

# Cell2Grid: an efficient, spatial, and convolutional neural network-ready representation of cell segmentation data

Laurin Herbsthofe<sup>a,b,\*</sup>, Martina Tomberger,<sup>a</sup> Maria A. Smolle,<sup>c</sup>  
Barbara Prietl<sup>a,b,d</sup>, Thomas R. Pieber<sup>a,b,d,e</sup> and Pablo López-García<sup>a</sup>

<sup>a</sup>CBmed, Center for Biomarker Research in Medicine GmbH, Graz, Austria

<sup>b</sup>BioTechMed, Graz, Austria

<sup>c</sup>Medical University of Graz, Department of Orthopaedics and Trauma, Graz, Austria

<sup>d</sup>Medical University of Graz, Division of Endocrinology and Diabetology, Graz, Austria

<sup>e</sup>Health Institute for Biomedicine and Health Sciences, Joanneum Research  
Forschungsgesellschaft mbH, Graz, Austria

## Abstract

**Purpose:** Cell segmentation algorithms are commonly used to analyze large histologic images as they facilitate interpretation, but on the other hand they complicate hypothesis-free spatial analysis. Therefore, many applications train convolutional neural networks (CNNs) on high-resolution images that resolve individual cells instead, but their practical application is severely limited by computational resources. In this work, we propose and investigate an alternative spatial data representation based on cell segmentation data for direct training of CNNs.

**Approach:** We introduce and analyze the properties of Cell2Grid, an algorithm that generates compact images from cell segmentation data by placing individual cells into a low-resolution grid and resolves possible cell conflicts. For evaluation, we present a case study on colorectal cancer relapse prediction using fluorescent multiplex immunohistochemistry images.

**Results:** We could generate Cell2Grid images at 5- $\mu\text{m}$  resolution that were 100 times smaller than the original ones. Cell features, such as phenotype counts and nearest-neighbor cell distances, remain similar to those of original cell segmentation tables ( $p < 0.0001$ ). These images could be directly fed to a CNN for predicting colon cancer relapse. Our experiments showed that test set error rate was reduced by 25% compared with CNNs trained on images rescaled to 5 $\mu\text{m}$  with bilinear interpolation. Compared with images at 1- $\mu\text{m}$  resolution (bilinear rescaling), our method reduced CNN training time by 85%.

**Conclusions:** Cell2Grid is an efficient spatial data representation algorithm that enables the use of conventional CNNs on cell segmentation data. Its cell-based representation additionally opens a door for simplified model interpretation and synthetic image generation.

© The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JMI.9.6.067501](https://doi.org/10.1117/1.JMI.9.6.067501)]

**Keywords:** medical imaging; image processing; neural networks; image segmentation; image compression; image storage.

Paper 22084GRR received Apr. 11, 2022; accepted for publication Nov. 3, 2022; published online Nov. 30, 2022.

## 1 Introduction

Convolutional neural networks (CNNs) have revolutionized image analysis and are considered the *de facto* standard for image classification tasks.<sup>1-5</sup> Their main strength is the ability to learn complex spatial patterns directly from labeled examples without the need for prior knowledge or manual feature extraction. The practical application of CNNs, however, depends on several factors, the size of input images being one of the most limiting. As an example, ImageNet,

\*Address all correspondence to Laurin Herbsthofe, [laurin.herbsthofe@cbmed.at](mailto:laurin.herbsthofe@cbmed.at)

which is a widespread academic resource for annotated image data,<sup>6</sup> contains images that are less than a megapixel in size on average and are even further cropped and resized to  $256 \times 256$  before training CNNs. In contrast, medical images, especially whole-slide imaging (WSI) of tissue sections, can be many orders of magnitude larger, ranging from multimegapixel image tiles or regions-of-interest (ROI) to gigapixels for WSI,<sup>7</sup> making their direct analysis using CNNs unfeasible due to computational constraints.

Even though conventional image rescaling is frequently used in other imaging domains,<sup>1</sup> it is rarely applied to histologic images because small details of individual cells might be lost, eliminating the advantages of microscope magnification. Therefore, most biomedical applications using CNNs on histologic images derived from hematoxylin and eosin (H&E)<sup>8–10</sup> and fluorescent multiplex immunohistochemistry (fm-IHC)<sup>11–17</sup> have relied on tiling or patching<sup>18–20</sup> as the preferred image preprocessing step. However, extensive image tiling leads to new challenges, including loss of large-scale image context, weakly supervised labels, and multiinstance learning settings. Researchers are actively proposing new solutions to these problems,<sup>21–23</sup> but they are still not general enough to be applied systematically.

