Clinic for Nuclear Medicine, University Hospital Essen (AöR), Essen, Germany
[4]Institute of Pathology, University Hospital Essen (AöR), Essen, Germany
[5]Institute of Interventional and Diagnostic Radiology and Neuroradiology, University Hospital Essen (AöR), Essen, Germany
[6]Department of Dermatology, University Hospital Essen (AöR), Essen, Germany
[7]German Cancer Consortium (DKTK, Partner site Essen), Heidelberg, Germany
[8]Bridge Institute of Experimental Tumor Therapy, West German Cancer Center Essen, University Hospital Essen (AöR), Essen, Germany
[9]Division of Solid Tumor Translational Oncology, German Cancer Consortium (DKTK, Partner site Essen) and German Cancer Research Center (DKFZ), Heidelberg, German
[10]West German Cancer Center, Department of Medical Oncology, University Hospital Essen (AöR), Essen, Germany
[11]Medical Faculty, University Duisburg-Essen, Essen, Germany
[12]Department of Urology, West German Cancer Center, University Hospital Essen (AöR), Germany
[13]Princess Margaret Cancer Centre, Toronto, Ontario, Canada
[14]Department of Physics, TU Dortmund University, Dortmund, Germany

Abstract

Nuclei detection and segmentation in hematoxylin and eosin-stained (H&E) tissue images are important clinical tasks and crucial for a wide range of applications. However, it is a challenging task due to nuclei variances in staining and size, overlapping boundaries, and nuclei clustering. While convolutional neural networks have been extensively used for this task, we explore the potential of Transformer-based networks in this domain. Therefore, we introduce a new method for automated instance segmentation of cell nuclei in digitized tissue samples using a deep learning architecture based on Vision Transformer called CellViT. CellViT is trained and evaluated on the PanNuke dataset, which is one of the most challenging nuclei instance segmentation datasets, consisting of nearly 200,000 annotated Nuclei into 5 clinically important classes in 19 tissue types. We demonstrate the superiority of large-scale in-domain and out-of-domain pre-trained Vision Transformers by leveraging the recently published *Segment Anything Model* and a ViT-encoder pre-trained on 104 million histological image patches - achieving state-of-the-art nuclei detection and instance segmentation performance on the PanNuke dataset with a mean panoptic quality of 0.51 and an F_1-detection score of 0.83. The code is publicly available at https://github.com/TIO-IKIM/CellViT.

Keywords Cell Segmentation · Digital Pathology · Deep Learning · Computer Vision · Vision Transformer · Segment Anything

1 Introduction

Cancer is a severe disease burden worldwide, with millions of new cases yearly and ranking as the second leading cause of death after cardiovascular diseases [1]. Despite novel and powerful non-invasive radiological imaging modalities, collecting tissue samples and evaluating them with a microscope remains a standard procedure for diagnostic evaluation. A pathologist can draw conclusions about potential therapeutic approaches or use them as a starting point for further investigations by identifying abnormalities within the tissue. One crucial component is the analysis of the cells and their distribution within the tissue, such as detecting tumor-infiltrating lymphocytes [2] or inflammatory cells in the tumor microenvironment [3, 4]. However, large-scale analysis on the cell level is time-consuming and suffers from a high intra- and inter-observer variability.

Due to the development of high-throughput scanners for pathology, it is now possible to create digitized tissue samples (whole-slide images, WSI), enabling the application of computer vision (CV) algorithms. CV facilitates automated slide analysis, for example, to create tissue segmentation [5], detect tumors [6], evaluate therapy response [7], and the computer-aided detection and segmentation of cells [8, 9]. In addition to the clinical applications mentioned above, cell instance segmentation can be leveraged for downstream deep learning tasks, as each WSI contains numerous nuclei of diverse types, fostering systematic analysis and predictive insights [10]. Sirinukunwattana et al. [11] showed that cell analysis supports the creation of high-level tissue segmentation based on cell composition. Corredor et al. [12] used hand-crafted features extracted from cells to detect tumor regions in a slide. Existing algorithms for analyzing WSI [6, 13, 7] are often based on Convolutional Neural Networks (CNNs) used as feature extractors for image regions. The algorithms, despite achieving clinical-grade performance [13], face limitations in interpretability, which in turn poses challenges in defining novel human-interpretable biomarkers. However, accurate cell analysis within these slides presents an opportunity to construct explainable pipelines, incorporating human-interpretable features effectively in downstream tasks [10, 14]. Nevertheless, since subtask WSI analysis models [6, 13, 7] rely on abstract entity embeddings, features must be extracted from the detected cells. One approach is to generate hand-crafted features, such as morphological attributes, from the segmentation [15, 16]. In the radiology setting, this is referred to as Radiomics [17]. Alternatively, employing a CNN on image

*Corresponding author: fabian.hoerst@uk-essen.de

sections of single cells can derive deep learning features. While hand-crafted features may have limited performance, using CNNs for each cell is computationally complex. Thus, the need for automated and reliable detection and segmentation of cells in conjunction with cell-feature extraction in WSI is evident.

We developed a novel deep learning architecture based on Vision Transformer for automated instance segmentation of cell nuclei in digitized tissue samples (**CellViT**). Our approach eliminates the need for additional computational effort for deriving cell features via parallel feature extraction during runtime. The CellViT model proves to be highly effective in collecting nuclei information within patient cohorts and could serve as a reliable nucleus feature extractor for downstream algorithms. Our solution demonstrates exceptional performance on the PanNuke [18] dataset by leveraging transfer learning and pre-trained models [19, 20]. The PanNuke dataset contains 189,744 segmented nuclei and includes 19 different types of tissues. Among these tissues, there are five clinically important nuclei classes: Neoplastic, inflammatory, epithelial, dead, and connective/soft cells. In addition to the high number of tissue classes and nuclei types, the dataset is highly imbalanced, creating additional complexity. Besides class imbalance, segmenting cell nuclei itself is a difficult task. The cell nuclei may overlap, have a high level of heterogeneity and inter- or intra-instance variability in shape, size, and staining [9]. Sophisticated training methods such as transfer learning, data augmentation, and specific training sampling strategies next to post-processing algorithms are necessary to achieve satisfactory results.

The proposed network architecture is based on a U-Net-shaped encoder-decoder architecture similar to HoverNet [8], one of the leading models for nuclei segmentation. Notably, we replace the traditional CNN-based encoder network with a Vision Transformer, inspired by the UNETR architecture [21]. This approach is depicted in Figure 1. Vision Transformers are token-based (16 px × 16 px token size) neural networks that use the attention mechanism to capture both local and global context information. This ability enables ViTs to understand relationships among all cells in an image, leveraging long-range dependencies and substantially improving their segmentation. Moreover, when using the common token sizes of 16 px and input magnifications such as ×40 or ×20 of the images, the token size of ViTs is approximately equivalent to that of a cell, enabling a direct association between a detected cell and its corresponding token embedding from the ViT encoder. As a results, we directly obtain a localizable feature vector, unlike CNN networks.
Given the limited amount of available data in the medical domain, pre-trained models are an essential requirement as ViTs have increased data requirement compared to CNNs. Chen et al. [19] recently published a ViT pre-trained on 104 million histological images (ViT_{256}). Their network outperformed current state-of-the-art (SOTA) cancer subtyping and survival prediction methods. Another important contribution is the *Segment Anything Model (SAM)*, proposed by Kirillov et al. [20]. They developed a generic segmentation network for various image types, whose zero-shot performance is almost equivalent to many supervised trained networks. In our work, we compare the performance of pre-trained ViT_{256} [19] and SAM [20] models

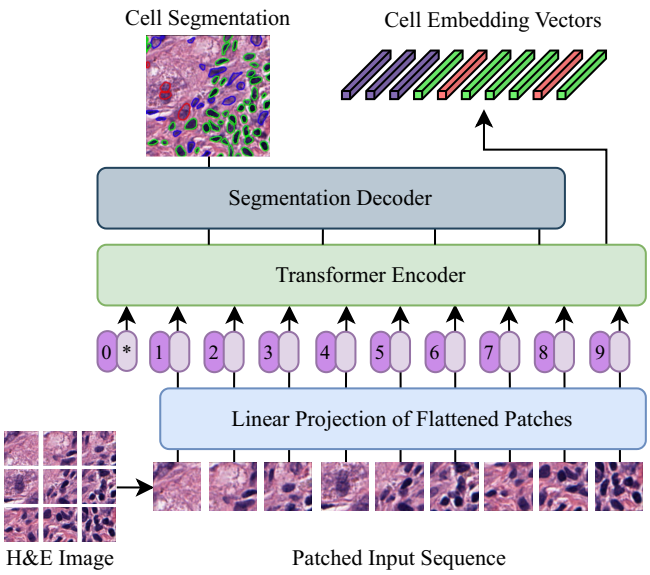

Figure 1: Network structure of CellViT. An input image is transformed into a sequence of tokens (flattened patches). By using skip connections at multiple encoder depth levels and a dedicated upsampling decoder network, precise nuclei instance segmentations are derived. Nuclei embeddings are extracted from the Transformer encoder.

as building blocks of our architecture for nuclei segmentation and classification. We demonstrate superior performance over existing nuclei instance segmentation models. We summarize our contributions as follows:

1. We present a novel U-Net-shaped encoder-decoder network for nuclei instance segmentation, levering Vision Transformers as encoder networks. Our approach surpasses existing methods for nuclei detection by a substantial margin and achieves competitive segmentation results with other state-of-the-art methods on the PanNuke dataset.

2. We are the first to employ Vision Transformer Networks for nuclei instance segmentation on the PanNuke dataset, demonstrating their effectiveness in this domain. The proposed approach combines pre-trained ViT encoders with a decoder network connected by skip connections. To further enhance the performance, we introduce a new weighted sampling strategy.

3. We demonstrate the generalizability of CellViT by applying it to the MoNuSeg dataset without finetuning.

4. We provide a framework that enables fast inference results applied on Gigapixel WSI by using a large inference patch size of 1024 × 1024 px in contrast to conventional 256 px-sized patches. Our approach generates localizable deep features, potentially valuable for downstream tasks.

2 Related Work

2.1 Instance Segmentation of Nuclei

Numerous methods have been developed to solve the challenging task of cell nuclei instance segmentation in WSIs. Previous works have explored diverse approaches, ranging from traditional image processing techniques to deep learning (DL) methods. Commonly used image processing techniques involve the design and extraction of domain-specific features. These features encompass characteristics such as intensity, texture, shape, and morphological properties of the nuclei. The primary challenge is separating overlapping nuclei, and different techniques have been devised to do this [22, 23, 24, 25, 26, 27, 28, 29]. For instance, the works of Cheng and Rajapakse [25], Veta et al. [26], and Ali and Madabhushi [27] rely on a predefined nuclei geometry and the watershed algorithm to separate clustered nuclei, while Wienert et al. [28] used morphological operations without watershed and Liao et al. [29] utilized eclipse-fitting for cluster separation. A common drawback of these techniques is their dependency on hand-crafted features, which require expert-level domain knowledge, have limited representative power, and are sensitive to hyperparameter selection [8, 30]. The complexity of extracting meaningful features increases when cell nuclei classification is added to the segmentation task. Consequently, their performance is insufficient for our needs to classify and segment nuclei in various tissue types [30].

To overcome the limitations of traditional image processing techniques, DL has emerged as a powerful approach for nuclei instance segmentation. An inherent advantage of DL networks is their automatic extraction of relevant features for the given task, surpassing the need for expert-level domain knowledge to generate hand-crafted features. DL algorithms, particularly convolutional neural networks (CNNs) [31, 32], have shown remarkable success in various computer vision tasks [33]. Especially the invention of the U-Net architecture by Ronneberger et al. [34] has significantly impacted medical image analysis by enabling accurate and efficient segmentation of complex structures, contributing to advancements in various medical domains such as radiology [35, 36] and digital pathology [37]. It consists of a U-shaped encoder-decoder structure with skip connections at multiple network depths to preserve fine-grained details in the decoder. However, the original U-Net implementation is not able to separate clustered nuclei [8]. Therefore, specialized network architectures are necessary to separate clustered and overlapping cell nuclei. In the current literature, DL algorithms for nuclei instance segmentation are further divided into two-stage and one-stage methods [9].

Two-stage methods incorporate a cell detection network in the first stage to localize cell nuclei within an image, generating bounding box predictions of nuclei. These detected nuclei are then passed on to a subsequent segmentation stage to retrieve a fine-grained nucleus segmentation. Mask-RCNN [38] is one of the leading two-stage models built on top of the object detection model Fast-RCNN [39]. Koohbanani et al. [40] utilized Mask-RCNN networks for nuclei instance segmentation. Based on the proposed nuclei detections in the first stage, the model incorporates a segmentation branch for the fine-grained nucleus segmentations in the second stage. A rectangular image section of the detected nuclei is used as input for the segmentation stage, which causes the problem that overlapping neighboring nuclei may be segmented as well and needs to be cleaned up by an additional post-processing algorithm. Another two-stage method for nuclei segmentation is BRP-Net [41], which creates nuclei proposals in the first place, then refines the boundary, and finally creates a segmentation out of this. However, this network structure is computationally complex and not designed for end-to-end training due to three independent stages. Additionally, the network requires a considerable time of 12 minutes to segment a 1360×1024 px image, making its practical application nearly impossible [41]. While two-stage systems offer advantages in localizing cells and improving individual nucleus detection, they often require additional post-processing for segmentation and suffer from time and computational complexity.

