Computerized Medical Imaging and Graphics 107 (2023) 102238

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag

Valuing vicinity: Memory attention framework for context-based semantic segmentation in histopathology

Oliver Ester [a,b], Fabian Hörst [a,b,*], Constantin Seibold [c], Julius Keyl [a,d], Saskia Ting [d,e], Nikolaos Vasileiadis [f], Jessica Schmitz [f], Philipp Ivanyi [g], Viktor Grünwald [b,h], Jan Hinrich Bräsen [f], Jan Egger [a,b], Jens Kleesiek [a,b,i]

[a] Institute for AI in Medicine (IKIM), University Hospital Essen (AöR), Essen, Germany
[b] Cancer Research Center Cologne Essen (CCCE), West German Cancer Center Essen, University Hospital Essen (AöR), Essen, Germany
[c] Institute of Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[d] Institute of Pathology, University Hospital Essen (AöR), University of Duisburg–Essen, Essen, Germany
[e] Institute of Pathology Nordhessen, Kassel, Germany
[f] Nephropathology Unit, Institute for Pathology, Hannover Medical School, Hannover, Germany
[g] Department of Hematology, Hemostasis, Oncology and Stem Cell Transplantation, Hannover Medical School, Hannover, Germany
[h] Clinic for Medical Oncology, Clinic for Urology, West German Cancer Center, University Hospital Essen (AöR), Essen, Germany
[i] German Cancer Consortium (DKTK), Partner Site Essen, Germany

ARTICLE INFO

Keywords:
Semantic segmentation
Computational pathology
Histopathology
Context
Renal cell carcinoma

ABSTRACT

The segmentation of histopathological whole slide images into tumourous and non-tumourous types of tissue is a challenging task that requires the consideration of both local and global spatial contexts to classify tumourous regions precisely. The identification of subtypes of tumour tissue complicates the issue as the sharpness of separation decreases and the pathologist's reasoning is even more guided by spatial context. However, the identification of detailed tissue types is crucial for providing personalized cancer therapies. Due to the high resolution of whole slide images, existing semantic segmentation methods, restricted to isolated image sections, are incapable of processing context information beyond. To take a step towards better context comprehension, we propose a patch neighbour attention mechanism to query the neighbouring tissue context from a patch embedding memory bank and infuse context embeddings into bottleneck hidden feature maps. Our memory attention framework (MAF) mimics a pathologist's annotation procedure — zooming out and considering surrounding tissue context. The framework can be integrated into any encoder–decoder segmentation method. We evaluate the MAF on two public breast cancer and liver cancer data sets and an internal kidney cancer data set using famous segmentation models (U-Net, DeeplabV3) and demonstrate the superiority over other context-integrating algorithms — achieving a substantial improvement of up to 17% on Dice score. The code is publicly available at https://github.com/tio-ikim/valuing-vicinity.

1. Introduction

In the digital age of histopathology, specialized scanners digitize a tissue specimen with suspected cancer into an image at high magnification, resulting in a whole slide image (WSI). In the slides, tumourous tissue can be identified, graded and the most promising therapy recommended. The response of therapies is yet not fully understood and more research about the tumour microenvironment (TME) is ongoing (Junttila and De Sauvage, 2013). One line of research is the cell analysis of tissue subtypes. For this, detailed identification of tissue types is crucial, on which we focus in this work.

Fig. 1 shows an example hematoxylin and eosin (H&E) WSI with renal cell cancer (RCC) and its corresponding extensive subtype annotations. To identify a specific type of tissue, a pathologist examines a slide section at high magnification and then considers neighbouring tissue to integrate context information into the decision. As the manual annotation process is a tedious, complex task, there is ongoing research for developing WSI segmentation algorithms. The most promising algorithms for WSI segmentation are based on supervised, parameterizable convolutional neural networks (CNN) (Long et al., 2015; Ronneberger et al., 2015; Chen et al., 2017; Jégou et al., 2017), that are trained on a set of WSIs and their ground truth annotations. However, since a single

* Corresponding author at: Institute for AI in Medicine (IKIM), University Hospital Essen (AöR), Essen, Germany.
E-mail address: fabian.hoerst@uk-essen.de (F. Hörst).

https://doi.org/10.1016/j.compmedimag.2023.102238
Received 13 February 2023; Received in revised form 11 April 2023; Accepted 25 April 2023
Available online 11 May 2023

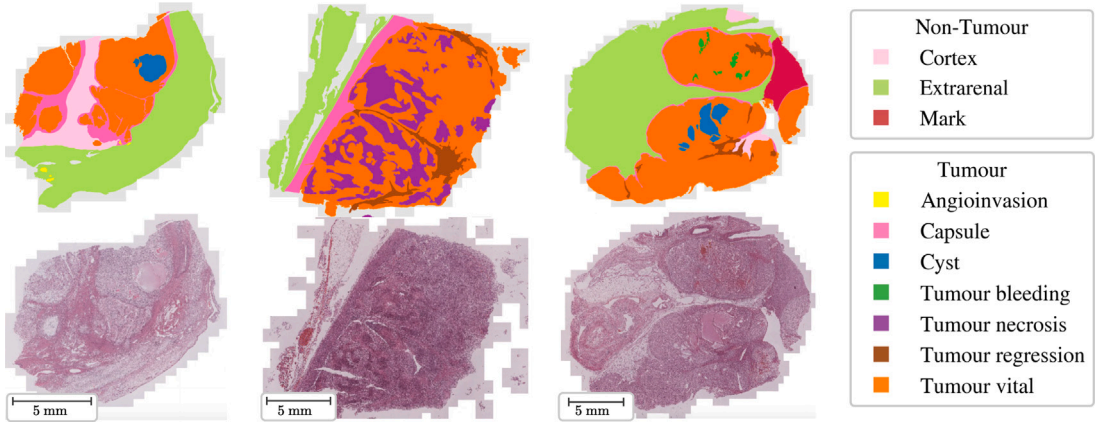

Fig. 1. RCC examples — Top: Annotations of tumourous and non-tumourous subtypes. Bottom: H&E WSIs with kidney cancer.

WSI at the highest resolution exceeds the hardware limits of current GPUs, most methods decompose WSIs into a set of patches while trying to find a balance between a narrower field of view (FOV) with less context information and a lower physical image resolution with fewer tissue details (Wang et al., 2019). Either way, the algorithm is withheld with crucial context or detailed tissue information.

In this work, we present a novel memory extension framework for CNN encoder–decoder architectures in semantic segmentation to improve the exploitation of context information around high-resolution WSI patches. Our framework is able to segment an entire WSI patch-wise, but leverages vicinity information to improve and spatially align segmentation masks across patches. The proposed framework is based on the attention mechanism and can be incorporated into the bottleneck layer of any encoder–decoder segmentation method.

1.1. Related work

Semantic segmentation is the computer vision task of assigning each pixel to its corresponding class. Due to the dense prediction characteristic and often cohesive label regions (annotations), specialized segmentation CNNs with an encoder–decoder structure are used. The encoder derives an internal image representation, which is then upsampled into a segmentation mask by the decoder. Current state-of-the-art (SOTA) segmentation networks are mainly based on the U-Net (Ronneberger et al., 2015) or DeepLabV3 (Chen et al., 2017) architecture, both CNN-based encoder–decoder networks. Variants of these networks, especially of the U-Net, have been successfully applied in medical imaging segmentation tasks for various image modalities, such as the nnU-Net (Isensee et al., 2020) for radiological image data or a U-Net for nuclei segmentation of multi-organ histopathology images (Kiran et al., 2022).

Unlike segmenting radiological images like CT- or MRI scans, the semantic segmentation of entire WSIs poses a considerable challenge: Due to the huge resolution of WSIs (larger than 150k × 150k px per WSI), applying SOTA segmentation algorithms directly onto the slide is impeded by current GPU hardware limitations. Therefore, in the infancy of semantic segmentation of WSIs, segmentation algorithms were applied in a patch-wise manner (Wang et al., 2019). As this is sufficient for segmentations of local isolated objects such as cells (Graham et al., 2019, 2023; Ilyas et al., 2022; Kiran et al., 2022), the patch-wise approach leads to contextual and structural problems for segmentations of connected tissue structures on an entire tissue sample. Jin et al. (2020) showed that patch-wise WSI segmentation quality heavily depends on the selection of two patch hyperparameters – *downsampling (resolution)* and *FOV (spatial extent of context)*. To overcome the selection process, they developed a foveation module, which learns to dynamically select the best trade-off between both hyperparameters.

Other works exploit context information to enlarge the FOV while preserving a detailed resolution. To do so, concentric context patches with lower physical resolution around a central patch are additionally fed into a neural network, and context information is merged. Li et al. (2019) extracted small and large-size image patches to cover cell-level and tissue-level features and used them for breast cancer classification. Nonetheless, using patch classification algorithms to create segmentation maps based on patch class predictions is insufficient because the results are either coarse or a tremendous computational effort for sliding window approaches to generate detailed segmentation maps is required (Van Rijthoven et al., 2021; Li et al., 2017). Alternatively, for the segmentation of histological images of prostatectomies, a U-Net with three input images with a different FOV was used and the context fused at the input layer (Li et al., 2017). However, the runtime is three times higher than the baseline U-Net, and the context expansion is limited to the input layers. Related approaches fusing the context information of different FOVs at the U-Net bottleneck layer use dedicated encoder networks for each input and context patch (Tokunaga et al., 2019; Gu et al., 2018). However, the performance increase compared to baseline U-Net and DeeplabV3 is only marginal. The Hooknet (Van Rijthoven et al., 2021) with two parallel context (low-resolution) and target (high-resolution) U-Net branches leverages a different fusion architecture. The wider FOV of the context branch is hooked into the target branch by spatially aligning the context feature map of the bottleneck layer to the target feature map. At the same time, Schmitz et al. (2021) developed msY-Net with two U-Net encoder branches. Similarly to Hooknet, target and context embeddings are fused at the bottleneck layer by spatial alignment and cropping, but msY-Net uses one combined decoder branch to retrieve the segmentation mask. Both models achieve SOTA performance, but the Hooknet is less memory efficient and has significantly more trainable parameters due to the additional context decoder branch (Schmitz et al., 2021).