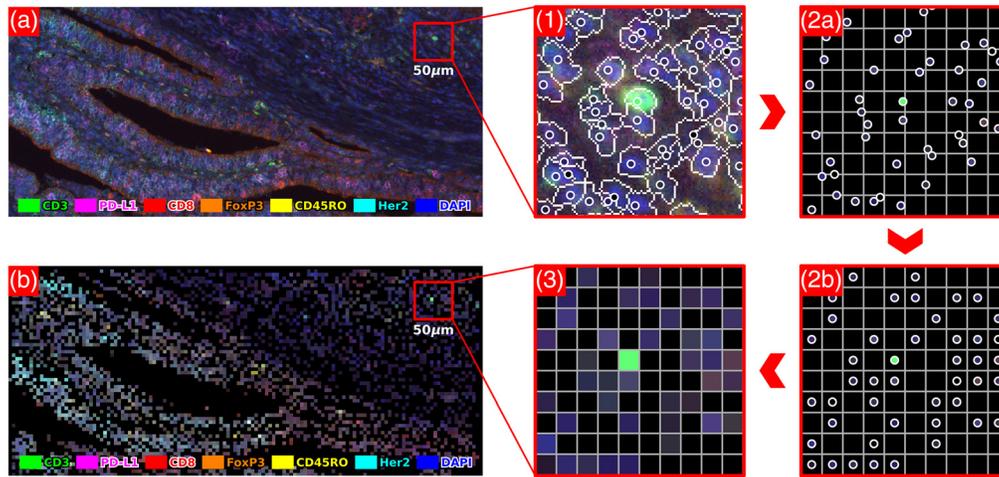
An alternative and more classical approach to using histologic images is to focus on individual cells instead.<sup>15,16,24–27</sup> Cell segmentation algorithms take a histologic image as input and generate a list of identified cells with their coordinates in the image and other features (e.g., cell size, shape, texture, antibody marker expression levels and others). Many cell segmentation algorithms are already implemented in open source and commercial software (e.g., CellProfiler,<sup>28</sup> ImageJ,<sup>29</sup> and inForm (Akoya Biosciences, Inc.), or HALO software (Indica Labs)) and new ones are published every year.<sup>30,31</sup> Cell segmentation allows one to perform more traditional hypothesis-driven spatial analysis, including cell phenotype quantification in different tissue regions, nearest neighbor analysis, identification of touching cells and cell clusters, and phenotype colocalization.<sup>32–35</sup> In fact, several biomarkers have been identified this way, indicating the importance of cell segmentation in tissue imaging.<sup>36–39</sup> However, in contrast to feeding histologic images to CNNs directly, this kind of analysis requires a substantial amount of a priori domain knowledge to extract meaningful spatial features.

Most importantly, some biological questions still remain elusive to such hypothesis-driven approaches, as is the case of colorectal cancer relapse and survival prediction.<sup>40–44</sup> In such cases, machine learning methods, such as random forests, support vector machines, or neural networks could be helpful, but these algorithms cannot directly consume the output of cell segmentation algorithms (one table per sample). To overcome this limitation, graph neural networks (GNNs),<sup>45,46</sup> which first convert the list of identified cells into a so-called cell graph that represents cell–cell interactions in the data,<sup>47</sup> have been proposed. Recent research explored the design space for GNNs for graph classification, highlighting the complexity of choosing an appropriate GNN for any given task.<sup>48</sup> Additionally, several hyperparameters for the construction of cell graphs (usually the parameters for  $k$ -nearest neighbor connections of cells and a maximum edge length cutoff<sup>46</sup>) require careful tuning. The identification of best-practices for constructing cell graphs and the selection of a suitable GNN architecture therefore remain active fields of research.<sup>48–50</sup> Compared with these graph-based methods, conventional pixel-based CNNs are a mature technology that is well established in biomedical tissue analysis.<sup>51,52</sup>

In this paper, we introduce Cell2Grid, an algorithm for efficient representation of histologic images that transform cell segmentation data into low-resolution images suitable for training conventional CNNs. To the best of our knowledge, this is a novel approach that has not been explored in depth. For evaluation, we present a case study on colorectal cancer relapse prediction where we show that Cell2Grid images maintain cell spatial information while providing better performance compared with conventional image rescaling when training CNNs.