In comparison, one-stage methods combine a single DL network with post-processing operations. Micro-Net [42] extends the U-Net by using multiple resolution input images to be invariant against nuclei of varying sizes. The DIST model by Naylor et al. [43] adds an additional decoder branch next to the segmentation branch to detect nuclei markers for a watershed post-processing algorithm. For this, they predict distance maps from the nucleus boundary to the center of mass of the nuclei. Distance maps are regression maps indicating the distance of a pixel to a reference point, e.g., from a nuclei pixel to the center of mass. HoVerNet [8], one of the current SOTA methods for automatic nuclei instance segmentation, uses horizontal and vertical distances of nuclei pixels to their center of mass and separates the nuclei by using the gradient of the horizontal and vertical distance maps as an input to an edge detection filter (Sobel operator). The models STARDIST [44] and its extension CPP-Net [30] generate polygons defining the nuclei boundaries over a set of predicted distances. For this, STARDIST utilizes a star-convex polygon representation to approximate the shape of nuclei. Whereas in STARDIST, the polygons are derived just by features of the centroid pixel, CPP-Net uses context information from sampled points within a nucleus and proposes a shape-aware perceptual loss to constrain the polygon shape. STARDIST demonstrates comparable segmentation performance to HoVerNet, while CPP-Net exhibits slightly superior results.

In contrast, boundary-based methods such as DCAN [45] and TSFD-Net [9] adopt a different approach, where instead of using distance maps, watershed markers, or polygon predictions, they directly predict the nuclear contour using a prediction map. While DCAN is based on the U-Net architecture, TSFD-Net utilizes a Feature Pyramid Network (FPN) [46] to leverage multiple scales of features. Additionally, the authors of TSFD-Net introduce a tissue-classifier branch to learn tissue-specific features and guide the learning process. To address the class imbalance across nuclei and tissue types, they employ the focal loss [47] for the tissue detection branch, a modified cross-entropy loss with dynamic scaling, and the Focal Tversky loss [48] for the segmentation branch, which enlarges the contribution of challenging regions. While TSFD-Net shows promising results, its comparability to other methods is limited due to the lack of a standardized evaluation procedure.

2.2 Vision Transformer

All promising DL models [38, 41, 42, 43, 44, 8, 30, 45, 9] for nuclei instance segmentation mentioned previously are based on CNNs. Even though CNN models have demonstrated their effectiveness in image processing, they are bound to local receptive fields and may struggle to capture spatial long-range

relationships [5]. Inspired by the Transformer architecture in NLP [49], Vision Transformers [50] have recently emerged as an alternative to CNNs for computer vision [51]. Their architecture is based on the self-attention mechanism [49], allowing the model to attend to any region within an image to capture long-range dependencies. Unlike CNNs, they are also not bound to fixed input sizes and can process images of arbitrary sizes depending on computational capacity. Vision Transformers have shown promising results not only in image classification [50, 51, 52], but also in other vision tasks such as object detection [53] and semantic segmentation [21, 5].

Vision Transformers for Instance Segmentation In recent years, various ideas to use the Transformer architecture for instance segmentation have been developed [54, 55, 21, 56, 57, 58]. Primarily, these methods integrate Transformer models into encoder-decoder architectures by exchanging or extending the encoder network of existing U-Net-based solutions. Chen and Yu [54] used a Transformer in their TransUNet network to encode tokenized patches from a CNN feature map as the input sequence to derive global context within the CNN network. Li et al. [55] applied a squeeze-and-expansion Transformer as a variant of the original Vision Transformer by Dosovitskiy et al. [50] for medical segmentation. The Segformer model by Xie et al. [57] incorporates an adapted Transformer as an image encoder connected to a lightweight MLP decoder segmentation head. In contrast to these methods, the SETR model [58], used the original ViT as encoder and a fully convolution network as decoder, both connected without intermediate skip connections. Building upon these advancements, the UNETR model [21] combined a standard ViT connected to a U-Net-like decoder with skip connections, outperforming TransUNet and the SETR model on three medical image segmentation datasets. The integration of the original ViT implementation without adaptions into the powerful U-Net framework allows the use of pre-trained ViT-networks, which is an important property exploited in our work.

Large-scale Pre-Training Pre-training a Vision Transformer on a large amount of data serves as a crucial step to initialize the model's parameters with meaningful representations. Dosovitskiy et al. [50] demonstrated that ViTs require a larger amount of data compared to CNNs to learn meaningful representations. This is attributed to the inductive biases of the receptive fields of CNNs that are useful for smaller datasets. In contrast, ViTs need to learn relevant patterns, but when provided with sufficiently large datasets, they are more meaningful [52]. In the medical domain, where annotated data is often limited, pre-trained ViT-based networks become even more critical. By utilizing self-supervised pre-training approaches [59, 60, 61, 62, 63, 51], available unlabelled data can be facilitated effectively initialize network weights before finetuning the network on the target domain. One popular self-supervised pre-training approach, specifically adapted for Vision Transformers, is DINO (knowledge distillation with no labels) [51]. Vision Transformers trained with this method contain features that explicitly include information about the semantic segmentation of images, which does not emerge as clearly with CNNs [51].
In the histopathological domain, Chen et al. [19] developed a hierarchical network for slide-level representation by stacking multiple ViT blocks. Their approach involves a three-stage hierarchical architecture performing a bottom-up aggregation, with each stage pre-trained independently with DINO. The first stage focuses on processing 16×16 px-sized visual tokens out of 256×256 px patches to create a local cell-cluster token. This first stage ViT, which we refer to as $\mathbf{ViT_{256}}$ (ViT-Small, 21.7 M parameter), is particularly relevant for semantic segmentation. The authors pre-trained the ViT_{256} on 104 million 256×256 px-sized histological image patches from The Cancer Genome Atlas (TCGA) and made the network weights publicly available. It was demonstrated that the ViT_{256} network successfully learned visual concepts specific to histopathological tissue images, including fine-grained cell locations, stroma, and tumor regions, making the model a powerful pre-trained backbone network for histological image analysis.
As for the "natural image"-domain, Kirillov et al. [20] recently published a promptable open-source segmentation model as a "foundation model" [64] for semantic segmentation, also known as **Segment Anything (SAM)**. The SAM framework comprises an image encoder (ViT) and a lightweight mask decoder network. The final backbone (ViT-H) of SAM was trained supervised on 1.1 billion segmentation masks from 11 million images. A three-stage data engine consisting of assisted manual, semi-automatic, and automatic mask generation acquired this extensively annotated dataset. Pre-trained weights for three different ViT-scales (ViT-Base with 86 M parameter, denoted as SAM-B, ViT-Large with 307 M parameter, denoted as SAM-L, and ViT-Huge with 632 M parameter, denoted as SAM-H) are publicly available.

3 Methods

Our architecture is inspired by the UNETR model [21] for 3D volumetric images, but we adapt its architecture for processing 2D images as shown in Fig. 2. Unlike traditional segmentation networks that employ a single decoder branch for computing the segmentation map, our network employs three distinct multi-task output branches inspired by the approach of HoverNet [8]. The first branch predicts the binary segmentation map of all nuclei (nuclei prediction, NP), capturing their boundaries and shapes. The second branch generates horizontal and vertical distance maps (horizontal-vertical prediction, HV), providing crucial spatial information for precise localization and delineation. Lastly, the third branch predicts the nuclei type map (NT), enabling the classification of different nucleus types. In summary, our network has the following multi-task branches for instance segmentation:

- NP-branch: Predicts binary nuclei map
- HV-branch: Predicts the horizontal and vertical distances of nuclear pixels to their center of mass, normalized between -1 and 1 for each nuclei
- NT-branch: Predicts the nuclei types as instance segmentation maps

To integrate these outputs, we utilize additional postprocessing steps. These steps involve merging the information from the different branches, separating overlapping nuclei to ensure accurate individual segmentation, and determining the nuclei class based on the nuclei type map.

3.1 Network Structure

In our network, we utilize a Vision Transformer as an encoder that is connected to an upsampling decoder network via skip connections. This architecture allows us to leverage the strengths of a Vision Transformer as an image encoder for instance segmentation without losing fine-grained information. Even though many other adaptions of the U-Net structure for Vision Transformers have been proposed (e.g., SwinUNETR [56]), it was important for us to choose a network structure that incorporates the original ViT structure by Dosovitskiy et al. [50] without modifications such that we can make use of the large-scale pre-trained ViTs ViT_{256} and SAM.

As in NLP [49], Vision Transformers takes as input a 1D sequence of tokens embeddings [50, 49]. Therefore we need to divide an input image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$ with height H, width W and C input channels into a sequence of flattened patches $\boldsymbol{x}_\text{p} \in \mathbb{R}^{N\times(P^2\cdot C)}$. Each patch is a squared image section with the dimension $P \times P$. Thus, the number of patches N can be calculated via $N = HW/P^2$, which is the effective input sequence length [21]. Accordingly, a linear projection layer $\boldsymbol{E} \in \mathbb{R}^{N\times D}$ is used to map the flattened patches $\boldsymbol{x}_\text{p}$ into a D-dimensional latent space. The latent vector size D remains constant through all of the Transformer layers. In contrast to the UNETR-network, we incorporate a learnable class token $\boldsymbol{x}_\text{class}$ [50], which we can use for classification tasks and append to the patch sequence.

Unlike CNNs, which inherently capture spatial relationships through their local receptive fields, Transformers are permutation invariant and, therefore, cannot capture spatial relationships. Thus, a learnable 1D positional embedding $\boldsymbol{E}_\text{pos} \in \mathbb{R}^{(N+1)\times D}$ is added to the projected patch embeddings to preserve spatial context [21]. In summary, the final input sequence z_0 for the Transformer encoder is:

$$z_0 = \left[\boldsymbol{x}_\text{class}; \boldsymbol{x}_\text{p}^1\boldsymbol{E}; \boldsymbol{x}_\text{p}^2\boldsymbol{E}; \ldots; \boldsymbol{x}_\text{p}^N\boldsymbol{E}\right] + \boldsymbol{E}_\text{pos}. \tag{1}$$

The Transformer encoder comprises alternating layers of multiheaded self-attention (MHA) [50] and multilayer perceptrons (MLP), assembled in one Transformer block. A ViT is composed of several stacked Transformer blocks such that the latent tokens $\boldsymbol{z}_i$ are calculated by

$$z_i' = \text{MHA}(\text{Norm}(z_{i-1})) + z_{i-1},\ i = 1 \ldots L \tag{2}$$

$$z_i = \text{MLP}(\text{Norm}(z_{i-1})) + z_{i-1},\ i = 1 \ldots L, \tag{3}$$

with L denoting the number of Transformer blocks, $\text{Norm}(\cdot)$ denoting layer normalization, and i is the intermediate block identifier [21]. Inspired by the U-Net and UNETR architectures, we add skip connections to leverage information at multiple encoder depths in the decoder. In total, we use five skip connections. The first skip connection takes $\boldsymbol{x}$ as input and processes it by two convolution layers (3×3 kernel size) with batch-normalization and ReLU activation functions. For the remaining four skip connections, the intermediate and bottleneck latent tokens $\boldsymbol{z}_j,\ j \in \left\{\frac{L}{4}, \frac{2L}{4}, \frac{3L}{4}, L\right\}$ are extracted without the class token and reshaped to a 2D tensor $\boldsymbol{Z}_j \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}\times D}$. This is only valid if $4 \mid L$ holds, which is commonly satisfied for typical ViT implementations [50, 19, 20]. Each of the feature maps $\boldsymbol{Z}_j$ is transformed by a combination of deconvolutional layers that increase the resolution in both directions by a factor of two and convolutions to adjust the latent dimension. Subsequently, the transformed feature maps are successively processed in each decoder, beginning with $\boldsymbol{Z}_L$, and fused with the corresponding skip connection at each stage. This iterative fusion ensures the effective incorporation of multi-scale information, enhancing the overall performance of the decoder. Our network is designed in such a way that the output resolution of the segmentation results exactly matches the input image resolution.

As denoted in Fig. 2, our three segmentation branches (NP, HV, NT) share the same image encoder with the same skip connections and their transformations. The only difference lies in the isolated upsampling pathways of the decoders specific to each branch.

To leverage the additional tissue type information available in the PanNuke dataset, we introduce a tissue classification branch (TC) to guide the learning process of the encoder. For this, we use the class token $z_{L,\text{class}}$ as input to a linear layer with softmax activation function to predict the tissue class.