Even though CNN models achieve exceptional image processing capabilities, they have limitations in modelling long-range spatial relationships because their convolutional operations are bounded to local receptive fields (Chen et al., 2021). In contrast, the adaption of the Transformer architecture (Vaswani et al., 2017) into the computer vision field by Dosovitskiy et al. (2020) has led to network structures solely based on the self-attention mechanism without leveraging convolutions to capture long-range dependencies. The attention mechanism processes input in the form of tokens represented as feature vectors and calculates global relationships among them via a weighted sum. This is achieved using learnable attention weights to identify the most significant regions or features. While common Transformer segmentation networks without CNNs suffer from coarse segmentations when trained with limited data (Strudel et al., 2021), various ideas about exploiting the attention mechanism in combination with CNNs were developed (Wang et al., 2021; Li et al., 2021; Chen et al., 2021; Xie

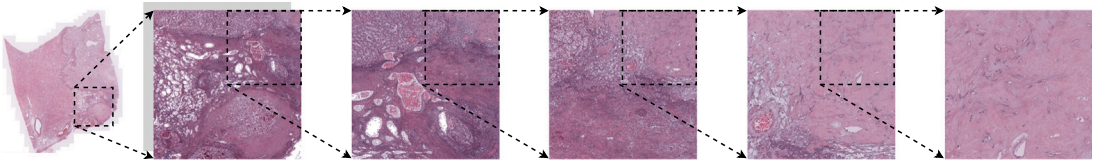

Fig. 2. Example of patches with 256 × 256 px and different pixel resolutions (→ different FOVs). From left to right: Thumbnail, 22.14 µm, 11.07 µm, 5.53 µm, 2.77 µm, 1.38 µm.

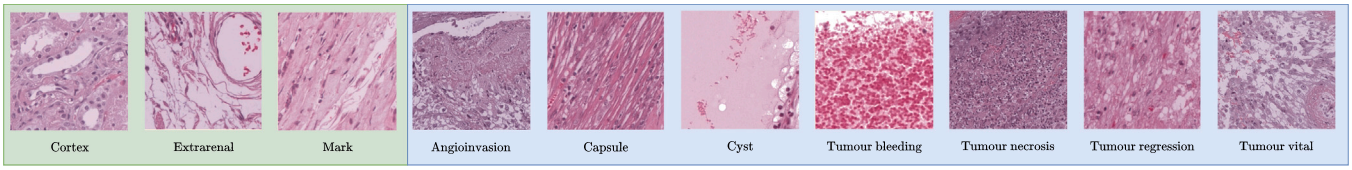

Fig. 3. Patch example for each type of tissue with an edge length of 256 px and 35,328 µm (1.38 µm/px). Non-tumourous Tumourous.

et al., 2021; Zheng et al., 2021). Mostly, these methods integrate Transformer modules into encoder–decoder architectures to increase the receptive field using attention mechanisms. Chen et al. (2021) added a Transformer into the U-Net bottleneck. Thereby, the self-attention mechanism can attend to all (spatial) features of the feature map within an input image. Guo et al. (2021) introduced an external attention mechanism to exploit potential correlation with other samples by attending an embedding memory in the U-Net bottleneck layer to catch the most essential information of the entire data set. Based on this external attention mechanism, Wang et al. (2021) proposed a Mixed Transformer Module that extends the encoder and decoder depths. Their module consists of three attention mechanisms: a local, a global, and an external. The local one attends the close context by a local window constraint — the global one attends tokens by a row and column constraint globally (but still inside an image patch) – the external one attends a memory of queries and keys of the entire data set. While the previous works have made significant advances in addressing long-range and spatial dependencies in CNN-based networks utilizing the attention mechanism, they are still based on the assumption that an entire image can be processed in one forward pass. Either the transformer is used to attend tokens within an image (Chen et al., 2021; Li et al., 2021), or to incorporate information from the entire data set (Wang et al., 2021; Guo et al., 2021).

1.2. Our contribution

Our work attempts to retain fine-grained segmentation of WSIs, addressing the problem of incorporating context information into SOTA patch-wise segmentation networks. Previous works (Schmitz et al., 2021; Van Rijthoven et al., 2021) solved the context problem mainly by using context patches with a large FOV but required additional computational effort due to the context encoder. Contrarily, we propose a method based on an external context memory that stores compressed representations of entire WSIs and incorporates the tissue context without using additional context patches. We achieve this by using the attention mechanism to incorporate global dependencies between the central patch and surrounding tissue.

The memory approach is inspired by the external data set memory of Guo et al. (2021). Differing from their work, we build a memory for each WSI with spatial aligned patch representations, such that a high resolution input patch can query information from surrounding context patches. Related to Chen et al. (2021), we integrate the attention mechanism in the bottleneck of an encoder–decoder architecture. For the interaction between the external memory and the attention mechanism, our architecture attends neighbouring patch embeddings stored in the external memory while segmenting a given patch. Our patch neighbour **M**emory **A**ttention **F**ramework (MAF) enriches out-of-sample context information into the patch segmentation — thus **Valuing Vicinity**. We compare the MAF's capability to baseline U-Net and DeepLabV3 and the SOTA context-integrating msY-model of Schmitz et al. (2021) on three cancer data sets: renal cell cancer (internal), breast cancer (public), and liver cancer (public).

We summarize our contributions as:

1. We propose a novel out-of-sample attention-based extension for arbitrary encoder–decoder architectures and a corresponding memory framework to integrate neighbourhood context information from a memory into the patch segmentation process without using dedicated context patches.
2. We demonstrate that our method can be beneficial to tissue segmentation of kidney cancer and tumour segmentation of breast cancer and liver cancer — outperforming baseline models and the context-integration model of Schmitz et al. (2021).

2. Materials

We benchmark the MAF on an internal kidney cancer data set with renal cell carcinoma (RCC) and exhaustive annotations of various tumourous and non-tumourous types of tissue but also show the contribution of the MAF on a subset of the breast cancer data set from the CAMELYON16 (CY16) challenge (Bejnordi et al., 2017) following Schmitz et al. (2021) as well as on a liver cancer segmentation data set from the PAIP 2019 challenge (Kim et al., 2021).

2.1. RCC

The RCC data set consists of 175 WSIs of patients with metastatic renal cell carcinoma undergoing nephrectomy.[1] One representative archival paraffin block of the tumour was selected and sections of all specimens were stained with H&E using routine procedures. The slides were digitized at a pixel resolution of 0.1729 µm. We show a tissue example at different pixel resolutions in Fig. 2.

Annotation The annotation of 10 different tissue types was performed using QuPath (Bankhead et al., 2017) under the close supervision of a trained nephropathologist. First, the tissue area was detected by thresholding followed by manual verifications. Subsequently, the tissue mask was manually separated into either tumourous or non-tumourous types: The tumourous regions were subclassified into either *Tumour vital*, *Tumour regression*, *Tumour necrosis*, *Tumour bleeding*, *Angioinvasion*, *Capsule* or *Cyst* (resulting in 7 types of tumour tissue), the non-tumourous regions into either *Extrarenal*, *Cortex* or *Mark* (resulting in 3 types of non-tumour tissue). Fig. 3 shows patch examples of size 256 × 256 px of each type of tissue at a pixel resolution of 1.38 µm – resulting in an edge length of 35,272 µm.

[1] Approved by local ethics board: 8682_BO_K_2019, 10183_BO_K_2022.

2.2. CY16

We also validate our results on a publicly available data set of the CAMELYON16 challenge. The Cancer Metastases in Lymph Nodes (CAMELYON) 2016 challenge (Bejnordi et al., 2017) provided WSIs of sentinel lymph nodes with and without metastases of breast cancer. Following Schmitz et al. (2021), we select the same 20 WSIs with at least one lymph node macrometastasis. The slides were digitized at a pixel resolution of 0.243 μm. In the following, we refer to the subset of CAMELYON16 WSIs as CY16. Note, that the complexity of detecting tumourous tissue in the CY16 data set is assumed to be lower than the complexity of identifying all subtypes of tissue in the RCC data set due to the RCC's high histopathologic heterogeneity (Cai et al., 2020).

Annotation The data set provides annotations of the metastatic tissue. To come up with a tissue annotation, we followed Schmitz et al. (2021) and applied thresholding using QuPath and created a *Healthy* tissue class by subtracting the *Tumour* annotation from the tissue annotation – resulting in 2 tissue types – *Tumour* and *Healthy*.

2.3. PAIP 2019

As another public segmentation data set for benchmarking, we incorporate the PAIP 2019 (liver cancer) data set in our analysis. The data set consists of WSI acquired at a resolution of 0.502 μm, containing 50 train, 10 validation, and 40 test images. In line with Schmitz et al. (2021), we limit our analyses to the train data since the ground truth masks of the test data were not publicly accessible at the time of this publication.

Annotation The authors of PAIP 2019 provide annotations of the whole tumour area including viable tumour cell nests, tumour necrosis, and tumour capsule (Kim et al., 2021) as well as just of the viable tumour area without surrounding stroma, tumour necrosis, and tumour capsule. We subtracted the viable tumour mask from the whole tumour mask to generate two exclusive tumour classes and used the proposed thresholding protocol of Kim et al. (2021) to generate tissue masks. In total, we generated three mutually exclusive tissue annotations for this data set out of the provided annotations: *Healthy, Viable Tumour* and *Non-Viable Tumour*.

2.4. Patch extraction

For all data sets, we split each WSI into non-overlapping patches using OpenSlide (Goode et al., 2013), resized their original pixel resolution (downsampling) and applied the Macenko-normalization (Macenko et al., 2009). We omitted all patches with no annotation overlap (→ background).

To analyse the impact of physical resolution versus FOV, we created multiple patch sets using different downsampling factors (ds), all of pixel size $256 \times 256 \times 3$ in RGB colour space. A larger ds corresponds to a larger FOV and a lower pixel resolution. Concurrently, for comparison with context-integrating models, context patches concentric to the central patch with identical pixel size but a larger FOV were extracted (related to Schmitz et al. (2021)). For the targets, we proceeded the same with the annotation masks and additionally enriched the patch metadata with the percentage class ratio, used as helper target.

3. Methods

In this section, we propose the **M**emory **A**ttention **F**ramework (MAF) for semantic segmentation of WSIs after first contextualizing the domain with preliminary definitions.