## 2 Methods

Cell2Grid transforms the table-style output of cell segmentation methods into spatial cell-based data representations (referred to as “Cell2Grid images”) that are orders of magnitudes smaller than original input images. The entire image-to-image pipeline consists of three steps (see Fig. 1): (1) cell segmentation and cell feature extraction, (2) assignment of cells to a target grid,



**Fig. 1** Visualization of the image-to-image pipeline of Cell2Grid. (a) Input fm-IHC image ( $0.5 \mu\text{m}/\text{px}$ ) with six fluorescence color channels + DAPI, shown as false-color RGB image; (1) cell segmentation with cell nuclei as circles; (2a) location of cells relative to  $5 \mu\text{m}$  target grid; (2b) one-to-one assignment of cells to target grid; (3) Cell2Grid data representation; and (b) final Cell2Grid image ( $5 \mu\text{m}/\text{px}$ , 100 times smaller compared with original data size), shown as false-color RGB image (best viewed on screen).

and (3) image creation. Step 2 is the main contribution of this work and depends on two parameters, such as the target grid spacing  $d$  and the maximum local conflict resolution size  $w_{\max}$ , which controls how cell assignment conflicts are resolved. Even though our method is more closely related to image rescaling rather than image compression, we will use the term compression ratio for the ratio between high-resolution input image and Cell2Grid output image sizes as a result of a change in pixel resolution.

## 2.1 Step 1: Cell Segmentation and Feature Extraction

Individual biological cells are identified in the input image using a cell segmentation method suited for the tissue type, see e.g., Refs. 24, 53, and 54. For each identified cell, the location  $\vec{X} = (x_1, x_2)$  of its nucleus in the image is stored together with  $f$  extracted cell features, e.g., the average marker intensity for each fluorescent color channel, the cells size and shape, etc. The final output of the cell segmentation step is a list of cells with their coordinates  $\vec{X}$  and  $f$  features, as shown in Table 1.

Cell segmentation in an fm-IHC laboratory setting is normally performed using established software, e.g., CellProfiler,<sup>28</sup> ImageJ,<sup>29</sup> inForm (Akoya Biosciences, Inc.), or HALO (Indica Labs), with supervision from an expert in the field, as the process depends on multiple parameters, such as tissue type and tissue stains, and could require manual annotation as training data. This was the setting and scope of our study, but it is worth mentioning that recently published state-of-the-art methods on cell segmentation<sup>30,31</sup> might soon become widespread in such a setting, providing even better results (see Sec. 4.1).

**Table 1** Example output of the cell identification (Step 1).

Cell ID	$x_1$ ( $\mu\text{m}$ )	$x_2$ ( $\mu\text{m}$ )	Feature 1	...	Feature $f$
1	42.0	3.5	233		93
2	65.5	12.5	20		209
...	...	...	...		...

## 2.2 Step 2: Assignment of Cells to Target Grid

After choosing a target grid spacing  $d$ , all identified cells are assigned to the square target grid by binning their original coordinates  $\vec{X}$  to the nodes of the grid using  $\vec{X}_g = \lfloor \vec{X}/d \rfloor$ , whereas  $\lfloor \cdot \rfloor$  denotes conventional rounding to the next integer and  $\vec{X}_g$  corresponds to coordinates of the target grid nodes (GNs) (i.e., indices of pixel locations in the final output image). While for most biological tissue types,  $d = 5 \mu\text{m}$  is a good choice, we provide guidelines for how to systematically choose the target grid spacing  $d$  in [Supplementary Material S.3](#).

Using this method, any two cells within a distance of  $d\sqrt{2}$  may be assigned to the same GN  $\vec{X}_g$ . To achieve a one-to-one relation of cells to output pixels, we require that each cell be assigned uniquely to a GN and that any GN can hold at most one biological cell. This one-to-one relation is the essential property of Cell2Grid data representation.

Assignment conflicts are resolved successively by applying Munkres' algorithm<sup>55</sup> to all cells within the local  $3 \times 3$  subgrid around a GN with conflicts. If the number of cells  $k$  within this window is higher than the number of GNs in the subgrid, the  $5 \times 5$  subgrid is considered. This process is continued until a maximum window size  $w_{\max} \times w_{\max}$  is reached. If the number of cells still exceeds the number of available GNs,  $k > (w_{\max})^2$ , then  $k - (w_{\max})^2$  cells are deleted from the data. This process is explained in more detail in [Supplementary Material S.1](#).