3.2 Target and Losses

For faster training and better convergence of the network, we employ a weighted combination of different loss functions for each network branch. The total loss is

$$\mathcal{L}_\text{total} = \mathcal{L}_\text{NP} + \mathcal{L}_\text{HV} + \mathcal{L}_\text{NT} + \mathcal{L}_\text{TC} \tag{4}$$

where $\mathcal{L}_\text{NP}$ denotes the loss for the NP-branch, $\mathcal{L}_\text{HV}$ the loss for the HV-branch, $\mathcal{L}_\text{NT}$ the loss for the NT-branch, and $\mathcal{L}_\text{TC}$ the loss for the TC-branch. Overall, the individual branch losses are composed of the following weighted loss functions:

$$\begin{aligned}
\mathcal{L}_\text{NP} &= \lambda_{\text{NP}_\text{FT}}\mathcal{L}_\text{FT} + \lambda_{\text{NP}_\text{DICE}}\mathcal{L}_\text{DICE}\\
\mathcal{L}_\text{HV} &= \lambda_{\text{HV}_\text{MSE}}\mathcal{L}_\text{MSE} + \lambda_{\text{HV}_\text{MSGE}}\mathcal{L}_\text{MSGE}\\
\mathcal{L}_\text{NT} &= \lambda_{\text{NT}_\text{FT}}\mathcal{L}_\text{FT} + \lambda_{\text{NT}_\text{DICE}}\mathcal{L}_\text{DICE} + \lambda_{\text{NT}_\text{BCE}}\mathcal{L}_\text{BCE}\\
\mathcal{L}_\text{TC} &= \lambda_{\text{TC}_\text{CE}}\mathcal{L}_\text{CE}
\end{aligned}$$

with the individual segmentation losses

$$\mathcal{L}_\text{BCE} = -\frac{1}{n}\sum_{i=1}^{N_\text{px}}\sum_{c=1}^{C} y_{i,c}\log\hat{y}_{ic} \tag{5}$$

$$\mathcal{L}_\text{DICE} = 1 - \frac{2\times\sum_{i=1}^{N_\text{px}} y_{ic}\hat{y}_{ic} + \varepsilon}{\sum_{i=1}^{N_\text{px}} y_{ic} + \sum_{i=1}^{N_\text{px}}\hat{y}_{ic} + \varepsilon} \tag{6}$$

$$\mathcal{L}_\text{FT} = \sum_{c=1}^{C}\left(1 - \frac{\sum_{i=1}^{N_\text{px}} y_{ic}\hat{y}_{ic} + \varepsilon}{\sum_{i=1}^{N_\text{px}} y_{ic}\hat{y}_{ic} + \alpha_\text{FT}\sum_{i=1}^{N_\text{px}} y_{ic}\hat{y}_{ic} + \beta_\text{FT}\sum_{i=1}^{N_\text{px}} y_{ic}\hat{y}_{ic}}\right)^{\frac{1}{\gamma_\text{FT}}} \tag{7}$$

and the cross-entropy as tissue classification loss

$$\mathcal{L}_\text{CE} = -\sum_{c_\text{T}=1}^{C_\text{T}} y_{c_\text{T}}\log\hat{y}_{c_\text{T}},\quad C_\text{T} = 19,$$

with the contribution of each loss to the total loss (4) controlled by the $i-$th hyperparameters λ_i. $\mathcal{L}_\text{MSE}$ denotes the mean squared error of the horizontal and vertical distance maps and $\mathcal{L}_\text{MSGE}$ the mean squared error of the gradients of the horizontal and vertical distance maps, each summarized for both directions separately. In the segmentation losses (5)-(7), y_{ic} is the ground-truth and $\hat{y}_{ic}$ the prediction probability of the ith pixel belonging to the class c, C the total number of nuclei classes, N_px the total amount of pixels, ε a smoothness factor and $\alpha_\text{FT}, \beta_\text{FT}$ and γ_FT are hyperparemters of the Focal Tversky loss.

The Cross-Entropy loss (5) and Dice loss (6) are commonly

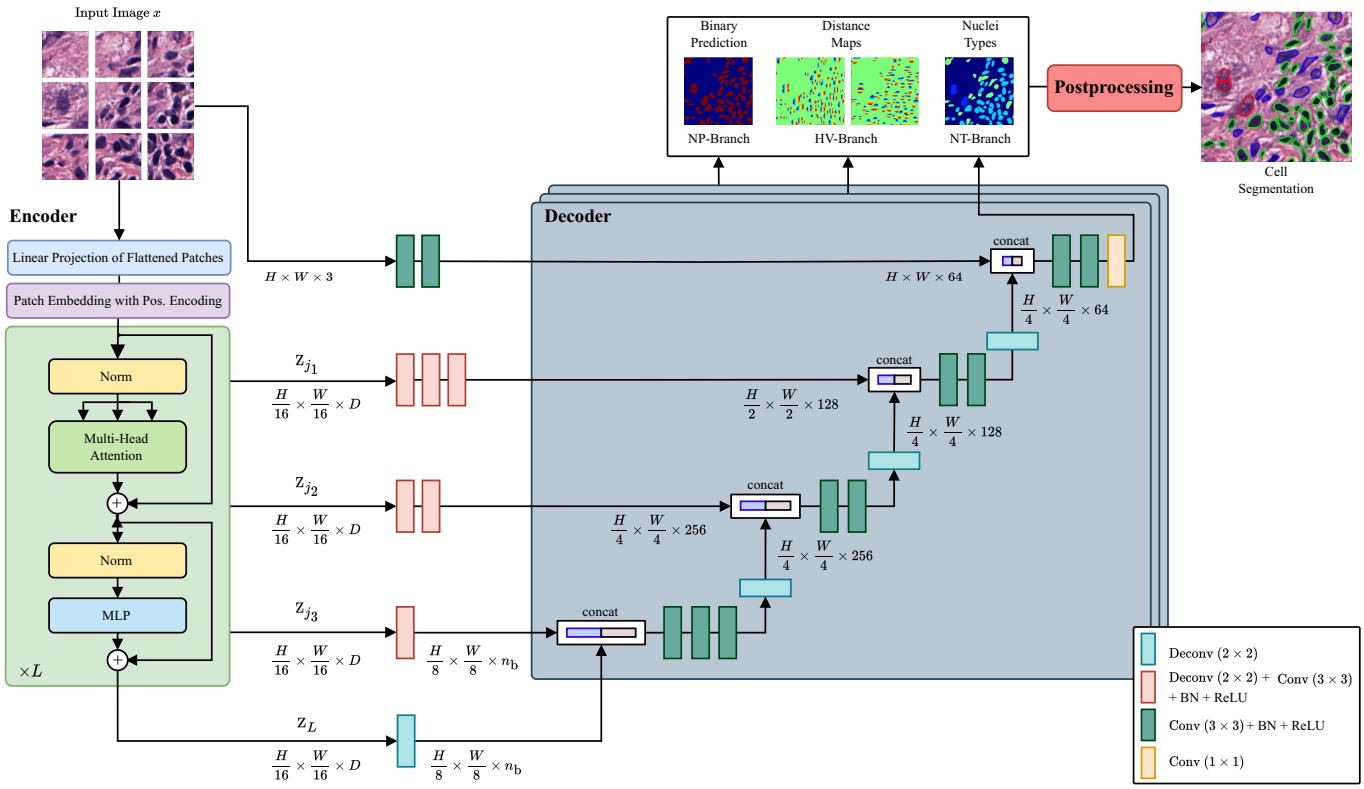

Figure 2: Network structure of our proposed CellViT-network consisting of a ViT encoder connected to multiple decoders via skip connections. Postprocessing is used to separate overlapping nuclei and perform nuclei type classification. For visualization purposes, the tissue classification branch is not illustrated. As encoder networks, we used the pre-trained ViT_{256} and SAM models.

used in semantic segmentation. To address the challenge of underrepresented instance classes, the Focal Tversky loss (7), a generalization of the Tversky loss, is used. The Focal Tversky loss places greater emphasis on accurately classifying underrepresented instances by assigning higher weights to those samples. This weighting enhances the model's capacity to handle class imbalance and focuses its learning on the more challenging regions of the segmentation task.

3.3 Postprocessing

As the network does not directly provide a semantic instance segmentation with separated nuclei, postprocessing is necessary to obtain accurate results. This involves several steps, including merging the information from the different branches, separating overlapping nuclei to ensure accurate individual segmentation, and determining the nuclei class based on the nuclei type map. Moreover, when performing inference on whole gigapixel WSI, a fusion mechanism is necessary. Due to the significant size of WSIs, inference needs to be performed on image patches extracted from them using a sliding-window approach. The segmentation results obtained from these patches must be assembled to generate a segmentation map of the entire WSI. The postprocessing methods are therefore explained in the following two paragraphs, starting with the segmentation of a single patch followed by its composition into a segmentation output for the entire WSI.

Nuclei Separation and Classification To separate adjacent and overlapping nuclei from each other, we utilize HoverNet's validated postprocessing pipeline. This involves computing the gradients of the horizontal and vertical distance maps to capture transitions between nuclei boundaries and the boundary between nuclei and the background. At these transition points significant value changes occur in the gradient. The Sobel operator (edge detection filter), is then applied to identify regions with substantial differences in neighboring pixels within the distance maps. Finally, a marker-controlled watershed algorithm is employed to generate the final boundaries.
To calculate the nuclei class, the output of the separated nuclei is merged with the nuclei type predictions. For this purpose, majority voting is performed in the nuclei region using the NT prediction map with the majority class assigned to all nuclei pixels.

Inference The encoder ViT offers a significant advantage for performing inference on gigapixel WSI over CNNs based U-Nets. Its capability to process input sequences of arbitrary length, constrained only by memory consumption and positional embedding interpolation, allows for increased input image sizes during inference. It is important to note that positional embedding interpolation must be considered when scaling the input images. In preliminary experiments on the MoNuSeg dataset (see 5.3), we found that our network achieves better performance when inferring on a single 1024×1024 px patch compared to cutting the same patch into 256×256 px sub-patches without overlap

and assembling the results. Based on these findings, we have chosen to perform WSI inference using 1024 × 1024 px large patches with a 64 px overlap. Due to the high computational overhead, it is not feasible to keep the segmentation results of the entire WSI in memory. Consequently, we process and merge only the overlapping nuclei during postprocessing. By utilizing just a small overlap in the inference patches relative to the patch size, the postprocessing effort is reduced. To efficiently store the results in a structured and readable format, as well as for compatibility with software such as QuPath [65], the nuclei predictions for an entire whole-slide image (WSI) are exported in a JSON file. Each nucleus is represented by several parameters, including the nuclei class, bounding-box coordinates, shape polygon of the boundaries, and the center of mass for detection location. In the Appendix, we provide example visualizations of the prediction results from an internal esophageal adenocarcinoma and melanoma cohort, imported into QuPath (see A.1). This approach ensures the accessibility of the instance segmentation results for further analysis and visualization. Moreover, for each detected nuclei $\hat{y}$, we store the corresponding embedding token $z_L^{\hat{y}} \in \mathbb{R}^D$. If a nucleus is associated with multiple tokens, we average over all token embeddings in which the nucleus is located. The tokens can be used as extracted cell-features for downstream DL algorithms addressing problems such as disease prediction, treatment response, and survival prediction.

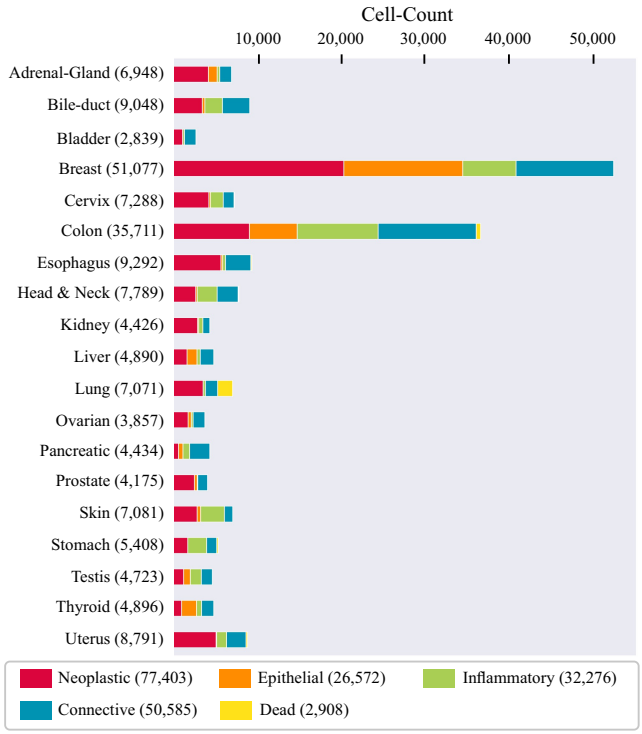

Figure 3: PanNuke nuclei distribution overview for each of the nineteen tissue types, sorted by the total number of nuclei inside the tissue. The total number of nuclei within a tissue type is given in parentheses. Adapted from [18].