3.1. Preliminaries

Let $w \in \mathbb{R}^{M \times N \times 3}$ be a WSI w with (M, N) spatial dimensions and three colour channels. Each w is divided into a set of quadratic, non-overlapping patches $\mathcal{P} = \{p_{i,j}\}$, $p_{i,j} \in \mathbb{R}^{S \times S \times 3}$, where i denotes the column position and j the row position in a uniform, two-dimensional grid with the dimensions n_x and n_y. In the case of multi-label segmentation with C classes, the label for a patch $p_{i,j}$ is defined as the segmentation mask $y^{\text{seg}}_{i,j} \in \{0,1\}^{S \times S \times C}$. Additionally, we define a second label $y^{\text{cls}}_{i,j} \in \mathbb{R}^C$ as the class distribution of all pixels in a patch. Similar to common segmentation methods (Fan et al., 2020; Zhao et al., 2017; Ronneberger et al., 2015; Chen et al., 2017), an encoder f_{enc} first maps a patch $p_{i,j}$ into a set of feature maps $\mathcal{F} = \{feat_0, \dots, feat_{l-1}, feat_l\}$ at different depth levels l. A decoder f_{dec} then learns to map $\mathcal{F}$ to the segmentation $\hat{y}^{\text{seg}}_{i,j}$.

3.2. Memory attention framework

Fig. 4 shows an overview of our MAF based on a common encoder–decoder architecture but extended by a patch memory and an attention mechanism. In our architecture, each patch is segmented individually but the framework enables an information flow of neighbourhood patches into each patch. The neighbourhood memory attention mechanism exploits the deepest feature maps $feat_l$ from the encoder and creates a context-modified version $feat'_l$ which updates the set of feature maps $\mathcal{F}$ to $\mathcal{F}' = \{feat_0, \dots, feat_{l-1}, \boldsymbol{feat}'_l\}$ before being decoded into $y_{i,j}$. $\mathcal{F}'$ now is aware of context information.

Patch embedding To store every patch of a WSI w in reasonable memory size, we learn compressed patch representations. For that, each patch $p_{i,j}$ is fed into the encoder f_{enc} and a learnable compression function f_{emb} (adaptive average pooling + linear projection) maps the features $feat_l \in \mathbb{R}^{D_{\text{hid}} \times m \times m}$ into an embedding $e_{i,j} \in \mathbb{R}^{D_{\text{MAF}}}$.

Embedding memory A spatial embedding memory $\mathcal{M} \in \mathbb{R}^{(n_x+2k) \times (n_y+2k) \times D_{\text{MAF}}}$ for a WSI w stores all patch embeddings $e_{i,j}$. At the boundary, we add a padding of size k to the memory dimensions for neighbourhoods exceeding the WSI region (see Fig. 5).

1. Memory insert: The memory $\mathcal{M}$ first has to be filled up, for each WSI w, by applying f_{enc} and f_{emb} to each patch $p_{i,j}$ and inserting the resulting compressed embeddings $e_{i,j}$ in $\mathcal{M}$. We update the memory once at the start of each epoch in a gradient-free forward pass balancing outdated embeddings with resource-expensive updates. For the first epoch, to exclude random initialized context information, $\mathcal{M}$ is initialized with empty embedding vectors $e_{i,j} = [0 \dots 0]$

2. Memory retrieval: After the completion of the memory fill-up, we can retrieve the embeddings of the patch neighbourhood for a complete forward pass. This two-step forward pass is inevitable for the training and also test phase but marginally affects the run-time due to the first gradient-free forward pass allowing for larger batch size (see 5.5). The backward pass does not affect the memory $\mathcal{M}$ instantly but merely updates the encoder weights and eventually leads to a delayed, epoch-wise update of the memory embeddings. We also experimented with an *online* memory retrieval (memory is not filled beforehand but the corresponding neighbourhood embeddings are determined in one forward pass) which suffers from exploding repetitions of encoder runs and exploding memory consumption with increasing neighbourhood size while, advantageously, offers *up-to-date* embeddings due to the direct effect of the backpropagation. However, we could not find any model performance increase.

Patch neighbourhood The neighbourhood $\mathcal{N}$ defines the context information that can be reached by a patch p. We hypothesize that a larger neighbourhood provides more context information and therefore should improve the segmentation quality of p. We define a concentric patch neighbourhood in the embedding space as

$$\mathcal{N}^k_{i,j} = \{e_{i',j'} | i' \in [i-k, \dots, i+k], j' \in [j-k, \dots, j+k]\}, \quad (1)$$

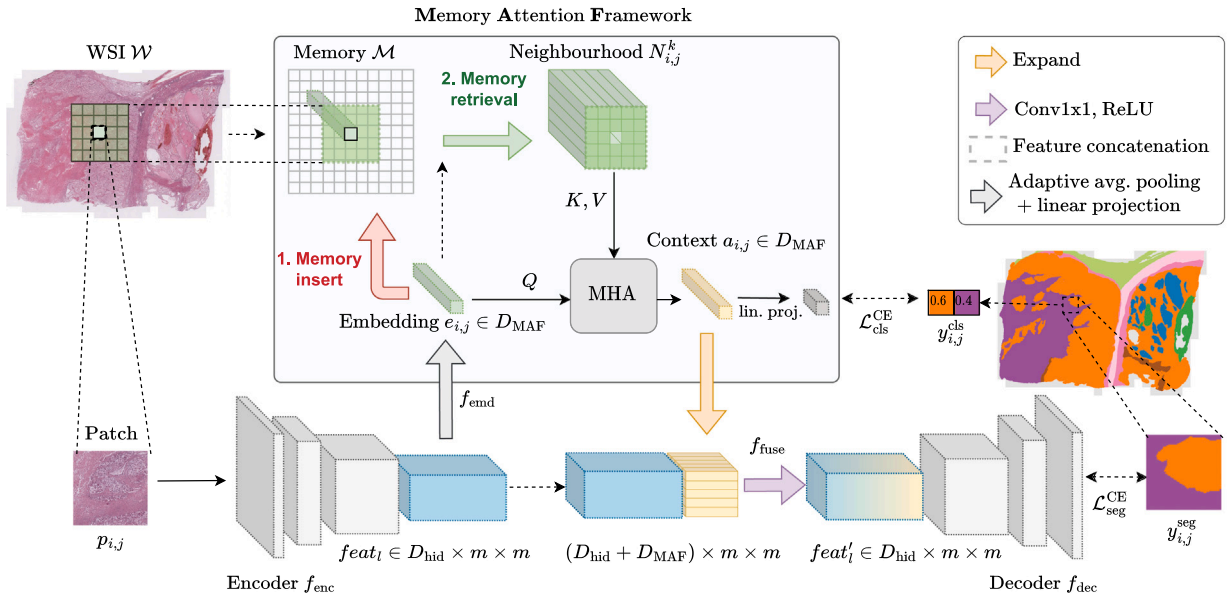

Fig. 4. Overview of neighbourhood memory attention framework (MAF): An encoder–decoder architecture is extended by a patch memory attention mechanism to fuse context information into the segmentation process. 1. The memory is built up in a patch-wise encoder run by compressing each patch through a gradient-free encoder-run (*Memory insert*). 2. For each patch, the neighbourhood is retrieved from the embedding memory (*Memory retrieval*), attended and the context embedding merged into the decoder run. A helper loss supports the context learning by predicting the patch's class ratios from the context embedding. For illustration purposes, we use a neighbourhood radius k of 2.

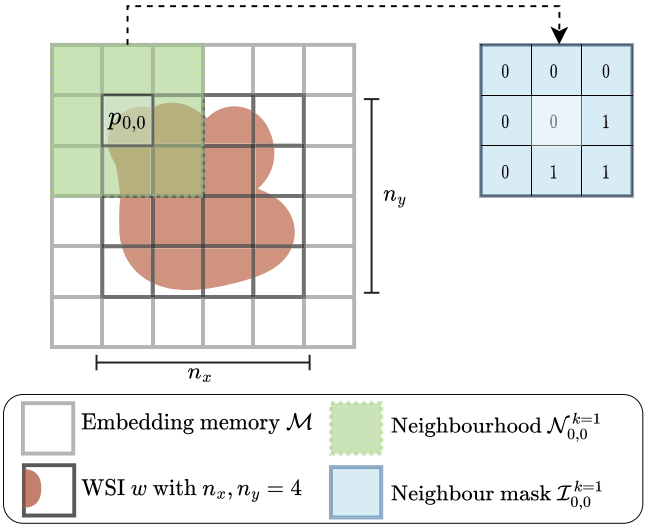

Fig. 5. An example of a patched WSI w with patch dimensions $n_x \times n_y = 4 \times 4$, a neighbourhood radius $k = 1$ and the memory $\mathcal{M}$ with dimensions $(n_x{+}2k){\times}(n_y{+}2k){\times}D = (4 + 2 \cdot 1) \times (4 + 2 \cdot 1) \times D = 6 \times 6 \times D$. For the patch $p_{0,0}$ and its neighbourhood $\mathcal{N}^{k=1}_{0,0}$, we can derive the neighbour mask $\mathcal{I}^{k=1}_{0,0}$. The centre of the neighbour mask is always 0 by definition.

$$\mathcal{N}^k_{i,j} \subseteq \mathcal{M}$$

for the central patch $p_{i,j}$, which is subject to the neighbourhood radius k. To handle non-existing neighbour patches (e.g., at boundary areas or background patches), we also define a binary neighbour mask

$$\mathcal{I}^k_{i,j} = \{\mathbb{1}_{\mathcal{M}}(i', j') | i' \in [i-k, \dots, i+k], j' \in [j-k, \dots, j+k]\}$$

$$\mathbb{1}_{\mathcal{M}}(i', j') = \begin{cases} 0 & \text{if } \nexists e_{i',j'} \text{ or } (i', j') = (i, j) \\ 1 & \text{else,} \end{cases} \tag{2}$$

where the mask is 1 if a patch embedding $e_{i',j'}$ exists in memory $\mathcal{M}$. We set $\mathbb{1}_{\mathcal{M}}(i, j)$ constant to 0 to exclude $p_{i,j}$ from its own neighbourhood and thereby avoid self-attention (see Fig. 5).

Neighbourhood attention We hypothesize that the relevance of context information varies over space and type of tissue and thus should be learnable: Hence, we make use of a Multi-Head-Attention (*MHA*) module (adjusted from Vaswani et al., 2017) to enable the central patch $p_{i,j}$ in form of its embedding $e_{i,j}$ to query its neighbourhood $\mathcal{N}^k_{i,j}$ and obtain a context embedding $a_{i,j} \in \mathbb{R}^{D_{\text{MAF}}}$ (see example in Fig. 6). The module learns to select and aggregate relevant context information for a central patch with the patch embedding $e_{i,j}$:

For each head h, we project $e_{i,j}$ to the query vector $q_h \in \mathbb{R}^{D_{\text{attn}}}$, with D_{attn} the hidden attention dimension, and $\mathcal{N}^k_{i,j}$ to the keys and values matrices K_h and V_h and define the attention function as:

$$\text{Attention}(q_h, K_h, V_h) = \text{softmax}(\mathcal{I}^k_{i,j} \frac{q_h K_h^T}{\sqrt{d_h}}) V_h, \tag{3}$$

where we mask the logits with our neighbour mask $\mathcal{I}^k_{i,j}$. We alter the keys to K'_h by adding position embeddings to K_h (see Positional Encoding) and calculate the context embedding $a_{i,j}$ as:

$$a_{i,j} = \text{Proj}(\underset{1..h}{\text{Concat}}(\text{Attention}(q_h, K'_h, V_h))) \tag{4}$$

Throughout all experiments, we use $h = 8$ and $d = 128$ as the hidden dimension for K, V and q. In practice, we compute the MHA on a set of queries, keys, and values simultaneously. Note, that the embedding $e_{i,j}$ origins from the second step of the forward pass and passes the gradients to the encoder while $\mathcal{N}^k_{i,j}$, being retrieved from the memory $\mathcal{M}$, is gradient-free.

Positional encoding The MHA enables the central patch $p_{i,j}$ to attend patches in the neighbourhood. However, in its raw form, the token order in K_h and V_h is permutation invariant and thus lacks spatial awareness. We, therefore, add position embeddings to the keys. Following Ramachandran et al. (2019), we introduce learnable 2D position embeddings $B = \{b_{i',j'} | i' \in [-k, k], j' \in [-k, k]\}$ with $b_{i,j} \in \mathbb{R}^{D_{\text{attn}}}$ and relative patch coordinates i', j' to the central patch $p_{i,j}$ (see example in Fig. 6). Each embedding $b_{i',j'}$ is a concatenation of a row-offset representation and a column-offset representation $b_{i',j'} = \text{Concat}(b^{\text{row}}_{i'}, b^{\text{col}}_{j'})$, with $b^{\text{row}}_{i'}$ and $b^{\text{col}}_{j'}$ being learnable parameters. We add B to K_h and receive position-aware keys K'_h. The position encodings are shared over the heads.

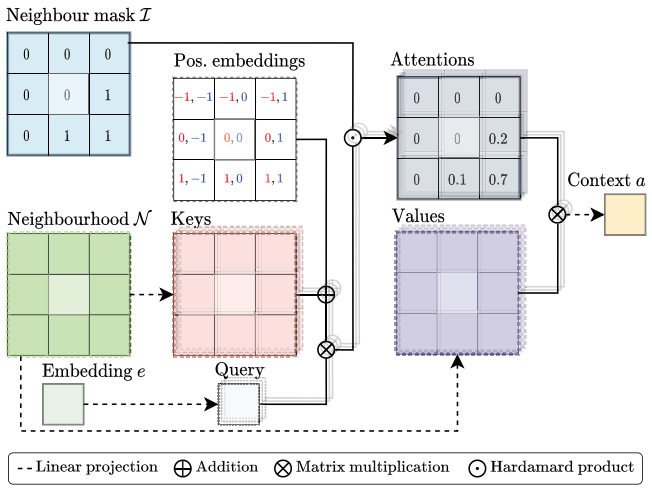

Fig. 6. An example of the MHA mechanism ($k = 1$): The embedding e is linearly projected to the query, the neighbourhood $\mathcal{N}$ to keys and values, for each head respectively. We determine the attention scores by applying softmax to the matrix multiplication of the query with position-aware keys — considering valid keys only by multiplying with the neighbour mask. Finally, we weight the values by the attention scores to receive the context embedding a.

We also experimented with common fixed sinusoidal embeddings (Vaswani et al., 2017) and no position embeddings.

Context fusion Eventually, a function f_{fuse} fuses the context embedding $a_{i,j}$ into the feature maps $feat_l \in \mathbb{R}^{D_{\text{hid}} \times m \times m}$, resulting in $feat'_l$. It expands (copies) $a_{i,j} \in \mathbb{R}^{D}_{\text{MAF}}$ over the dimensions $m \times m$, concatenates with $feat_l$ and applies a 1×1 convolution over the dimensions $D_{\text{hid}} + D_{\text{MAF}}$ with output dimension D_{hid}. $\mathcal{F}$ is updated to $\mathcal{F}' = \{feat_0, \dots, feat_{l-1}, \boldsymbol{feat}'_l\}$ and f_{dec} then predicts the segmentation mask $\hat{y}_{i,j}$ from $\mathcal{F}'$. We also experimented with different convolution kernel sizes but did not observe any significant differences.

Targets and losses The patch segmentation predictions $\hat{y}^{\text{seg}}_{i,j}$ are optimized on the common cross entropy loss $\mathcal{L}^{\text{CE}}_{\text{seg}}(\hat{y}^{\text{seg}}_{i,j}, y^{\text{seg}}_{i,j})$ given the patch segmentation labels $y^{\text{seg}}_{i,j}$. Inspired by Mehta et al. (2018), we introduce a helper classification loss $\mathcal{L}^{\text{CE}}_{\text{cls}}(\hat{y}^{cls}_{i,j}, y^{\text{cls}}_{i,j})$ that aims to optimize the class distribution prediction $y^{\text{cls}}_{i,j}$ given the true class distribution $\hat{y}^{cls}_{i,j}$. The class distribution prediction results from a linear projection of the context embedding $a_{i,j}$ – the output of the MHA module. Note, that the central patch is always excluded from the neighbourhood – so we hypothesize that predicting the central patch class distribution given the neighbourhood context only, can guide the loss in the MHA. We combine both losses using a weight λ as

$$\mathcal{L}^{+} = (1 - \lambda) \cdot \mathcal{L}^{\text{CE}}_{\text{seg}} + \lambda \cdot \mathcal{L}^{\text{CE}}_{\text{cls}} \tag{5}$$

and will refer to MAF^{+} when optimizing on combined losses.

4. Experiments

4.1. Experimental setup

Baseline and methods We evaluate our method on three histopathological data sets, a breast cancer (see CY16 2.2), a liver cancer (see PAIP 2019 2.3), and a kidney cancer data set (see RCC 2.1), and employ two baselines using patch-based segmentation architectures, namely U-Net (Ronneberger et al., 2015) and DeepLabV3 (Chen et al., 2017). To show the general applicability of our new method, we extend both encoder–decoder architectures with the MAF and compare them with the patch-based baselines. Finally, we compete our approach against another context-integrating architecture, the msY-Net (Schmitz et al., 2021) which learns from an additional context patch with larger FOV.

Patch-based baseline: We first set up patch-based segmentation baselines to later measure the isolated effect of context contribution. Since *the best* FOV for a patch remains unclear, we first evaluate the baseline models on different ds factors – keeping the patch dimension constant at 256 × 256 px. Also, we test different encoder backends – namely ResNet-50 and ResNet-18.

Context benchmark: We then set up a SOTA context-integrating benchmark using the msY-Net which processes two patches – a central patch (×1) and a context patch with the same patch dimension (256 × 256) but larger FOV (×4) – in two parallel encoders. The msY-Net is built upon the baseline U-Net architecture. To take the parallel encoder paths of the msY-Net and its increase in learnable parameters into consideration, we doubled the learning rate for all msY-Net experiments. Since the msY-Net implementation is optimized for ResNet-18, we will compare it to ResNet-18 versions of the U-Net and DeepLabV3 architectures for a fair comparison.

We note that Schmitz et al. (2021) validate the performance on randomly sampled image crops and on full-scale WSIs ($ds = 1$). For a competing benchmark, we, therefore, retrained the msY-Net maintaining our training procedure, and evaluated all models on the identical WSI data.

Model training All models are based on ImageNet (Russakovsky et al., 2015) pre-trained ResNet (He et al., 2016) encoders. The encoder hidden dimension D_{hid} is subject to the encoder network structure. Our default MAF is configured with an embedding size of $D_{\text{MAF}} = 1024$, MHA with 8 heads, a hidden attention dimension of $D_{\text{attn}} = 128$, learnable 2D position encodings, and a neighbourhood radius $k = 8$.

Patch sampling: To cope with the class imbalance and the excessive number of patches, we use a patch-sampling mechanism for the training and validation phase: To avoid losing important information about infrequent classes, we randomly draw at most 100 patches per WSI and tissue class — assigning each patch its predominant tissue class. For comparison, we ensure that all models are trained with the same patch sample, for each fold respectively. At test time, we evaluate the model on all patches of each WSI in the test set.

Data augmentation: For all models, we restrict to colour jitter only since spatial augmentations (flipping, rotating, cropping) might lead to spatial inconsistencies between a central patch and its neighbourhood memory. For context patches in the msY-Net, we ensure the identical random augmentation as its central patch.

Optimization: We use SGD with a momentum of 0.9 and a learning rate of 0.0001 (0.0002 for msY-Net) with exponential learning rate decay of $\beta = 0.95$. Each model is trained with a batch size of 32 for 100 epochs and its validation loss is determined after each epoch. Early stopping is applied if the validation loss does not further decrease for 10 epochs. At the best validation loss, the model is evaluated.

Implementation All models were implemented in PyTorch 1.10.0. using *Segmentation Models* (Yakubovskiy, 2020). For the MAF, we implemented a combined memory $\mathcal{M}$ for all WSIs in a data set, for each phase respectively, using PyTorch Tensors to allow an efficient, simultaneous access of all neighbourhood embeddings for one batch (cross-WSI). The memory is deployed on the GPU memory (optionally on the CPU memory if size exceeds VRAM). Depending on the number of WSIs in the train set $\mathcal{W}$, the maximum patch dimensions $n_{x,\max} = \max(n_x)$ and $n_{y,\max} = \max(n_y)$ from the entire train data set, the embedding size D_{MAF} and the neighbourhood radius k, the physical memory size of $\mathcal{M}$ can be determined as:

$$|\mathcal{W}| \cdot (2k + n_{x,\max}) \cdot (2k + n_{y,\max}) \cdot D_{\text{MAF}} \cdot 4\,\text{B/FP32}$$

Thus, the complete WSI train set can be compressed to a memory size feasible for a modern GPU. e.g., for the RCC data set with 112 training

WSI at $ds = 16$ and $k = 8$, the physical size of the embedding memory $\mathcal{M}$ for the *entire* train set is

$$112 \cdot (2 \cdot 8 + 65) \cdot (2 \cdot 8 + 36) \cdot 1024 \cdot 4\,\text{B/FP32} = 1.93\,\text{GB},$$

compared to 9.60 GB needed to store all 256 × 256 px RGB patches (CY16: 9.02 GB vs. approx. 41.34 GB for 12 WSIs at $ds = 2$, PAIP 2019: 2.30 GB vs. approx. 73.48 GB for 32 WSIs at $ds = 2$).