4 Experimental Setup

4.1 Datasets

PanNuke We use the PanNuke dataset as the main dataset to train and evaluate our model. The dataset contains 189,744 annotated nuclei in 7,904 256 × 256 px images of 19 different tissue types and 5 distinct cell categories, as depicted in Fig. 3. Cell-images were captured at a magnification of ×40. The dataset is highly imbalanced, especially the nuclei class of dead cells is severely underrepresented, as apparent in the nuclei and tissue class statistics (see Fig. 3). PanNuke is regarded as one of the most challenging datasets to perform the simultaneous nuclei instance segmentation task [9].

MoNuSeg The MoNuSeg[66, 67] dataset serves as an additional dataset for nuclei segmentation. In contrast to PanNuke, the dataset is much smaller and does not divide the nuclei into different classes. For this work, we only use the test dataset of MoNuSeg to evaluate our model. The test dataset consists of 14 images with a resolution of 1000 × 1000 px, acquired at ×40 magnification. In total, the test dataset contains more than 7000 annotated nuclei across the seven organ types kidney, lung, colon, breast, bladder, prostate, and brain at several disease states (benign and tumors at different stages). Since no nuclei labels are included, the dataset cannot be used for evaluation classification performance. To process the dataset more effectively with our ViT-based networks with a token size of 16 px, we resized the data to a size of 1024 × 1024 px. Due to the sufficient patch-size of the original data, we also created a ×20 dataset, where the patch size is 512 × 512 px accordingly.

4.2 Experiments

In this study, we conducted a total of three experiments, with two focusing on the PanNuke dataset. Given the higher clinical relevance of the detection task over achieving the optimal segmentation quality, we (1) performed an ablation study to determine the most suitable network architecture for nuclei detection. We compared the performance of pre-trained models (see 4.4) against randomly initialized models and explored the impact of regularization techniques such as data augmentation and customized oversampling (see 4.4). Based on these investigations, we identified the best models, which were (2) subsequently evaluated for segmentation quality. To assess both detection and segmentation performance, we compared our models with the multiple baseline architectures, namely DIST [43], Mask-RCNN [38], Micro-Net [42], HoVer-Net [8], TSFD-Net [9], and CPP-Net [30]. For comparison, we conducted our experiments using the same 3-fold cross-validation (CV) splits provided by the PanNuke dataset organizers and report the averaged results over all 3 splits. It is worth mentioning that all the comparison models we evaluate in this study adhere to the same evaluation scheme for the PanNuke dataset, with one exception. The TSFD-Net publication reports results based on an 80-20 train-test split, making their results more optimistic. Nevertheless, we include their results for the purpose of comparison.
Finally, we evaluated our models trained on PanNuke on the publicly available 14 test images of the MoNuSeg dataset as a third experiment to test generalizability.

4.3 Evaluation Metrics

Nuclear Instance Segmentation Evaluation Usually, the Dice coefficient (DICE) or the Jaccard index are used as evaluation metrics for semantic segmentation. However, as Graham et al. [8] have already shown, these two metrics are insufficient for evaluating nuclear instance segmentation as they did not account for the detection quality of the nuclei. Therefore, a metric is needed that assess the following three requirements (see Graham et al. [8]):

1. Separate the nuclei from the background
2. Detect individual nuclei instances and separate overlapping nuclei
3. Segment each instance

These three tasks cannot be evaluated with the Jaccard index and the DICE score, as they just satisfy requirement (1). In line with [8] and the PanNuke dataset evaluation recommendations [18], we use the panoptic quality (PQ) [68] to quantify the instance segmentation performance. The PQ us defined as

$$PQ = \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Detection Quality (DQ)}} \times \underbrace{\frac{\sum_{(y,\hat{y})\in TP} IoU(y,\hat{y})}{|TP|}}_{\text{Segmentation Quality (SQ)}}, \quad (8)$$

with $IoU(y,\hat{y})$ denoting the intersection-over-union [68]. In this equation, y denotes a ground-truth (GT) segment, and $\hat{y}$ denotes a predicted segment, with the pair $(y,\hat{y})$ being a unique matching set of one ground-truth segment and one predicted segment. As Kirillov et al. [68] proved, each pair of segments $(y,\hat{y})$, i.e., each pair of true and predicted nuclei, in an image is unique if $IoU(y,\hat{y}) > 0.5$ is satisfied. For each class, the unique matching of $(y,\hat{y})$ splits the predicted and the GT segments into three sets:

- True Positives (TP): Matched pairs of segments, i.e., correctly detected instances
- False Positives (FP): Unmatched predicted segments, i.e., predicted instances without matching GT instance
- False negatives (FN): Unmatched GT segments, i.e., GT instances without matching predicted instance

The PQ score can be intuitively decomposed into two parts, the detection quality similar to the F_1 score commonly used in classification and detection scenarios, and the segmentation quality as the average IoU of matched segments [8, 68].
To ensure a fair comparison, we use binary PQ (bPQ) pretending that all nuclei belong to one class (nuclei vs. background) and the more challenging multiclass PQ (mPQ), taking the nuclei class into account. In doing so for mPQ, we calculate the PQ independently for each nuclei class and subsequently average the results over all classes [18].

Nuclear Classification Evaluation To evaluate the detection quality of our model, we employ commonly used detection metrics. Similar to the approach used in the PQ-score for nuclear instance segmentation evaluation, we split GT and predicted instances into TPs, FPs, and FNs. We use the conventional detection metrics precision (P_d), recall (P_d) and the ($F_{1,\mathrm{d}}$)-score as a harmonic mean between precision and recall. The index 'd' indicates that these are the scores for the entire binary nuclei detection over all classes c. Thus, the binary detection scores are defined as follows:

$$F_{1,\mathrm{d}} = \frac{2TP_\mathrm{d}}{2TP_\mathrm{d} + FP_\mathrm{d} + FN_\mathrm{d}}$$
$$P_\mathrm{d} = \frac{TP_\mathrm{d}}{TP_\mathrm{d} + FP_\mathrm{d}}$$
$$R_\mathrm{d} = \frac{TP_\mathrm{d}}{TP_\mathrm{d} + FN_\mathrm{d}}$$

We further break down TP_d into correctly classified instances of class c (TP_c), false positives of class c (FP_c) and false negatives of class c (FN_c) to derive cell-type specific scores. We then define the $F_{1,c}$-score, precision (P_c) and recall (R_c) of each nuclei class c as

$$F_{1,c} = \frac{2(TP_c + TN_c)}{2(TP_c + TN_c) + 2FP_c + 2FN_c + FP_\mathrm{d} + FN_\mathrm{d}},$$
$$P_c = \frac{TP_c + TN_c}{TP_c + TN_c + 2FP_c + FP_\mathrm{d}},$$
$$R_c = \frac{TP_c + TN_c}{TP_c + TN_c + 2FN_c + FN_\mathrm{d}}.$$

In order to prioritize the classification of different nuclear types, we incorporated an additional weighting factor for the nuclei classes, as suggested in the official PanNuke evaluation metrics [18, 8], Since we cannot use the $IoU(y,\hat{y}) > 0.5$ criterion to find matching instances $(y,\hat{y})$ between GT-instances and predictions for the detection task, we use the methodology of Sirinukunwattana et al. [69] and define a match $(y,\hat{y})$ if both centers of mass are within a radius of 6 px (×20) and 12 px (×40), respectively.

4.4 Model Training

Oversampling Even though the PanNuke dataset has around 200,000 annotated nuclei, they are distributed just across a limited number of 8,000 patches with 256 × 256 px patch size. Furthermore, there is a substantial class imbalance among tissue types and nucleic classes (see Fig. 3). Thus, we developed a new oversampling strategy based on class weightings to balance both tissue classes and nuclei classes. For each patch i in the training dataset with N_Train training samples, we calculate the sampling weights for the tissue class and the cell class with

$$p_i(\gamma_\mathrm{s}) = \frac{w_\mathrm{Tissue}(i,\gamma_\mathrm{s})}{\max\limits_{j\in[1,N_\mathrm{Train}]} w_\mathrm{Tissue}(j,\gamma_\mathrm{s})} + \frac{w_\mathrm{Cell}(i,\gamma_\mathrm{s})}{\max\limits_{j\in[1,N_\mathrm{Train}]} w_\mathrm{Cell}(j,\gamma_\mathrm{s})}, \quad (9)$$

where $w_\mathrm{Tissue}(i,\gamma_\mathrm{s})$ is a weight factor for the tissue class and $w_\mathrm{Cell}(i,\gamma_\mathrm{s})$ for the nuclei class. The parameter $\gamma_\mathrm{s} \in [0,1]$ is a weighting factor that determines the strength of the oversampling. A γ_s value of 0 indicates no oversampling, while $\gamma_\mathrm{s} = 1$ corresponds to maximum balancing. To ensure neither $w_\mathrm{Tissue}(i,\gamma_\mathrm{s})$ nor $w_\mathrm{Cell}(i,\gamma_\mathrm{s})$ dominates the sampling, normalization is applied to both summands in eq. (9). The calculation of the weighting factor of the tissue class can be calculated directly via

$$w_\mathrm{Tissue}(i,\gamma_\mathrm{s}) = \frac{N_\mathrm{Train}}{\gamma_\mathrm{s}\left(\sum\limits_{j\in[1,N_\mathrm{Train}]|c_{\mathrm{T},j}=c_{\mathrm{T},i}} 1\right) + (1-\gamma_\mathrm{s})N_\mathrm{Train}} \quad (10)$$

as each patch can only belong to one tissue class denoted by $c_{\mathrm{T},i}$. For cell weighting, it must be considered that each patch can

Table 1: Precision (P), Recall (R) and F_1-score (F_1) for detection and classification across three dataset splits for each nuclei type. The centroid of each nucleus was used for computing detection metrics for segmentation networks. *TSFD-Net was not evaluated on the official 3-fold splits of the PanNuke dataset and left out by the comparison **Models trained on downscaled ×20 PanNuke images

	Detection			Classification														
				Neoplastic			Epithelial			Inflammatory			Connective			Dead		
	P_{d}	R_{d}	$F_{1,d}$	P_{Neo}	R_{Neo}	$F_{1,Neo}$	P_{Epi}	R_{Epi}	$F_{1,Epi}$	P_{Inf}	R_{Inf}	$F_{1,Inf}$	P_{Con}	R_{Con}	$F_{1,Con}$	P_{Dead}	R_{Dead}	$F_{1,Dead}$
DIST	0.74	0.71	0.73	0.49	0.55	0.50	0.38	0.33	0.35	0.42	0.45	0.42	0.42	0.37	0.39	0.00	0.00	0.00
Mask-RCNN	0.76	0.68	0.72	0.55	0.63	0.59	0.52	0.52	0.52	0.46	0.54	0.50	0.42	0.43	0.42	0.17	0.30	0.22
Micro-Net	0.78	0.82	0.80	0.59	0.66	0.62	0.63	0.54	0.58	0.59	0.46	0.52	0.50	0.45	0.47	0.23	0.17	0.19
HoVer-Net	0.82	0.79	0.80	0.58	0.67	0.62	0.54	0.60	0.56	0.56	0.51	0.54	0.52	0.47	0.49	0.28	0.35	0.31
TSFD-Net*	0.84	0.87	0.85	0.60	0.71	0.65	0.56	0.58	0.57	0.59	0.58	0.57	0.55	0.49	0.53	0.33	0.40	0.43
CellViT$_{256}$ – Raw	0.81	0.78	0.79	0.63	0.65	0.64	0.68	0.63	0.65	0.55	0.49	0.52	0.48	0.45	0.46	0.43	0.21	0.28
CellViT$_{256}$ – Over	0.81	0.78	0.80	0.65	0.63	0.64	0.66	0.63	0.64	0.57	0.50	0.53	0.47	0.47	0.47	0.45	0.23	0.30
CellViT$_{256}$ – Aug	0.83	0.82	**0.83**	0.71	0.71	0.71	0.73	0.74	0.74	0.60	**0.60**	**0.60**	0.56	0.53	0.54	0.38	**0.43**	0.40
CellViT$_{256}$	0.82	**0.83**	**0.83**	0.71	0.71	0.71	0.73	0.74	0.74	0.61	0.59	**0.60**	0.55	0.53	0.54	0.42	0.38	0.40
CellViT-SAM-B	**0.84**	0.82	**0.83**	**0.73**	0.71	**0.72**	**0.75**	0.74	**0.75**	**0.63**	0.58	**0.60**	0.56	**0.55**	**0.55**	0.45	0.36	0.40
CellViT-SAM-L	**0.84**	0.82	**0.83**	**0.73**	0.71	**0.72**	**0.75**	0.74	**0.75**	0.61	0.59	**0.60**	0.56	0.54	**0.55**	**0.48**	0.37	**0.42**
CellViT-SAM-H	**0.84**	0.81	**0.83**	0.72	**0.72**	**0.72**	**0.75**	**0.75**	**0.75**	0.62	0.59	**0.60**	**0.57**	0.53	**0.55**	**0.48**	0.38	**0.42**
CellViT-Random	0.75	0.75	0.75	0.60	0.62	0.61	0.64	0.64	0.64	0.59	0.45	0.57	0.47	0.47	0.47	0.34	0.38	0.36
CellViT$_{256}$ (×20)**	0.88	0.60	0.71	0.74	0.59	0.66	0.77	0.61	0.68	0.63	0.39	0.48	0.55	0.33	0.42	0.37	0.03	0.05
CellViT-SAM-H (×20)**	0.89	0.62	0.73	0.76	0.62	0.68	0.79	0.63	0.70	0.64	0.42	0.50	0.58	0.36	0.44	0.52	0.04	0.08

contain multiple nuclei from different cell classes. Therefore, we create a binary vector $\boldsymbol{c_i} \in \{0,1\}^C$, where each entry is set to 1 for each existing nuclei type c in the patch. To get a reference value for scaling similar to eq. (10), we calculate $N_{\text{Cell}} = \sum_{i=1}^{N_{\text{Train}}} \|\boldsymbol{c}_i\|_1$. The cell weighting for each training image i is then calculated by

$$w_{\text{Cell}}(i, \gamma_s) = (1-\gamma_s) + \gamma_s \sum_{j=1}^{C} c_{ij} \frac{N_{\text{Cell}}}{\gamma_s \sum_{k=1}^{N_{\text{Train}}} c_{kj} + (1-\gamma_s) N_{\text{Cell}}},$$

with c_{ij} the vector entry of $\boldsymbol{c}_i$ at position j. The training images are randomly sampled in a training epoch with replacement based on their sampling weights $p_i(\gamma_s)$.