The memory $\mathcal{M}$ of the MAF is updated at the start of each epoch. All patches of all WSIs are compressed by the model in PyTorch evaluation mode with disabled gradient calculation (see 3.2). We deviate from the patch sampling to ensure a complete neighbourhood for every sampled patch querying the memory. For the first epoch, $\mathcal{M}$ is initialized with `torch.zeros()` not to use context information initially.

For the msY-Net, we used the model implementation of the official repository. All experiments were run on a 48 GB NVIDIA RTX A6000.

Validation We run each experiment with 5-fold cross validation strategy (CV) – splitting on the level of WSI – resulting in 140/35 WSIs in the training/test set per fold for the RCC, 16/4 WSIs in the training/test set per fold for the CY16, and 40/10 WSIs in the training/test set per fold for the PAIP 2019 data set. For early stopping, we use 20% of the training set — resulting in 28 WSIs (RCC), 4 WSIs (CY16) and 8 WSIs (PAIP 2019) in the validation set, respectively. For comparison between all experiments, we ensure the same splits between all models, respectively per fold.

Evaluation metrics We measure the semantic segmentation performance with the micro-average Dice Similarity Coefficient (DSC) for each tissue class c as $DSC_c = \overline{m}^{\mathcal{W}}_{\text{NaN}}(DSC(c, w))$ and in total as $DSC_{\text{total}} = \overline{m}^{\mathcal{W}}_{\text{NaN}}(\overline{m}^{\mathcal{C}}_{\text{NaN}}(DSC(c, w)))$ with $\overline{m}_{\text{NaN}}$ as the mean function for all defined DSCs.

We estimate the model performance with the mean DSC over all folds defined as $\overline{DSC}$ alongside with its standard deviation.

5. Results

5.1. Determining patch-based baselines

To benchmark the performance gain of enabling the MAF, we first determined the performance of patch-based segmentation models and studied the effect of expanding the FOV (increasing ds) – without enabling the MAF.

For the *RCC* data set, both, U-Net and DeepLabV3, in combination with both encoders, ResNet-18 and ResNet-50, consistently performed best at $ds = 16$ (see Appendix Table A.4). The combination of DeepLabV3 and ResNet-50 yielded the best results with 0.50 $\overline{DSC}_{total}$. Zooming out too far (fewer details — larger FOV) significantly deteriorates performance.

For the *CY16* data set, we observe a strong performance (0.73 $\overline{DSC}_{Tumour}$) of patch-based models at $ds = 2$ using the U-Net with ResNet-18 (see Appendix Table A.5).

Similar results are obtained for the PAIP 2019 data set with 0.70 $\overline{DSC}_{Total}$ for a U-Net with ResNet18 at $ds = 2$ (see Appendix Table A.6), thus, in line with CY16, $ds = 2$ is used for further analysis.

Hence, we select $ds = 16$ for all RCC MAF experiments and $ds = 2$ for all CY16 MAF and PAIP 20919 MAF experiments.

5.2. MAF model variations

Next, we studied the effect of altering architectural concepts in our MAF — based on the RCC data set and the best baseline setup with DeepLabV3 and ResNet-50 encoder at $ds = 16$. We analyse the effect of the memory embedding size, position encodings, helper loss, and the neighbourhood size in a gradual manner. We start with a default MAF experiment setup using $D = 1024$, 1D position encodings, $\lambda = 0$ (no helper loss), and $k = 8$, and gradually alter the concepts.

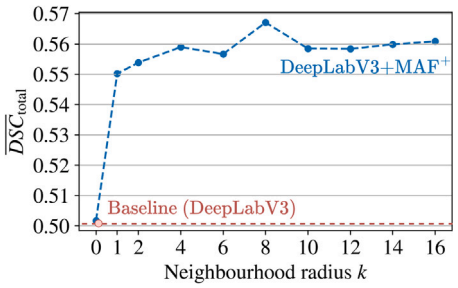

Fig. 7. $\overline{DSC}_{\text{total}}$ at different neighbourhood radii for RCC with DeepLabV3+MAF$^+$ at $ds = 16$.

Effect of memory embedding size We analysed the impact of the embedding dimension D_{MAF} and hypothesize that a higher D_{MAF} allows for compressing more information into the memory. Table 1 shows that the performance is best for $D_{\text{MAF}} = 1024$. Increasing it to 2048 deteriorates the performance — assuming that too large embeddings might lead to overfitting. On the other hand, decreasing D_{MAF} to 512 also deteriorate the performance – assuming too little semantic space for storing the compression. Subsequently, for all following experiments, we use $D_{\text{MAF}} = 1024$.

Effect of positional encoding Subsequently, we analysed the effect of positional encodings. We compare the performance of applying 1D sinusoidal embeddings (Vaswani et al., 2017) with using no position embeddings and applying relative 2D learnable embeddings (Ramachandran et al., 2019). Table 1 shows that the relative 2D learnable embeddings add most benefit, followed by 1D sinusoidal embeddings. Using no position embeddings performs worst — still outperforming the 0.50 $\overline{DSC}_{total}$ baseline score by +0.03.

Effect of helper loss We hypothesized that a loss $\mathcal{L}^{\text{CE}}_{\text{cls}}$ predicting the central patch's class distribution from the context embedding might guide the MHA module, as stated in Eq. (5). Table 1 shows the results for different weight factors λ. We observe the best performance for $\lambda = 0.2$ and freeze it for the following experiments (referred to MAF$^+$).

Effect of neighbourhood radius To better understand the effect of the neighbourhood attention on the segmentation performance, we compared the DeepLabV3+MAF$^+$ performance using different neighbourhood radii k (see Fig. 7). Note that $k = 0$ equals DeepLabV3 w/o MAF. While most performance gain is observed from $k = 0 \rightarrow k = 1$ (using no neighbourhood → adjacent patches only), we observe a further increase of performance until $k = 8$ followed by a subtle lower performance with $k > 8$. Still, all versions of MAF$^+$ with $k > 0$ outperform the baseline significantly. We decided to freeze $k = 8$ for all following experiments. The resulting neighbourhood size is comparable to a FOV expansion from $ds = 16$ to $ds = 512$, thus, increasing the FOV by ×32.

5.3. RCC baseline

Using the best performing MAF$^+$ setting (patch embedding dimension $D_{\text{MAF}} = 1024$, learnable 2D position encoding, helper loss with $\lambda = 0.2$, neighbourhood radius $k = 8$), we compare its class-wise performance on the RCC data set against the patch-based baselines in Table 2, both for U-Net and DeepLabV3 base architectures. We found both – U-Net+MAF$^+$ (0.50 $\overline{DSC}_{\text{total}}$) and DeeplabV3+MAF$^+$ (0.57 $\overline{DSC}_{\text{total}}$) – outperforming their baseline by 0.02 for U-Net and 0.07 (+14%) for DeepLabV3. The former (U-Net+MAF$^+$) improves best in the segmentation of *Capsule* tissue by 0.07 but disimproves in the segmentation of *Tumour Bleeding* by 0.04. The latter (DeeplabV3+MAF$^+$)

Table 1
Analysis of variations for DeepLabV3+MAF with $ds = 16$ and $k = 8$ on RCC, using ResNet50 encoder.

D_{MAF}			Pos. enc.			$\mathcal{L}_{cls}$ with λ			$\overline{DSC}_{total}$
512	1024	2048	no pos.	sin. 1D pos.	rel. 2D pos.	0.2	0.5	0.8	
✓	–	–	–	✓	–	–	–	–	0.5315
–	✓	–	–	✓	–	–	–	–	**0.5470**
–	–	✓	–	✓	–	–	–	–	0.5398
–	✓	–	✓	–	–	–	–	–	0.5312
–	✓	–	–	–	✓	–	–	–	**0.5517**
–	✓	–	–		✓	✓	–	–	**0.5725**
–	✓	–	–	–	✓	–	✓	–	0.5574
–	✓	–	–	–	✓	–	–	✓	0.5583
									0.5017[a]

[a]DeepLabV3 baseline.

Table 2
RCC – 5-fold CV $\overline{DSC}_c$ of baseline U-Net and DeepLabV3 without MAF and with MAF$^+$ extension. Setting: $ds = 16$, $k = 8$, $\lambda = 0.2$, $D_{MAF} = 1024$, ResNet50, and rel. 2D position encoding. (*Angioinv.* excluded since all architectures failed to detect).

	Tissue class	U-Net	U-Net + MAF$^+$	Δ	DeepLabV3	DeepLabV3 + MAF$^+$	Δ
Tumour	Vital	0.90 ± 0.02	0.91 ± 0.02	+0.01	0.90 ± 0.02	0.91 ± 0.02	+0.01
	Regression	0.46 ± 0.04	0.51 ± 0.04	+0.05	0.47 ± 0.04	0.52 ± 0.04	+0.05
	Necrosis	0.23 ± 0.06	0.22 ± 0.06	−0.01	0.24 ± 0.07	0.30 ± 0.08	+0.06
	Bleeding	0.22 ± 0.06	0.21 ± 0.06	−0.01	0.25 ± 0.04	0.31 ± 0.05	+0.06
	Capsule	0.25 ± 0.02	0.35 ± 0.02	+0.10	0.29 ± 0.23	0.36 ± 0.03	+0.07
	Cyst	0.00 ± 0.00	0.00 ± 0.00	+0.00	0.02 ± 0.03	0.05 ± 0.05	+0.03
Non-T[a]	Cortex	0.51 ± 0.07	0.53 ± 0.07	+0.02	0.52 ± 0.07	0.56 ± 0.09	+0.04
	Mark	0.28 ± 0.05	0.30 ± 0.04	+0.02	0.30 ± 0.05	0.38 ± 0.04	+0.08
	Extrarenal	0.66 ± 0.02	0.68 ± 0.02	+0.02	0.67 ± 0.03	0.70 ± 0.04	+0.03
	WSI[b]	0.48 ± 0.01	0.50 ± 0.01	+0.02	0.50 ± 0.01	**0.57** ± 0.01	+0.06

[a]Non-Tumour.
[b]$\overline{DSC}_{total}$.

Table 3
Context-integrating contribution of MAF$^+$ compared with msY-Net and baseline U-Net and DeepLabV3 (all based on ResNet-18). Difference Δ refers to the corresponding baseline structure without MAF$^+$ on each data set.

Data set	RCC ($ds = 16$)		CY16 ($ds = 2$)		PAIP 2019 ($ds = 2$)	
Data set score	$\overline{DSC}_{Total}$	Δ	$\overline{DSC}_{Tumour}$	Δ	$\overline{DSC}_{Total}$	Δ
U-Net	0.45 ± 0.02		0.73 ± 0.06		0.70 ± 0.01	
U-Net + MAF$^+$	0.47 ± 0.01	+0.02	0.75 ± 0.08	+0.02	0.73 ± 0.01	+0.03
DeepLabV3	0.49 ± 0.02		0.76 ± 0.07		0.73 ± 0.02	
DeepLabV3 + MAF$^+$	**0.54 ± 0.01**	**+0.05**	**0.80 ± 0.09**	**+0.04**	**0.77 ± 0.01**	**+0.04**
msY-Net	0.46 ± 0.01		0.79 ± 0.08		0.73 ± 0.02	

shows improvements for every subtype of tissue with the largest performance gain for *Mark* tissue by 0.11. All methods – baseline and MAF$^+$ – could not detect the minority classes *Angioinvasion* yielding 0.00 $\overline{DSC}_{Angioinvasion}$.

5.4. Context benchmark

Next, we benchmark the context integration effect of the MAF with the effect of the msY-Net (based on ResNet-18) proposed by Schmitz et al. (2021) on all three data sets. For comparison, we changed all encoders to ResNet-18 (Table 3).

For **RCC** with $ds = 16$, the U-Net baseline reaches 0.45 $\overline{DSC}_{total}$. The msY-Net model marginally outperforms the baseline with 0.46 $\overline{DSC}_{total}$ (+0.01). The U-Net+MAF$^+$ scores at 0.47 $\overline{DSC}_{total}$ (+0.02). DeepLabV3+MAF$^+$ (0.54 $\overline{DSC}_{total}$) significantly outperforms the baseline by 0.09 (+%20) and the msY-Net by 0.08 (+17%). In-depth tissue subtype results can be found in Table A.7 in Appendix.

For **CY16** with $ds = 2$, we report the *Tumour* $\overline{DSC}$ following Schmitz et al. (2021). The U-Net baseline scores at 0.73 $\overline{DSC}_{Tumour}$. Our msY-Net model outperforms the baseline by 0.06 with 0.79 $\overline{DSC}_{Tumour}$. We observe the U-Net+MAF$^+$ outperforming the U-Net baseline by 0.03 with 0.76 $\overline{DSC}_{Tumour}$, however does not reach the msY-Net performance. Finally, we show that DeepLabV3+MAF$^+$ scores best with 0.80 $\overline{DSC}_{Tumour}$ marginally outperforming the msY-Net.

For **PAIP 2019** with $ds = 2$, the msY-Net outperforms the U-Net baseline by 0.03 reaching a $\overline{DSC}_{Total}$ of 0.73. This performance is also achieved by our MAF framework in combination with the U-Net. However, changing the segmentation network structure from U-Net to DeepLabV3, our MAF reaches 0.77 $\overline{DSC}_{Total}$, outperforming all other networks and improving the DeepLabV3 baseline by 0.04 $\overline{DSC}_{Total}$.

5.5. Runtime comparison

Incorporating the MAF for the RCC data set (with 112 training WSIs) required an additional training time of 219.10 s per epoch for DeeplabV3 and 102.55 s for U-Net, both using ResNet50 encoder. This corresponds to a 25% (DeepLabV3) and 38% (U-Net) epoch runtime increase since baseline runtime for DeepLabV3 was 868.32 s and for U-Net 265.40 s. With a ResNet18 encoder, the training time per epoch with MAF for the DeepLabV3 network increased by 34% to 412.20 s. For the U-Net, we observed an increase of 43% to a runtime of 199.06 s. In comparison, the msY-Net training epoch time with equal settings was 399.75 s which is much higher than the U-Net with MAF.

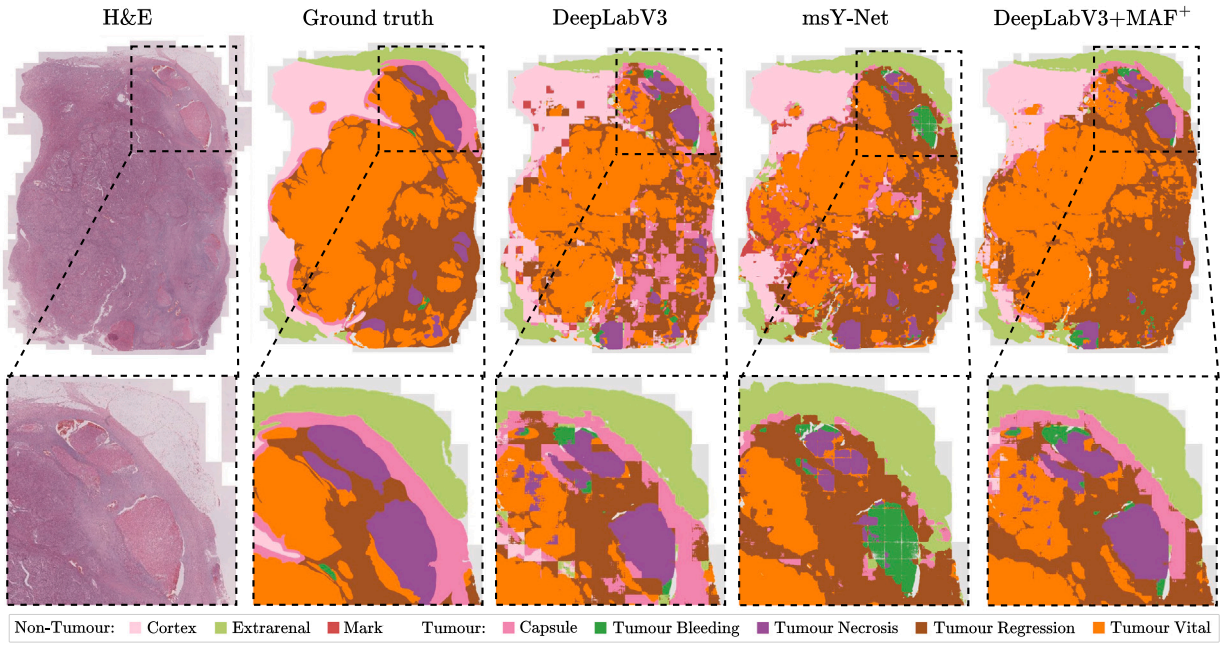

Fig. 8. Comparison of RCC segmentation with $ds = 16$ at different magnifications (all based on ResNet-18).

Due to architectural differences, the DeepLabV3 runtime, on the other hand, was higher than the msY-Net runtime, but also achieved superior performance (+0.09 $\overline{DSC}_{\text{total}}$).

For the CY16 data set, the effect of the MAF on training runtime (with 12 WSI in train set) is much higher. Overall, the DeepLabV3 and U-Net average training runtime of 53.37 s (DeepLabV3) and 24.24 s (U-Net) was nearly triplet when using MAF. The increase originates from shorter training epochs because CY16 just contains the two tissue types *tumour* and *healthy*, each sampled 100 times per WSI. The memory, on the other hand, was still built for entire WSIs, resulting in computational overhead for inserting unused embeddings. The msY-Net required 39.89 s per training epoch, almost twice as long as the baseline U-Net network with ResNet18 encoder. The runtime behaviour observed for the PAIP 2019 data set exhibits a similar trend, albeit with less pronounced effects. In general, the relative time required for building up the memory compared to one training epoch is primarily influenced by the size of the training data set and the sampling strategy. If the training data set is small with few segmentation classes and just a small subset of all patches are sampled during a training epoch, the upfront time for building up the MAF memory becomes more significant.

5.6. Qualitative results

To visually perceive the benefit of the MAF, we show an example of an RCC segmentation of one WSI based on DeepLabV3+MAF$^+$ with $k = 8$ and $ds = 16$ in Fig. 8 and compare it to a patch-based DeepLabV3 with $ds = 16$ and the msY-Net with $ds = 16$ for the centre patch and $ds = 64$ for the context patch. Zooming in reveals that some patch segmentations of DeepLabV3 without the MAF are entirely wrong — resulting in a scattered, non-cohesive segmentation map. The msY-Net improves in detecting more cohesive tissue but is more error-prone for tissue confusion. Integrating the MAF almost solves this issue. The attention mechanism identifies cohesive tissue much better. We provide more qualitative results for the RCC data set in the Appendix (see Fig. A.13). Fig. 9 shows a segmentation result for the CY16 data set. While the U-Net patch-based approach wrongly detects scattered tumourous tissue patches, both – msY-Net and DeeplabV3+MAF$^+$ – improve their precision and are more aware of coherent tissue sections. Examples for the PAIP 2019 data set are given in Fig. A.14 in Appendix.

5.7. Analysis of attention

Enabling the attention mechanism shows an improvement in segmentation performance since context information can be retrieved from the neighbourhood. To better understand which information is used, we visualize the attention scores over the queried neighbourhood. All attention visualizations are based on DeepLabV3+MAF$^+$ with learned 2D position encodings.

Attention maps We enlarge an RCC segmentation result and visualize a cut-out in size of the corresponding neighbourhood — here $k = 8$. For each central patch, we can retrieve the resulting attention values of the MHA in the MAF. We average the values over all 8 heads and highlight the neighbourhood with respect to their relative attention importance. Fig. 10 shows the resulting attention map. Interestingly, we can observe close-by neighbours attracting more attention. Also, the attention is not concentric to the central patch but rather biased by the patches' tissue classes. The segmentation of this patch without the MAF is of lower quality, as shown in part (b) of Fig. 10.