Data Augmentation In addition to our customized oversampling strategy, we extensively employ data augmentation techniques to enhance data variety and discourage overfitting. We use a combination of the following geometrical and noisy/intensity-based augmentation methods: random 90-degree rotation, horizontal flipping, vertical flipping, downscaling, blurring, gaussian noise, color jittering, superpixel representation of image sections (SLIC), zoom blur, random cropping with resizing and elastic transformations. These augmentation techniques were selected to introduce variations in the shape, orientation, texture, and appearance of the nuclei, enhancing the robustness and generalization capabilities of the model. For detailed information on the augmentation methods utilized, including the selected probabilities and corresponding hyperparameters, please refer to the Appendix.

Optimization and Training Strategy We train all our models for 130 epochs and incorporate exponential learning rate scheduling with a scheduling factor of 0.85 to gradually reduce the learning rate during training. To balance our training, we use our modified oversampling strategy with $\gamma_s = 0.85$. A complete overview of all hyperparameters, including optimizer, data augmentation, and weighting factors of the loss function (4), is provided in the Appendix.
As for the encoder models, we leverage the ViT$_{256}$-model (ViT-S, $D = 384$, $L = 12$), which has been pre-trained on histological data (see 2.2). Additionally, we compare the performance with the three pre-trained SAM checkpoints: SAM-B (ViT-B, $D = 768$, $L = 12$), SAM-L (ViT-L, $D = 1024$, $L = 24$) and SAM-H (ViT-H, $D = 1280$, $L = 32$). These checkpoints provide different model sizes and complexities, allowing us to evaluate their respective performance and choose the most suitable one for our task. During training, we initially freeze the encoder weights for the first 25 epochs. After this initial warm-up phase to train the decoder, we proceed to train the entire model including the image encoder.

Implementation All models are implemented in PyTorch 1.13.1. To augment images and masks, we used the Albumentation library [70]. For the pre-trained ViT$_{256}$-model, we utilized the ViT-S checkpoint[1] provided by Chen et al. [19]. As for the SAM-B, SAM-L, and SAM-H models, we use the encoder backbones of each final training stage of SAM [20], published on GitHub[2]. All experiments were conducted on an 80 GB NVIDIA A100 GPU. However, it is worth noting that a 48 GB NVIDIA RTX A6000 is also sufficient for the ViT$_{256}$ and SAM-B model training.

5 Results

In the section below, the results for the three experiments (1) nuclei detection quality on PanNuke, (2) segmentation quality on PanNuke and (3) generalization performance on the independent MoNuSeg cohort are described.

5.1 *Detection Quality on PanNuke*

Considering the clinical importance of nuclei detection and classification over achieving the best possible segmentation quality, our ablation study aimed to determine the best model based on the detection results using the PanNuke dataset. Tab. 1 presents the precision, recall, and F_1-Score for both

[1] https://github.com/mahmoodlab/HIPT
[2] https://github.com/facebookresearch/segment-anything

Table 2: Average mPQ and bPQ across the 19 tissue types of the PanNuke dataset for 3-fold cross-validation. The standard deviation (STD) of the splits is provided in the final row. *TSFD-Net was not evaluated on the official 3-fold splits of the PanNuke dataset and left out by the comparison **Head & Neck

Tissue	DIST		Mask-RCNN		Micro-Net		HoVer-Net		TSFD-Net*		CPP-Net		CellViT_{256}		CellViT-SAM-H	
	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ
Adrenal	0.3442	0.5603	0.3470	0.5546	0.4153	0.6440	0.4812	0.6962	0.5223	0.6900	0.4944	**0.7066**	0.5186	0.7014	**0.5248**	0.7008
Bile Duct	0.3614	0.5384	0.3536	0.5567	0.4124	0.6232	0.4714	0.6696	0.5000	0.6284	0.4670	**0.6768**	0.5006	0.6345	**0.5092**	0.6421
Bladder	0.4463	0.5625	0.5065	0.6049	0.5357	0.6488	0.5792	0.7031	0.5738	0.6773	**0.5936**	**0.7053**	0.5851	0.6529	0.5721	0.6497
Breast	0.3790	0.5466	0.3882	0.5574	0.4407	0.6029	0.4902	0.6470	0.5106	0.6245	0.5090	**0.6747**	0.5165	0.6593	**0.5228**	0.6690
Cervix	0.3371	0.5309	0.3402	0.5483	0.3795	0.6101	0.4438	0.6652	0.5204	0.6561	0.4792	**0.6912**	0.5171	0.6024	**0.5234**	0.6138
Colon	0.2989	0.4508	0.3122	0.4603	0.3414	0.4972	0.4095	0.5575	0.4382	0.5370	0.4315	**0.5911**	0.4510	0.5263	**0.4577**	0.5392
Esophagus	0.3942	0.5295	0.4311	0.5691	0.4668	0.6011	0.5085	0.6427	0.5438	0.6306	0.5449	**0.6797**	0.5469	0.6573	**0.5474**	0.6636
H&N**	0.3177	0.4764	0.3946	0.5457	0.3668	0.5242	0.4530	0.6331	0.4937	0.6277	0.4706	**0.6523**	0.4924	0.5454	**0.4996**	0.5502
Kidney	0.3339	0.5727	0.3553	0.5092	0.4165	0.6321	0.4424	0.6836	0.5517	0.6824	0.5194	**0.7067**	0.5582	0.6643	**0.5621**	0.6806
Liver	0.3441	0.5818	0.4103	0.6085	0.4365	0.6666	0.4974	0.7248	0.5079	0.6675	0.5143	**0.7312**	0.5201	0.7179	**0.5316**	0.7268
Lung	0.2809	0.4978	0.3182	0.5134	0.3370	0.5588	0.4004	0.6302	0.4274	0.5941	0.4256	**0.6386**	0.4224	0.6268	**0.4263**	0.6331
Ovarian	0.3789	0.5289	0.4337	0.5784	0.4387	0.6013	0.4863	0.6309	0.5253	0.6431	0.5313	**0.6830**	0.5259	0.6539	**0.5379**	0.6661
Pancreatic	0.3395	0.5343	0.3624	0.5460	0.4041	0.6074	0.4600	0.6491	0.4893	0.6241	0.4706	**0.6789**	**0.5065**	0.6699	0.4997	0.6701
Prostate	0.3810	0.5442	0.3959	0.5789	0.4341	0.6049	0.5101	0.6615	0.5431	0.6406	0.5305	**0.6927**	0.5267	0.6583	**0.5489**	0.6727
Skin	0.2627	0.5080	0.2665	0.5021	0.3223	0.5817	0.3429	0.6234	0.4354	0.6074	0.3574	0.6209	0.4155	0.6202	**0.4417**	**0.6289**
Stomach	0.3369	0.5553	0.3684	0.5976	0.3872	0.6293	0.4726	0.6886	0.4871	0.6529	0.4582	0.7067	0.4544	0.7033	**0.4691**	**0.7162**
Testis	0.3278	0.5548	0.3512	0.5420	0.4088	0.6300	0.4754	0.6890	0.4843	0.6435	0.4931	**0.7026**	**0.5157**	0.6692	0.4978	0.6719
Thyroid	0.2574	0.5596	0.3037	0.5712	0.3712	0.6555	0.4315	0.6983	0.5154	0.6692	0.4392	**0.7155**	**0.4792**	0.6914	0.4771	0.6887
Uterus	0.3487	0.5246	0.3683	0.5589	0.3965	0.5821	0.4393	0.6393	0.5068	0.6204	**0.4794**	**0.6615**	0.4468	0.6270	0.4691	0.6384
Average	0.3406	0.5346	0.3688	0.5528	0.4059	0.6053	0.4629	0.6596	0.5040	0.6377	0.4847	**0.6789**	0.5000	0.6464	**0.5062**	0.6538
STD	0.0156	0.0098	0.0047	0.0076	0.0082	0.0050	0.0076	0.0036	-	-	0.0059	0.0015	0.0156	0.0092	0.0145	0.0078

Table 3: Average PQ across three dataset splits for each nuclear category on the PanNuke dataset. *TSFD-Net was not evaluated on the official 3-fold splits of the PanNuke dataset and left out by the comparison **Models trained on downscaled ×20 PanNuke images

	Neoplastic	Inflammatory	Connective	Dead Cell	Epithelial
DIST	0.439	0.343	0.275	0.000	0.290
Mask-RCNN	0.472	0.290	0.300	0.069	0.403
Micro-Net	0.504	0.333	0.334	0.051	0.442
HoVer-Net	0.551	0.417	0.388	0.139	0.491
TSFD-Net*	0.572	0.453	0.423	0.214	0.566
CellViT_{256}	0.578	0.431	0.423	0.172	0.582
CellViT-SAM-H	**0.591**	**0.432**	**0.428**	**0.181**	**0.587**
CellViT_{256} (×20)**	0.505	0.300	0.298	0.020	0.483
CellViT-SAM-H (×20)**	0.539	0.324	0.325	0.029	0.510

detection and classification performance across all nuclei classes, including the binary case. To determine the optimal settings, we evaluated different variations of our network. These include a randomly initialized network (CellViT-Random), networks with pre-trained weights from the ViT_{256} network (CellViT_{256}), and networks with different pre-trained SAM backbones (CellViT-SAM-B, CellViT-SAM-L, CellViT-SAM-H). To ensure comparability, the CellViT-Random network shares the same architecture (ViT-S, $D = 384$, $L = 12$) as the CellViT_{256} network. All mentioned model variants were trained using data augmentation and our customized sampling strategy as regularization methods. Compared to the baseline models, the randomly initialized CellViT-Random network achieves detection results comparable to the HoverNet network. However, when using pre-trained encoder networks, we observe a significant performance increase, reaching state-of-the-art performance. We notice a strong increase in F_1-scores compared to all existing solutions, especially for the epithelial nuclei class. Both the ViT_{256} and the three different SAM encoders exhibit significantly better performance, all at a similar level, with the CellViT-SAM-H model as the best solution. Notably, we even outperform purely detection-based methods like Mask-RCNN and all state-of-the-art approaches by a large margin with up to a 31 % increase in the $F_{1,\text{Epi}}$-score of epithelial nuclei.

To demonstrate the effect of extensive data augmentation and customized sampling strategy, we additionally report the results for a CellViT_{256} model without regularization (CellViT_{256}-Raw), with oversampling only (CellViT_{256}-Over) and with data-augmentation only (CellViT_{256}-Aug) in Tab. 1. Our experiments reveal that data augmentation, in particular, is a crucial regularization method that significantly enhances performance. Specifically, the addition of data augmentation results in a 0.12 increase in the $F_{1,\text{Dead}}$ score for the dead nuclei class. However, oversampling leads to minimal improvements in the results. For subsequent investigations, we decided to further consider the CellViT_{256} and CellViT-SAM-H models to enable a comparison between in-domain and out-of-domain pre-training.

In addition, we performed training and evaluation on downscaled PanNuke data (from 256×256 to 128×128 patch size), resulting in a magnification of ×20, for the two best model variants CellViT_{256} and CellViT-SAM-H. The results are presented in the last two rows of Tab. 1. This downsizing lead to a substantial drop in performance compared to the ×40 networks, with detection results becoming comparable to the baseline models. Notably, the recall of individual classes significantly decreases (by an average of −0.18). In particular, the recall for the dead nuclei class drops to 0.03, indicating that this class is almost not detected at all. Interestingly, the precision increases minimally or remains almost the same compared to our best ×40 models. We conclude that although significantly fewer nuclei are detected, if a nuclei has been detected and classified appropriately, this also corresponds to the true nuclei class with high accuracy for most of the classes.

5.2 *Segmentation Quality on PanNuke*

We evaluate the segmentation performance of the CellViT_{256} and CellViT-SAM-H models against baseline models by computing the binary panoptic quality (*bPQ*) and the more

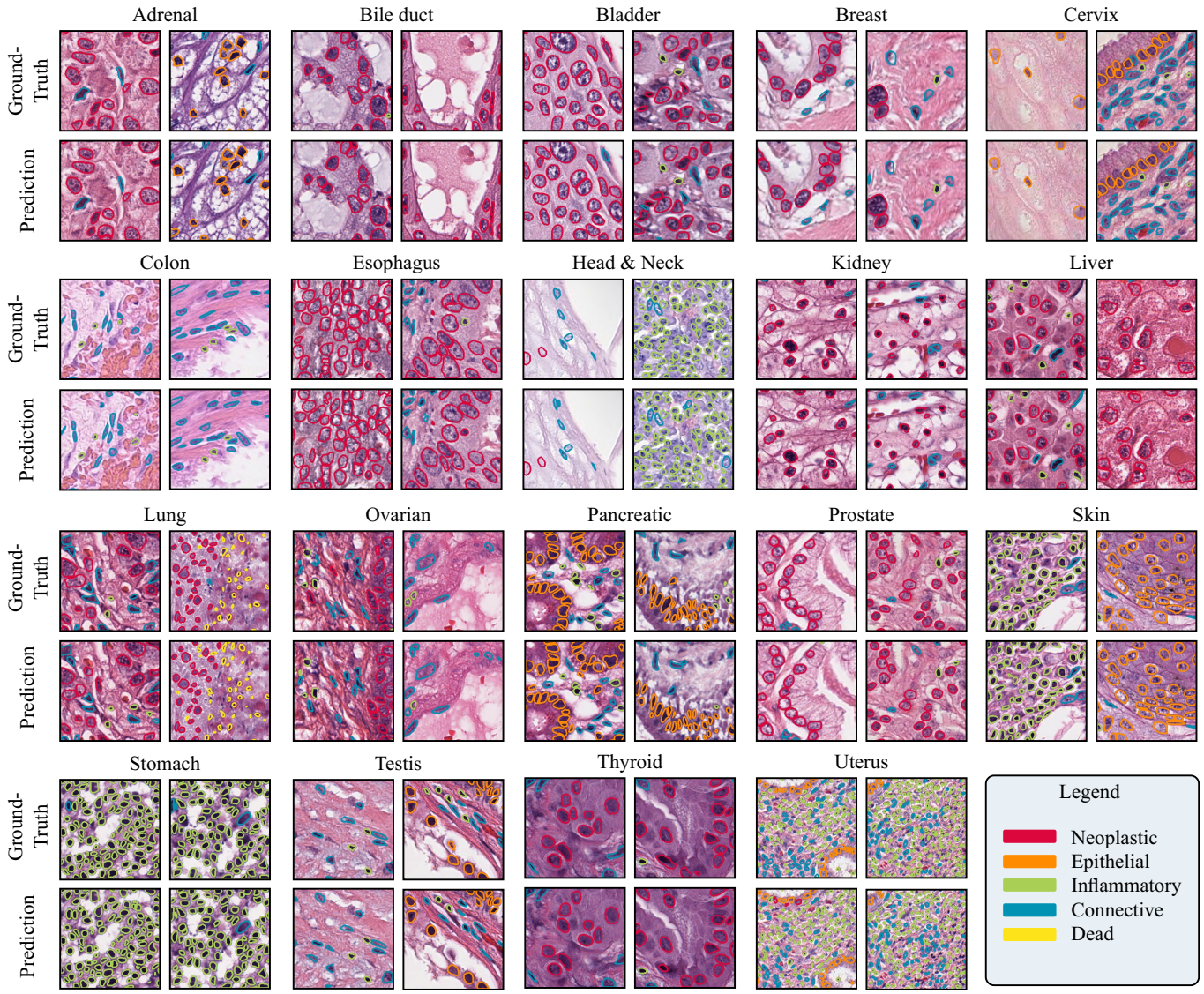

Figure 4: Example of PanNuke patches with ground-truth annotations and CellVit-SAM-H predictions overlaid for each tissue type.

challenging multi-class PQ (mPQ) for each of the 19 tissue types, providing an assessment of both instance segmentation qualities. Our experimental results, given in Tab. 2, demonstrate that CPP-Net and TSFD-Net exhibit better bPQ, whereas our models achieve superior mPQ. The findings from this experiment can be explained as follows: Although our network exhibits slightly lower segmentation quality (SQ) compared to CPP-Net and TSFD-Net indicated by the lower bPQ, it excels in detection quality (DQ), resulting in an overall better mPQ for our models. Therefore, our network outperforms existing solutions in tackling the more challenging multi-class problem, yielding superior results.

Furthermore, Tab. 3 presents the PQ values for each nuclei type, averaged over all tissue types. Both CellViT$_{256}$ and CellViT-SAM-H outperform all other models in neoplastic, connective, and epithelial tissue. For inflammatory and connective tissue, they are outperformed by TSFD-Net, which benefits from a larger training dataset (80/20 split vs. 67/33 split). Notably, the results for dead cells are consistently the lowest among all models, which can be attributed to the substantial class imbalance and the small physical size of dead cells. Additionally, we include results for models trained on ×20 mPanNuke data, as explained in section 5.1. The results for the ×20 models reveal a significant drop in performance when using the downscaled data compared to the ×40 models.

To provide a visual representation of the segmentations, we include tissue-wise comparisons between ground-truth and segmentation predictions of the CellViT-SAM-H model in Fig. 4. As observed in the lung example, the instance segmentation of dead cells poses a significant challenge due to their small size. Furthermore, detecting and segmenting dead nuclei becomes even more difficult when these images are scaled down from ×40 to ×20 magnification. Segmentation results per tissue for ×20 are given in the Appendix A.1.

Table 4: MoNuSeg validation result for CellViT$_{256}$ and CellViT-SAM-H models on different dataset magnifications and inference patch sizes averaged over all three PanNuke training folds. The original image size for ×40 is 1024 px, and 512 px for ×20 (no patching). *Models trained on downscaled ×20 PanNuke images

Magnification	×40								×20							
Inference patch size	1024 px (no patching)				256 px (patched)				512 px (no patching)				256 px (patched)			
Metric	*bPQ*	P_d	R_d	$F_{1,d}$	*bPQ*	P_d	R_d	$F_{1,d}$	*bPQ*	P_d	R_d	$F_{1,d}$	*bPQ*	P_d	R_d	$F_{1,d}$
CellViT$_{256}$	0.665	0.838	0.861	0.848	0.625	0.813	0.897	0.852	0.601	0.916	0.766	0.833	0.595	0.911	0.764	0.830
CellViT-SAM-H	**0.673**	**0.843**	0.880	**0.860**	**0.631**	**0.815**	0.899	**0.854**	0.625	**0.919**	0.795	**0.852**	0.618	**0.916**	0.784	**0.844**
CellViT$_{256}$ (×20)*	0.546	0.769	0.911	0.822	0.522	0.743	0.912	0.811	**0.652**	0.870	0.809	0.838	**0.646**	0.869	0.805	0.835
CellViT-SAM-H (×20)*	0.539	0.760	**0.961**	0.848	0.515	0.735	**0.967**	0.834	0.637	0.841	**0.821**	0.830	0.630	0.835	**0.812**	0.822

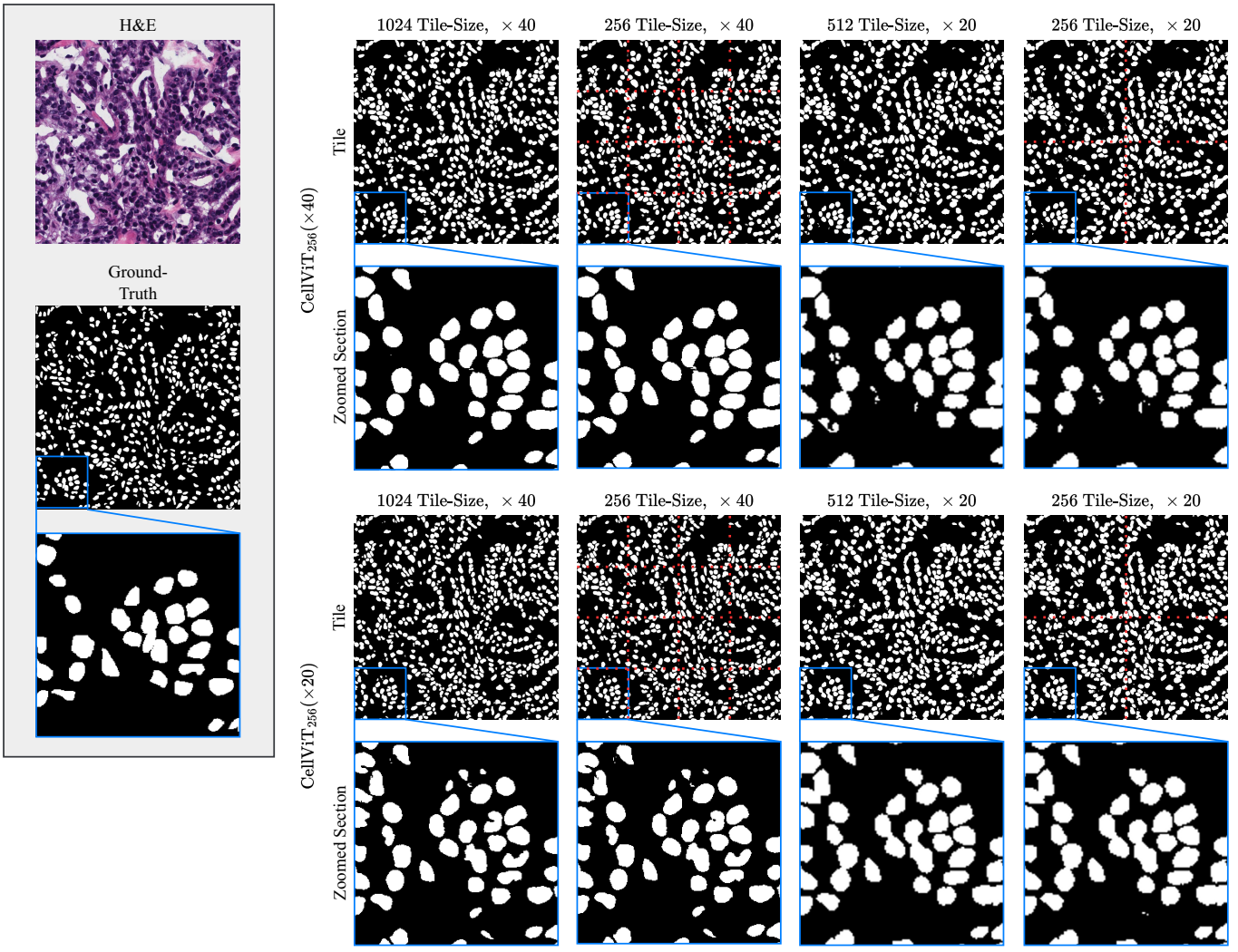

Figure 5: Example of one MoNuSeg tissue sample with ground-truth binary masks and predictions of the CellViT$_{256}$ (×40) and CellViT$_{256}$ (×20) model for different input sizes and magnifications.

5.3 *MoNuSeg Test Performance*

In this experiment, we focused on instance segmentation without classification on the MoNuSeg dataset to assess the generalizability of our models at magnifications of ×40 and ×20. Additionally, we aim to evaluate the impact of changing the input sequence size by performing inference on large-scale tiles of size 1024 px (×40) and 512 px (×20), respectively, and non-overlapping 256 px patches, derived by a shifting window approach. We utilized the three final models of the PanNuke training folds for each architecture and conducted inference on the MoNuSeg data. The evaluation results are presented in Tab. 4. Consistent with the previous experiments, the CellViT-SAM-H model is the best-performing model. It achieves a *bPQ*-score of 0.673 on 1024 px tiles when no patching was applied. However, when using 256 px patches, the *bPQ*-score slightly decreases to 0.631, likely due to the absence of merging overlapping nuclei at cell borders. The models utilizing patched tiles as input exhibit higher recall since the prediction maps were merged without

overlap, allowing cells to be detected multiple times. Importantly, the overall results indicate that inference on larger tiles did not lead to a degradation in performance, confirming our assumption that our architecture can be applied to larger patches. This justifies our inference pipeline for large-scale whole-slide images (WSI), in which we are using 1024 px sized patches with an overlap of 64 px and overlapping merging strategies, as explained in section 3.3. The CellViT$_{256}$ model yields slightly inferior results compared to the CellViT-SAM-H model.
Regarding the ×20 models on the ×40 data and vice versa, the ×20 models exhibit poor performance, while the ×40 models experience a less severe performance drop on the ×20 data. Nevertheless, the ×20 models on the ×20 data are still inferior to the ×40 models on the ×40 data. Among the ×20 models, the CellViT$_{256}$ (×20) model emerged as the best-performing model. Overall, our findings highlight the robustness and adaptability of our models in instance segmentation tasks. The CellViT-SAM-H model consistently demonstrates superior performance, even at different magnifications and input sizes, reinforcing its effectiveness in handling various image resolutions and patch sizes. Nevertheless, the best performance is consistently achieved for WSI acquired at ×40 magnification.