Attention views To further understand the attention mechanism, we create a view of aggregated attention maps over a WSI (see. Fig. 11). We first average the attention scores over all heads and yield an aggregated 2D attention map with scores summing up to 1. Subsequently, for each central patch $p_{i,j}$, we fit the parameters of a bivariate normal distribution using least-squares on the attention scores — for illustration purposes, simplified assuming the attention scores to form an empirical bivariate normal distribution around the central patch. Fig. 11(d) shows two example central patches and their attending neighbourhood with its characteristic 90% confidence ellipse (axis in the direction of Eigenvectors scaled by the square root of the Eigenvalue times $\chi^2_{\alpha=0.9}(r = 2)$, respectively). We also show the shift of the ellipse centre from the patch centre as a blue arrow — indicating the attention focus. To come up with an aggregated attention view, for each central patch, we plot a miniature version (Fig. 11(c)) of its characteristic ellipse — colouring the ellipse area by its deviation from the mean area (■: > mean area → long reach attention, ■: < mean area → close reach attention) and indicating its attention focus with a black arrow representing the ellipse centre shift. We provide more attention views in the Appendix (see Fig. A.12).

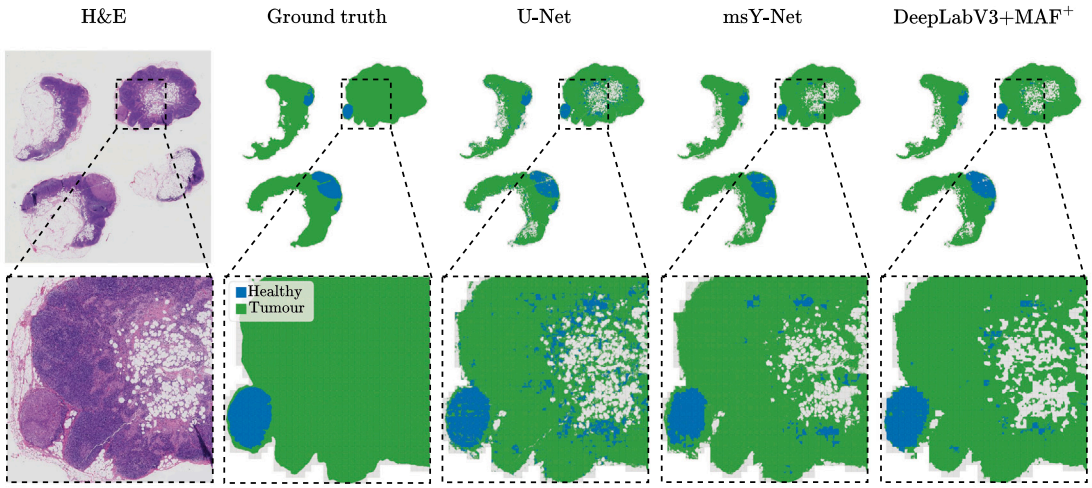

Fig. 9. Comparison of CY16 segmentation with $ds = 2$ at different magnifications (all based on ResNet-18).

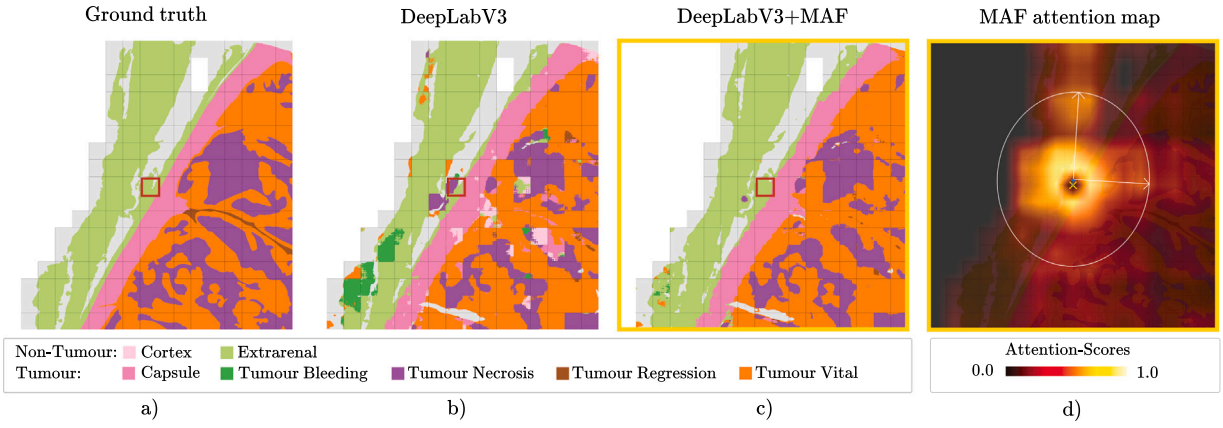

Fig. 10. Example of RCC segmentation details and attention map of the central patch. (a) Ground truth (d) Segmentation prediction of DeepLabV3 (c) Segmentation prediction of DeepLabV3+MAF (b) Attention map with respect to the central patch. □ Central patch □ Attended neighbourhood of central patch.

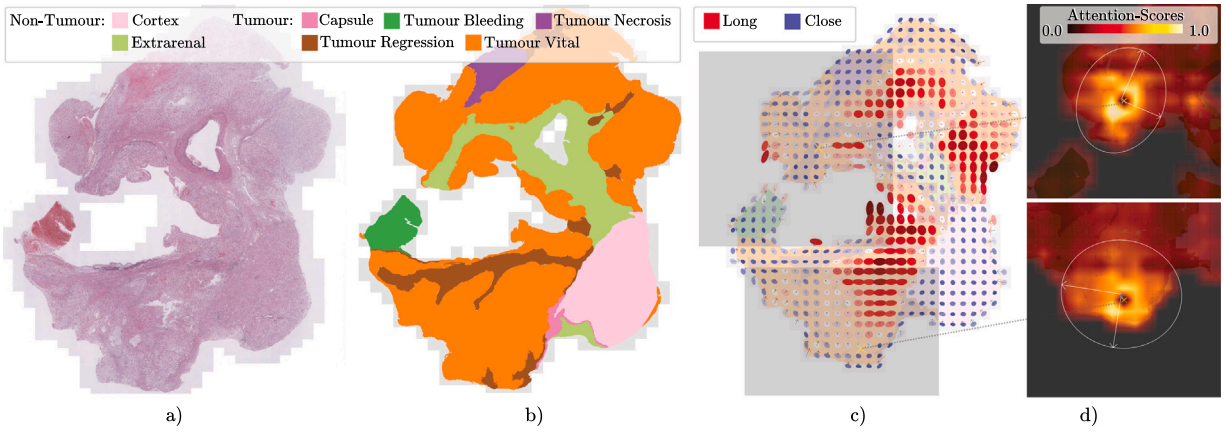

Fig. 11. Example of aggregated attention view: (a) Tissue (b) Annotations (c) Segmentation prediction with aggregated attention view. For each patch p^c, we determine its characteristic ellipse from the attention of the neighbourhood. We colourize each ellipse with respect to its deviation of area to the mean area (■: > mean area → long reach attention, ■: < mean area → close reach attention) and plot its miniature version for each patch. (d) Attention heatmaps and its characteristic ellipse for a central patch p^c (×).

6. Discussion and conclusions

We proposed a semantic segmentation extension for WSIs using a patch neighbour attention mechanism that queries the neighbouring tissue context from an embedding memory bank and infuses context embeddings into bottleneck feature maps. We showed that our approach is superior to patch-based segmentation algorithms and even outperforms a SOTA context-integrating algorithm in a multi-class cancer data set. Our MAF facilitates a much wider FOV to access context information compared to SOTA — while simultaneously processing details on a patch level. In addition, the MAF is able to learn, first, a compressed form of the context and, second, to selectively attend to relevant context information. We show that the MAF is robust in terms of neighbourhood size.

One could observe that the performance boost of the MAF on the RCC segmentation is superior to the CY16 and PAIP 2019 segmentation. This is reasonable as first, the RCC segmentation task is more complex due to its multiple subtypes of tissue and second, the tumour identification of macrometastasis (CY16) can be achieved at cell level from a pathologists perspective, thus less favourable for the MAF. Due to more distinct tumour classes, the complexity of the PAIP 2019 dataset is higher than that of the CY16 dataset, and similarly, we observed a stronger improvement by the MAF on the more complex PAIP 2019 data. We conclude that the MAF is more beneficial to the segmentation of complex tissue structures where a human pathologist needs to make use of context information (zooming out and considering surrounding tissue context). Our visualization of the attention maps – mimicking a pathologist's view – supports this conclusion.

From a pathologist's view, the attention views show interesting characteristics: The attention reach of *Vital Tumour* exceeds all other tissue types. At the tumour border lamella, we can see an intensified reach of attention clearly with a focus on the border — indicating the detection of the tumour border. Also, we can observe an increase of attention reach for areas with subtle type of tissue borders (e.g., *Cortex* → *Mark*) – indicating the intensified usage of attention for regions with clear context needs. We can also observe intensified attention reach for tissue with morphological heterogeneity – e.g., in *Extrarenal* tissue – indicating the access to more context information in case of indecisiveness using the centre patch information only. On the other hand, the attention reach is close for tissue regions with morphological homogeneity e.g., *Extrarenal* fat tissue. As stated by our pathologist, we can observe similar usage of context information compared to their workflow of examining tissue types.

Applying the MAF to DeepLabV3 shows a larger benefit than to U-Net and we assume that architectural elements (e.g., atrous spatial pyramid pooling) favour the context fusion. Different fusing mechanisms and encoder–decoder architectures should therefore be studied. In addition, we applied one MHA layer only. We plan to change it to a more receptible Transformer encoder to extend the attention capability. A recently published work (Zaffar et al., 2022) on augmentations techniques in embedding spaces may provide an additional method applied to our embedding memory for improving the segmentation results. Future works see an additional evaluation on the upcoming test PAIP 2019 when it gets available. In addition, we will continue to work on the efficiency of our method to avoid building up the embedding memory for entire WSIs if a limited sampling strategy for network training is applied (e.g., CY16 data set). In future work, we believe the memory can be exploited even further to better process context information – e.g., by integrating the context embeddings into different decoder levels or fully Transformer-based architectures. Also, we believe that it will help with memory-expensive 3D segmentation tasks by applying the MAF on image slices. To tackle the trade-off between FOV and physical resolution, research about hierarchical memory attention mechanisms storing patches each at multiple FOVs seems a promising direction.