We also show an example tissue tile from the MoNuSeg test set along with binary segmentation masks of the CellViT$_{256}$ and CellViT$_{256}$(×40) models in Fig. 5, allowing for a visual comparison of the performance of the models. This visual assessment provides additional insights into the segmentation accuracy and helps to further evaluate the effectiveness of the models in accurately identifying and delineating nuclei within the tissue samples. The visualization confirms that the ×40 model performs best on the ×40 data, giving satisfactory results on the ×20 data, whereas the ×20 model is only applicable to the ×20 data.

6 Discussion and Conclusion

Nuclei instance segmentation are crucial for clinical applications, requiring automated tools that offer high robustness and reliability. In the clinical context of performing large-scale analysis on clinical patient cohorts, accurate detection is considered more important than precise segmentation.
In this work, we introduced a novel deep learning-based method for simultaneously segmenting and detecting nuclei in digitized H&E tissue samples. Our work was inspired by the success of previous works using large-scale trained Vision Transformers, particularly by the contributions made by Chen et al. [19] (ViT$_{256}$) and Kirillov et al. [20] (SAM). The CellViT network proposed in this study demonstrates state-of-the-art performance for both nuclei instance segmentation and nuclei detection on the PanNuke dataset. Additionally, the results on the MoNuSeg dataset validate the generalizability of our model to previously unseen cohorts. Notably, our model surpasses all other existing methods by a significant margin for nuclei detection and classification, elevating nuclei detection in H&E-slides to a new level. By leveraging the most recent computer vision approaches, we showed that both in-domain pre-training (ViT$_{256}$) and the use of the SAM foundation model yields significantly better results compared to randomly initialized network weights. Moreover, our framework allows direct assessment of a localizable ViT-token from a detected nucleus that can be further used in downstream tissue analysis tasks. Although an evaluation of this aspect is pending, we anticipate promising prospects from the utilization of these localizable embeddings. Our work provides the potential to design interpretable algorithms that directly correlate with specific cells or cell patterns. One possible direction for future research involves graph-based networks with attention mechanisms using these embeddings.
Nevertheless, external validation of the results is necessary. Yet, additional datasets are required, especially to verify the detection quality of our model. Furthermore, our models exhibit reliable performance only for WSI acquired at ×40 magnification. While the results obtained with ×20 magnified images are acceptable in terms of detection, there is room for improvement, as there is a huge performance gap between ×40 and ×20-WSI processing. We recommend to scan the tissue samples on a resolution of ×40 if technically possible. In the future, we plan to apply the proposed model with extracted nuclei tokens to downstream histological image analysis tasks. This will enable us to validate if simultaneously extracted tokens are an advantage for building interpretable algorithms for computational pathology. Additionally, it will allow us to evaluate which tokens have achieved a more meaningful representation of the tissue and are better suited for downstream tasks, as there are just minimal differences in the segmentation and detection performance between our best-performing CellViT$_{256}$ and CellViT-SAM-H models. To ensure the accessibility of our results, we have made both the code and pre-trained models publicly available under an open-source license for non-commercial use.

Acknowledgments

This work received funding from 'KITE' (Plattform für KI-Translation Essen) from the REACT-EU initiative (https://kite.ikim.nrw/, EFRE-0801977) and the Cancer Research Center Cologne Essen (CCCE).

References

[1] K. B. Tran, J. J. Lang, K. Compton, R. Xu, A. R. Acheson, and H. J. Henrikson, et al. The global burden of cancer attributable to risk factors, 2010–19: a systematic analysis for the global burden of disease study 2019. *The Lancet*, 400(10352):563–591, August 2022. doi: 10.1016/s0140-6736(22)01438-6.

[2] S. E. Stanton and M. L. Disis. Clinical significance of tumor-infiltrating lymphocytes in breast cancer. *Journal for ImmunoTherapy of Cancer*, 4(1), October 2016. doi: 10.1186/s40425-016-0165-6.

[3] F. R. Greten and S. I. Grivennikov. Inflammation and cancer: Triggers, mechanisms, and consequences. *Immunity*, 51(1):27–41, July 2019. doi: 10.1016/j.immuni.2019.06.025.

[4] B. T. Grünwald, A. Devisme, G. Andrieux, F. Vyas, K. Aliar, and C. W. McCloskey, et al. Spatially confined sub-tumor microenvironments in pancreatic cancer. *Cell*, 184(22):5577–5592.e18, October 2021. doi: 10.1016/j.cell.2021.09.022.

[5] O. Ester, F. Hörst, C. Seibold, J. Keyl, S. Ting, and N. Vasileiadis, et al. Valuing vicinity: Memory attention framework for context-based semantic segmentation in histopathology. *Computerized Medical Imaging and Graphics*, 107:102238, July 2023. ISSN 08956111. doi: 10.1016/j.compmedimag.2023.102238.

[6] M. Y. Lu, D. F. K. Williamson, T Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5(6):555–570, March 2021. doi: 10.1038/s41551-020-00682-w.

[7] F. Hörst, S. Ting, S. T. Liffers, K. L. Pomykala, K. Steiger, and M. Albertsmeier, et al. Histology-based prediction of therapy response to neoadjuvant chemotherapy for esophageal and esophagogastric junction adenocarcinomas using deep learning. *JCO Clinical Cancer Informatics*, 2023. Forthcoming.

[8] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, and J. T. Kwak, et al. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019. ISSN 1361-8415. doi: 10.1016/j.media.2019.101563.

[9] T. Ilyas, Z. I. Mannan, A. Khan, S. Azam, H. Kim, and F. De Boer. TSFD-Net: Tissue specific feature distillation network for nuclei segmentation and classification. *Neural Networks*, 151:1–15, July 2022. ISSN 0893-6080. doi: 10.1016/j.neunet.2022.02.020.

[10] S. Graham, Q. D. Vu, M. Jahanifar, S. E. A. Raza, F. Minhas, and D. Snead, et al. One model is all you need: Multi-task learning enables simultaneous histology image segmentation and classification. *Medical Image Analysis*, 83:102685, 2023. ISSN 1361-8415. doi: 10.1016/j.media.2022.102685.

[11] K. Sirinukunwattana, D. Snead, D. Epstein, Z. Aftab, I. Mujeeb, and Y. W. Tsang, et al. Novel digital signatures of tissue phenotypes for predicting distant metastasis in colorectal cancer. *Scientific Reports*, 8(1), September 2018. doi: 10.1038/s41598-018-31799-3.

[12] G. Corredor, J. Whitney, V. Arias, A. Madabhushi, and E. Romero. Training a cell-level classifier for detecting basal-cell carcinoma by combining human visual attention maps with low-level handcrafted features. *Journal of Medical Imaging*, 4(2):021105, March 2017. doi: 10.1117/1.jmi.4.2.021105.

[13] G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. Werneck, and K. Silva, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature Medicine*, 25(8):1301–1309, July 2019. doi: 10.1038/s41591-019-0508-1.

[14] S. Graham, M. Jahanifar, Q. D. Vu, G. Hadjigeorghiou, T. Leech, and D. Snead, et al. CoNIC: Colon nuclei identification and counting challenge 2022. *arXiv Preprint*, November 2021. doi: 10.48550/arXiv.2111.14485.

[15] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, In. Kang, and O. Friman, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, 2006. doi: 10.1186/gb-2006-7-10-r100.

[16] S. Kothari, Q. Chaudry, and M.D. Wang. Extraction of informative cell features by segmentation of densely clustered tissue images. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, September 2009. doi: 10.1109/iembs.2009.5333810.

[17] J. M. Murray, G. Kaissis, R. Braren, and J. Kleesiek. Wie funktioniert radiomics? *Der Radiologe*, 60(1):32–41, December 2019. doi: 10.1007/s00117-019-00617-w.

[18] J. Gamper, N. A. Koohbanani, K. Benes, S. Graham, M. Jahanifar, and S. A. Khurram, et al. PanNuke dataset extension, insights and baselines. *arXiv Preprint*, April 2020. doi: 10.48550/arXiv.2003.10778.

[19] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, and R. G Krishnan, et al. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16144–16155, June 2022. doi: 10.1109/CVPR52688.2022.01567.

[20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, and L. Gustafson, et al. Segment anything. *arXiv Preprint*, April 2023. doi: 10.48550/arXiv.2304.02643.

[21] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, and B. Landman, et al. UNETR: Transformers for 3D medical image segmentation. *arXiv Preprint*, October 2021. doi: 10.48550/arXiv.2103.10504.

[22] X. Yang, H. Li, and X. Zhou. Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(11):2405–2414, 2006. doi: 10.1109/TCSI.2006.884469.

[23] N. Malpica, C. O. de Solórzano, J.J. Vaquero, A. Santos, I. Vallcorba, and J. M. García-Sagredo, et al. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28(4):289–297, December 1998. doi: 10.1002/(sici)1097-0320(19970801)28:4<289::aid-cyto3>3.0.co;2-7.

[24] A. Tareef, Y. Song, H. Huang, D. Feng, M. Chen, and Y. Wang, et al. Multi-pass fast watershed for accurate segmentation of overlapping cervical cells. *IEEE Transactions on Medical Imaging*, 37(9):2044–2059, 2018. doi: 10.1109/TMI.2018.2815013.

[25] J. Cheng and J. C. Rajapakse. Segmentation of clustered nuclei with shape markers and marking function. *IEEE Transactions on Biomedical Engineering*, 56(3):741–748, 2009. doi: 10.1109/TBME.2008.2008635.

[26] M. Veta, P. J. van Diest, R. Kornegoor, A. Huisman, M. A. Viergever, and J. P. W. Pluim. Automatic nuclei segmentation in h&e stained breast cancer histopathology images. *PLOS ONE*, 8(7):null, 07 2013. doi: 10.1371/journal.pone.0070221.

[27] S. Ali and A. Madabhushi. An integrated region-, boundary-, shape-based active contour for multiple object overlap resolution in histological imagery. *IEEE Transactions on Medical Imaging*, 31(7):1448–1460, 2012. doi: 10.1109/TMI.2012.2190089.

[28] S. Wienert, D. Heim, K. Saeger, A. Stenzinger, M. Beil, and P. Hufnagl, et al. Detection and segmentation of cell nuclei in virtual microscopy images: A minimum-model approach. *Scientific Reports*, 2(1), July 2012. doi: 10.1038/srep00503.

[29] M. Liao, Y. Q. Zhao, X. H. Li, P. S. Dai, X. W. Xu, and J. K. Zhang, et al. Automatic segmentation for cell images based on bottleneck detection and ellipse fitting. *Neurocomputing*, 173:615–622, January 2016. doi: 10.1016/j.neucom.2015.08.006.

[30] S. Chen, C. Ding, M. Liu, J. Cheng, and D. Tao. CPP-Net: Context-aware polygon proposal network for nucleus segmentation. *IEEE Transactions on Image Processing*, 32:980–994, 2023. ISSN 1941-0042. doi: 10.1109/TIP.2023.3237013.

[31] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022. doi: 10.1109/TPAMI.2021.3059968.

[32] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, and K. Chou, et al. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, January 2019. doi: 10.1038/s41591-018-0316-z.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. doi: 10.1038/nature14539.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24574-4_28.

[35] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnU-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, December 2020. doi: 10.1038/s41592-020-01008-z.

[36] B. S. Kelly, C. Judge, S. M. Bollard, S. M. Clifford, G. M. Healy, and A. Aziz, et al. Radiology artificial intelligence: a systematic review and evaluation of methods (RAISE). *European Radiology*, 32(11):7998–8007, April 2022. doi: 10.1007/s00330-022-08784-6.

[37] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9:82031–82057, 2021. doi: 10.1109/ACCESS.2021.3086020.

[38] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[39] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. doi: 10.1109/ICCV.2017.322.

[40] N. A. Koohbanani, M Jahanifar, A. Gooya, and N. Rajpoot. Nuclear instance segmentation using a proposal-free spatially aware deep learning framework. In *Lecture Notes in Computer Science*, pages 622–630. Springer International Publishing, 2019. doi: 10.1007/978-3-030-32239-7_69.