CRediT authorship contribution statement

Oliver Ester: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Fabian Hörst:** Methodology, Software, Writing – original draft, Writing – review & editing, Visualization. **Constantin Seibold:** Conceptualization, Writing – review & editing. **Julius Keyl:** Validation, Writing – review & editing. **Saskia Ting:** Validation, Writing – review & editing, Visualization. **Nikolaos Vasileiadis:** Data curation, Writing – review & editing. **Jessica Schmitz:** Investigation, Writing – review & editing, Project administration. **Philipp Ivanyi:** Data curation, Writing – review & editing. **Viktor Grünwald:** Writing – review & editing, Supervision. **Jan Hinrich Bräsen:** Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Jan Egger:** Writing – review & editing, Supervision. **Jens Kleesiek:** Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

This work received funding from 'KITE' (Plattform für KI-Translation Essen, Germany) from the REACT-EU initiative (https://kite.ikim.nrw/, EFRE-0801977), HIDSS4HEALTH and the Cancer Research Center Cologne Essen (CCCE), Germany. Jessica Schmitz and Jan Hinrich Bräsen were supported by German Ministry for Education and Research, Germany (BMBF 13GW0399B) and Jessica Schmitz, Jan Hinrich Bräsen, Nikolaos Vasileiadis, Philipp Ivanyi and Viktor Grünwald by the Wilhelm Sander Foundation, Germany. We would like to thank E. Christians, M. Taleb-Naghsh and J. Jost for their excellent technical assistance.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compmedimag.2023.102238.

References

Bankhead, P., Loughrey, M.B., Fernández, J.A., Dombrowski, Y., McArt, D.G., Dunne, P.D., McQuaid, S., Gray, R.T., Murray, L.J., Coleman, H.G., et al., 2017. Qupath: Open source software for digital pathology image analysis. Sci. Rep. 7 (1), 1–7. http://dx.doi.org/10.1038/s41598-017-17204-5.

Bejnordi, B.E., Veta, M., Van Diest, P.J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J.A., Hermsen, M., Manson, Q.F., Balkenhol, M., et al., 2017. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. JAMA 318 (22), 2199–2210. http://dx.doi.org/10.1001/jama.2017.14585.

Cai, Q., Christie, A., Rajaram, S., Zhou, Q., Araj, E., Chintalapati, S., Cadeddu, J., Margulis, V., Pedrosa, I., Rakheja, D., et al., 2020. Ontological analyses reveal clinically-significant clear cell renal cell carcinoma subtypes with convergent evolutionary trajectories into an aggressive type. EBioMedicine 51, 102526. http://dx.doi.org/10.1016/j.ebiom.2019.10.052.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y., 2021. Transunet: Transformers make strong encoders for medical image segmentation. http://dx.doi.org/10.48550/arXiv.2102.04306, arXiv preprint arXiv:2102.04306.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. http://dx.doi.org/10.48550/arXiv.1706.05587, arXiv preprint arXiv:1706.05587.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. http://dx.doi.org/10.48550/arXiv.2010.11929, arXiv preprint arXiv:2010.11929.

Fan, T., Wang, G., Li, Y., Wang, H., 2020. Ma-net: A multi-scale attention network for liver and tumor segmentation. IEEE Access 8, 179656–179665. http://dx.doi.org/10.1109/ACCESS.2020.3025372.

Goode, A., Gilbert, B., Harkes, J., Jukic, D., Satyanarayanan, M., 2013. OpenSlide: A vendor-neutral software foundation for digital pathology. J. Pathol. Inform. 4, http://dx.doi.org/10.4103/2153-3539.119005.

Graham, S., Vu, Q.D., Jahanifar, M., Raza, S.E.A., Minhas, F., Snead, D., Rajpoot, N., 2023. One model is all you need: Multi-task learning enables simultaneous histology image segmentation and classification. Med. Image Anal. 83, 102685. http://dx.doi.org/10.1016/j.media.2022.102685.

Graham, S., Vu, Q.D., Raza, S.E.A., Azam, A., Tsang, Y.W., Kwak, J.T., Rajpoot, N., 2019. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Med. Image Anal. 58, 101563. http://dx.doi.org/10.1016/j.media.2019.101563.

Gu, F., Burlutskiy, N., Andersson, M., Wilén, L.K., 2018. Multi-resolution networks for semantic segmentation in whole slide images. In: Computational Pathology and Ophthalmic Medical Image Analysis. Springer, pp. 11–18. http://dx.doi.org/10.1007/978-3-030-00949-6_2.

Guo, M.-H., Liu, Z.-N., Mu, T.-J., Hu, S.-M., 2021. Beyond self-attention: External attention using two linear layers for visual tasks. http://dx.doi.org/10.48550/arXiv.2105.02358, arXiv preprint arXiv:2105.02358.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Ilyas, T., Mannan, Z.I., Khan, A., Azam, S., Kim, H., Boer, F.D., 2022. TSFD-net: Tissue specific feature distillation network for nuclei segmentation and classification. Neural Netw. 151, 1–15. http://dx.doi.org/10.1016/j.neunet.2022.02.020.

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

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 1175–1183. http://dx.doi.org/10.1109/CVPRW.2017.156.

Jin, C., Tanno, R., Xu, M., Mertzanidou, T., Alexander, D.C., 2020. Foveation for segmentation of mega-pixel histology images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 561–571. http://dx.doi.org/10.1007/978-3-030-59722-1_54.

Junttila, M.R., De Sauvage, F.J., 2013. Influence of tumour micro-environment heterogeneity on therapeutic response. Nature 501 (7467), 346–354. http://dx.doi.org/10.1038/nature12626.

Kim, Y.J., Jang, H., Lee, K., Park, S., Min, S.-G., Hong, C., Park, J.H., Lee, K., Kim, J., Hong, W., Jung, H., Liu, Y., Rajkumar, H., Khened, M., Krishnamurthi, G., Yang, S., Wang, X., Han, C.H., Kwak, J.T., Ma, J., Tang, Z., Marami, B., Zeineh, J., Zhao, Z., Heng, P.-A., Schmitz, R., Madesta, F., Rösch, T., Werner, R., Tian, J., Puybareau, E., Bovio, M., Zhang, X., Zhu, Y., Chun, S.Y., Jeong, W.-K., Park, P., Choi, J., 2021. PAIP 2019: Liver cancer segmentation challenge. Med. Image Anal. 67, 101854. http://dx.doi.org/10.1016/j.media.2020.101854.

Kiran, I., Raza, B., Ijaz, A., Khan, M.A., 2022. DenseRes-Unet: Segmentation of overlapped/clustered nuclei from multi organ histopathology images. Comput. Biol. Med. 143, 105267. http://dx.doi.org/10.1016/j.compbiomed.2022.105267.

Li, J., Sarma, K.V., Ho, K.C., Gertych, A., Knudsen, B.S., Arnold, C.W., 2017. A multi-scale u-net for semantic segmentation of histological images from radical prostatectomies. In: AMIA Annual Symposium Proceedings, Vol. 2017. American Medical Informatics Association, p. 1140.

Li, S., Sui, X., Luo, X., Xu, X., Liu, Y., Goh, R., 2021. Medical image segmentation using squeeze-and-expansion transformers. http://dx.doi.org/10.48550/arXiv.2105.09511, arXiv preprint arXiv:2105.09511.

Li, Y., Wu, J., Wu, Q., 2019. Classification of breast cancer histology images using multi-size and discriminative patches based on deep learning. IEEE Access 7, 21400–21408. http://dx.doi.org/10.1109/ACCESS.2019.2898044.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3431–3440. http://dx.doi.org/10.1109/CVPR.2015.7298965.

Macenko, M., Niethammer, M., Marron, J.S., Borland, D., Woosley, J.T., Guan, X., Schmitt, C., Thomas, N.E., 2009. A method for normalizing histology slides for quantitative analysis. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE, pp. 1107–1110. http://dx.doi.org/10.1109/ISBI.2009.5193250.

Mehta, S., Mercan, E., Bartlett, J., Weaver, D., Elmore, J.G., Shapiro, L., 2018. Y-net: joint segmentation and classification for diagnosis of breast biopsy images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 893–901. http://dx.doi.org/10.1007/978-3-030-00934-2_99.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J., 2019. Stand-alone self-attention in vision models. Adv. Neural Inf. Process. Syst. 32.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Schmitz, R., Madesta, F., Nielsen, M., Krause, J., Steurer, S., Werner, R., Rösch, T., 2021. Multi-scale fully convolutional neural networks for histopathology image segmentation: from nuclear aberrations to the global tissue architecture. Med. Image Anal. 70, 101996. http://dx.doi.org/10.1016/j.media.2021.101996.

Strudel, R., Garcia, R., Laptev, I., Schmid, C., 2021. Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7262–7272. http://dx.doi.org/10.1109/ICCV48922.2021.00717.

Tokunaga, H., Teramoto, Y., Yoshizawa, A., Bise, R., 2019. Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12597–12606. http://dx.doi.org/10.1109/CVPR.2019.01288.

Van Rijthoven, M., Balkenhol, M., Siliņa, K., Van Der Laak, J., Ciompi, F., 2021. HookNet: Multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images. Med. Image Anal. 68, 101890. http://dx.doi.org/10.1016/j.media.2020.101890.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, H., Xie, S., Lin, L., Iwamoto, Y., Han, X.-H., Chen, Y.-W., Tong, R., 2021. Mixed transformer U-net for medical image segmentation. http://dx.doi.org/10.48550/arXiv.2111.04734, arXiv preprint arXiv:2111.04734.

Wang, S., Yang, D.M., Rong, R., Zhan, X., Xiao, G., 2019. Pathology image analysis using segmentation deep learning algorithms. Am. J. Pathol. 189 (9), 1686–1698. http://dx.doi.org/10.1016/j.ajpath.2019.05.007.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. Adv. Neural Inf. Process. Syst. 34.

Yakubovskiy, P., 2020. Segmentation models pytorch. URL: https://github.com/qubvel/segmentation_models.pytorch.

Zaffar, I., Jaume, G., Rajpoot, N., Mahmood, F., 2022. Embedding space augmentation for weakly supervised learning in whole-slide images. http://dx.doi.org/10.48550/arXiv.2210.17013, arXiv preprint arXiv:2210.17013.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2881–2890. http://dx.doi.org/10.1109/CVPR.2017.660.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al., 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890. http://dx.doi.org/10.1109/CVPR46437.2021.00681.