[41] Y. Song, E. L. Tan, X. Jiang, J. Z. Cheng, D. Ni, and S. Chen, et al. Accurate cervical cell segmentation from overlapping clumps in pap smear images. *IEEE Transactions on Medical Imaging*, 36(1):288–300, 2017. doi: 10.1109/TMI.2016.2606380.

[42] S. E. Raza, L. Cheung, M. Shaban, S. Graham, D. Epstein, and S. Pelengaris, et al. Micro-net: A unified model for segmentation of various objects in microscopy images. *Medical Image Analysis*, 52:160–173, February 2019. doi: 10.1016/j.media.2018.12.003.

[43] P. Naylor, M. Laé, F. Reyal, and T. Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE Transactions on Medical Imaging*, 38(2):448–459, 2019. doi: 10.1109/TMI.2018.2865709.

[44] M. Weigert and U. Schmidt. Nuclei Instance Segmentation and Classification in Histopathology Images with Stardist. In *2022 IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*, pages 1–4, March 2022. doi: 10.1109/ISBIC56247.2022.9854534.

[45] H. Chen, X. Qi, L. Yu, and P. Heng. Dcan: Deep contour-aware networks for accurate gland segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2487–2496. IEEE Computer Society, jun 2016. doi: 10.1109/CVPR.2016.273.

[46] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944. IEEE Computer Society, jul 2017. doi: 10.1109/CVPR.2017.106.

[47] T. Y. Lin, P. Goyal, R. Girshick, Ross K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. doi: 10.1109/tpami.2018.2858826.

[48] A. Nabila and N. M. Khan. 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)*, pages 683–687, 2019. doi: 10.1109/ISBI.2019.8759329.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and A. N. Gomez, et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and T. Unterthiner, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv Preprint*, June 2021. doi: 10.48550/arXiv.2010.11929.

[51] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, and P. Bojanowski, et al. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951.

[52] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[53] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu. Vit-yolo:transformer-based yolo for object detection. In *2021 IEEE/CVF International Conference on Computer*

Vision Workshops (ICCVW), pages 2799–2808, 2021. doi: 10.1109/ICCVW54120.2021.00314.

[54] L. Y. Chen and Q. Yu. Transformers make strong encoders for medical image segmentation. *arXiv*, February 2021. doi: 10.48550/arXiv.2102.04306.

[55] S. Li, X. Sui, X. Luo, X. Xu, Y. Liu, and R. Goh. Medical image segmentation using squeeze-and-expansion transformers. *arXiv*, May 2021. doi: 10.48550/arXiv.2105.09511.

[56] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu. Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 272–284. Springer International Publishing, 2022. doi: 10.1007/978-3-031-08999-2_22.

[57] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[58] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, and Y Wang, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. doi: 10.1109/CVPR46437.2021.00681.

[59] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[60] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. doi: 10.1109/CVPR42600.2020.00975.

[61] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[62] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, and E. Buchatskaya, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[63] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. doi: 10.1109/CVPR46437.2021.01549.

[64] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, and S. von Arx, et al. On the opportunities and risks of foundation models. *arXiv*, August 2021. doi: 10.48550/arXiv.2108.07258.

[65] P. Bankhead, M. B. Loughrey, J. A. Fernández, Y. Dombrowski, D. G. McArt, and P. D. Dunne, et al. QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7(1), December 2017. doi: 10.1038/s41598-017-17204-5.

[66] N. Kumar, R. Verma, D. Anand, Y. Zhou, O. F. Onder, and E.Tsougenis, et al. A multi-organ nucleus segmentation challenge. *IEEE Transactions on Medical Imaging*, 39(5):1380–1391, 2020. doi: 10.1109/TMI.2019.2947628.

[67] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017. doi: 10.1109/TMI.2017.2677499.

[68] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. doi: 10.1109/CVPR.2019.00963.

[69] K. Sirinukunwattana, S. E. A. Raza, Y. W. Tsang, D. Snead, I. A. Cree, and N. M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1196–1206, 2016. doi: 10.1109/TMI.2016.2525803.

[70] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and K. A. Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[71] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv*, November 2017. doi: 10.48550/arXiv.1711.05101.

Supplementary material

Table A.1: Average mPQ and bPQ across the 19 tissue types of the PanNuke dataset for 3-fold cross-validation for the ×20 data. For comparison, we also included the networks trained and evaluated on ×40 data in the first two columns. The standard deviation (STD) of the splits is provided in the final row. *Head & Neck

Tissue	$CellViT_{256}$		CellViT-SAM-H		$CellViT_{256}$		CellViT-SAM-H	
	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ	mPQ	bPQ
Adrenal	0.5186	0.7014	0.5248	0.7008	0.4027	0.5952	0.4334	0.6177
Bile Duct	0.5006	0.6345	0.5092	0.6421	0.3796	0.5096	0.4140	0.5394
Bladder	0.5851	0.6529	0.5721	0.6497	0.3385	0.4081	0.3561	0.4298
Breast	0.5165	0.6593	0.5228	0.6690	0.4363	0.5737	0.4704	0.6075
Cervix	0.5171	0.6024	0.5234	0.6138	0.4035	0.4874	0.4238	0.5095
Colon	0.4510	0.5263	0.4577	0.5392	0.3200	0.3959	0.3503	0.4261
Esophagus	0.5469	0.6573	0.5474	0.6636	0.4385	0.5440	0.4515	0.5658
H&N**	0.4924	0.5454	0.4996	0.5502	0.2725	0.3372	0.2956	0.3585
Kidney	0.5582	0.6643	0.5621	0.6806	0.3555	0.4603	0.3854	0.4800
Liver	0.5201	0.7179	0.5316	0.7268	0.3677	0.5295	0.3843	0.5569
Lung	0.4224	0.6268	0.4263	0.6331	0.2932	0.4088	0.3153	0.4343
Ovarian	0.5259	0.6539	0.5379	0.6661	0.4560	0.5821	0.4747	0.6101
Pancreatic	0.5065	0.6699	0.4997	0.6701	0.3743	0.5074	0.3820	0.5299
Prostate	0.5267	0.6583	0.5489	0.6727	0.3754	0.4995	0.3984	0.5210
Skin	0.4155	0.6202	0.4417	0.6289	0.3002	0.4422	0.3217	0.4702
Stomach	0.4544	0.7033	0.4691	0.7162	0.2985	0.5155	0.3292	0.5367
Testis	0.5157	0.6692	0.4978	0.6719	0.3810	0.5377	0.4232	0.5673
Thyroid	0.4792	0.6914	0.4771	0.6887	0.3892	0.5871	0.4056	0.6073
Uterus	0.4468	0.6270	0.4691	0.6384	0.3225	0.4635	0.3367	0.4970
Average	0.5000	0.6464	0.5062	0.6538	0.3634	0.4939	0.3869	0.5192
STD	0.0156	0.0092	0.0145	0.0078	0.0155	0.0122	0.0108	0.0083

Table A.2: Selected data augmentation techniques with probability and additional hyperparameters. Data augmentation is implemented with Albumentations.

Augmentation Technique	Probability	Hyperparamter
90-degree rotation	0.5	
Horizontal flipping	0.5	
Vertical flipping	0.5	
Downscaling	0.15	max-scale: 0.5 min-scale: 0.5
Blurring	0.2	blur-limit: 10
Gaussian noise	0.25	var_limit: 50
Color jittering	0.2	brightness: 0.25 contrast: 0.25 saturation: 0.1 hue: 0.05
Superpixel representation	0.1	p_replace: 0.1 n_segments: 200 max-size: $H/2$
Zoom blur	0.1	max-factor: 1.05
Random cropping with resizing	0.1	crop-level: 0.5-1.0 of input size
Elastic transformation	0.2	sigma: 25 alpha: 0.5 alpha-affine: 15
Normalization	1.0	Mean: [0.5, 0.5, 0.5] STD: [0.5, 0.5, 0.5]

Table A.3: Hyperparameters for all training runs on the PanNuke dataset

Parameter	Value
Loss	$\lambda_{NP_{FT}} = 1$, $\lambda_{NP_{FT}} = 1$, $\lambda_{NP_{DICE}} = 1$, $\lambda_{HV_{MSE}} = 2.5$, $\lambda_{HV_{MSGE}} = 8$, $\lambda_{NT_{FT}} = 0.5$, $\lambda_{NT_{DICE}} = 0.2$, $\lambda_{NT_{BCE}} = 0.5$, $\lambda_{TC_{CE}} = 0.1$, $\alpha_{FT} = 0.7, \beta_{FT} = 0.3, \gamma_{FT} = 4/3, \varepsilon_{FT} = 1 \cdot 10^{-6}$
Sampling	$\gamma_s = 0.85$
Optimizer	AdamW[71]
Training	$\eta = 3 \cdot 10^{-4}$, $\lambda = 1 \cdot 10^{-4}$, $\beta_1 = 0.85, \beta_2 = 0.85$, epochs = 130, batch-size = 16, lr-scheduling = 0.85

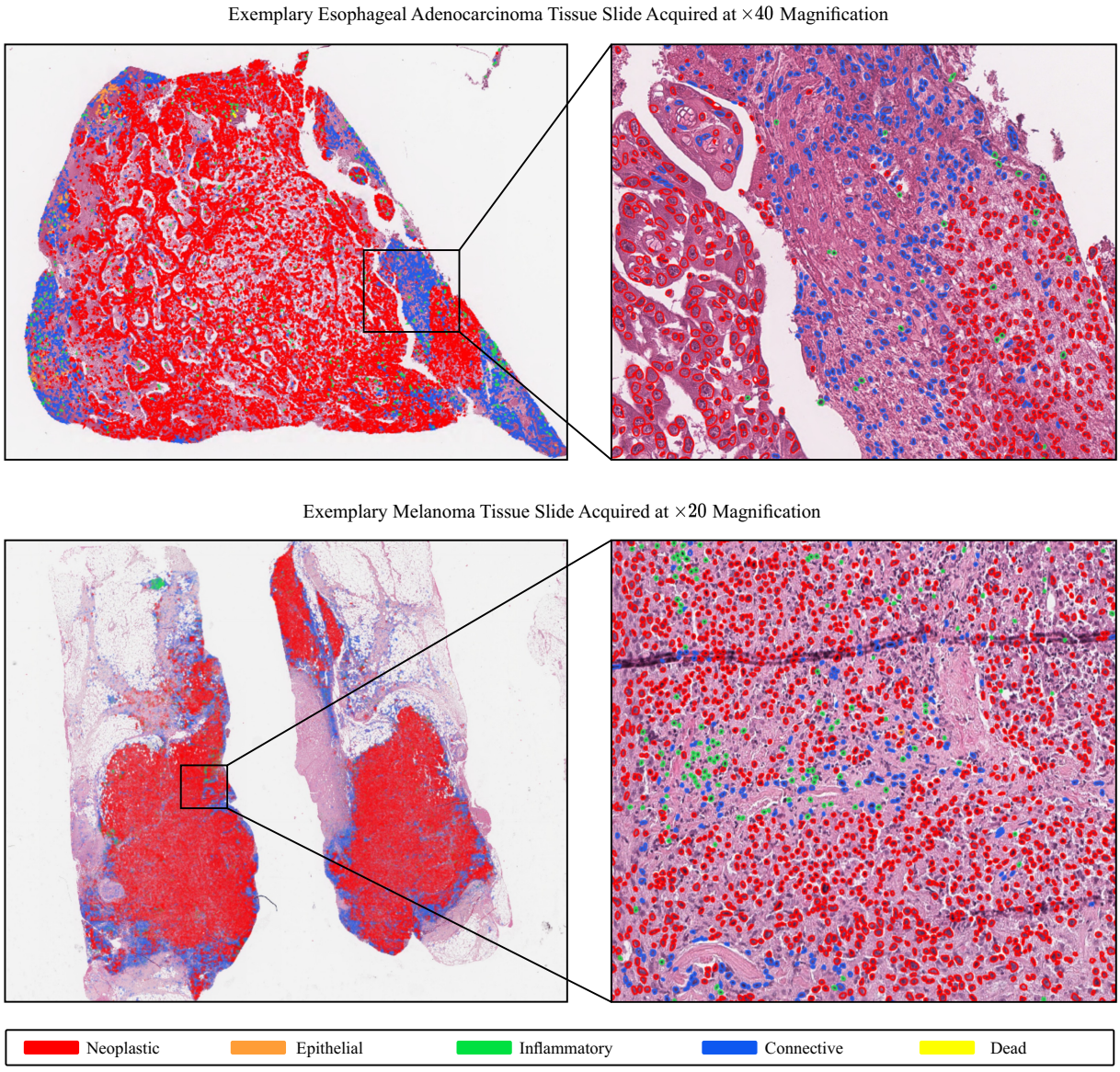

Figure A.1: Exemplary WSI files with corresponding cell polygons imported into QuPath to show the interoperability of our inference pipeline. For each of the files, approximately 150,000 nuclei have been detected, which can be imported into QuPath without any performance problems regarding fast file loading and zooming on a standard laptop.