After conflict resolution, every cell has either been uniquely assigned to a GN or was deleted due to unresolvable conflicts. Similarly, every target GN either holds exactly one or no cells.

### 2.2.1 Grid spacing and assignment conflicts – theoretical analysis

In this section, we present analytical expressions for the expected number of GNs with assignment conflicts and the total expected cell loss after local conflict resolution. These expressions assume a random uniform distribution of cell locations over the entire image and depend on the overall cell density  $\rho$ , the target grid spacing  $d$ , and the local conflict resolution window size  $w$ .

Using results from the ball-into-bin model that describes the occupancy problem,<sup>56</sup> the random allocation of  $n$  cells into  $m$  nodes is described by the Poisson distribution  $\text{Pois}(\lambda, X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$ , which is the probability mass function for a single bin (GN) containing  $k$  balls (cells) with  $\lambda = \frac{n_{\text{cells}}}{m_{\text{nodes}}}$ , and  $m_{\text{nodes}}$  being the number of GNs of the target grid. Therefore,  $\text{Pois}(\lambda, X \geq 2)$  estimates the fraction of GNs that hold two or more cells (i.e., assignment conflicts) and is only dependent on  $\lambda$ . For our purpose,  $\lambda$  can be defined using the cell density  $\rho = \frac{n_{\text{cells}}}{A}$  of  $n_{\text{cells}}$  in an image area  $A$  and target grid spacing  $d$  as  $\lambda(d) = \frac{n_{\text{cells}}}{m_{\text{nodes}}} = \frac{n_{\text{cells}}}{(A/d^2)} = \rho d^2$ . Therefore, the estimated fraction of GNs with  $k$  assigned cells is written as

$$\text{Pois}(\lambda(d), k) = \frac{(\rho d^2)^k}{k!} e^{-\rho d^2}. \quad (1)$$

Using this result, we express the fraction of GNs that hold two or more cells (i.e., nodes with assignment conflicts) as follows:

$$\text{GN}_{\text{conf}}(d) = \text{Pois}(\lambda(d), k \geq 2) = 1 - \text{Pois}(\lambda(d), k = 0) - \text{Pois}(\lambda(d), k = 1). \quad (2)$$

The fraction of cells in conflict is the number of cells that are not allocated to one-cell GNs,

$$\text{C}_{\text{conf}}(d) = 1 - \frac{m_{\text{nodes}}}{n_{\text{cells}}} \text{Pois}(\lambda(d), k = 1). \quad (3)$$

Next, we estimate the number of unresolvable conflicts when a local conflict resolution window of size  $w$  is used ( $w$  being an odd integer  $\geq 3$ ). Using a modified target grid spacing  $d_w = w \cdot d$ , we can estimate the number of cells in a  $w \times w$  subgrid around a GN with conflicts. Subgrids (or “neighborhoods”) that contain more than  $w^2$  cells constitute unresolvable conflicts. Because taking the local neighborhood into account is only relevant when the central node actually contains a conflict, we multiply with  $\text{GN}_{\text{conf}}(d)$ . By neglecting conditional probabilities for  $k > w^2$  when there are already  $k \geq 2$  cells at the central node, we arrive at

$$\begin{aligned} \text{GN}_{\text{unres}}(d, w) &\approx \text{GN}_{\text{conf}}(d, w) \cdot \text{Pois}(\lambda(wd), k > w^2) \\ &= \text{GN}_{\text{conf}}(d, w) \cdot \left(1 - \sum_{k=0}^{k=w^2} \text{Pois}(\lambda(wd), k)\right). \end{aligned} \quad (4)$$

Finally, we estimate the cell-loss-fraction due to unresolvable conflicts by considering cells that have been assigned successfully using  $C_{\text{loss}} = (n_{\text{cells}} - N_1 - N_2)/n_{\text{cells}}$ . Here,  $N_1$  is the number of cells located in “unsaturated” local neighborhoods (all cells are assigned if an area of size  $(wd)^2$  contains  $k \leq w^2$  cells):

$$N_1 = m_{\text{nodes}} \sum_{k=0}^{w^2} \frac{k}{w^2} \text{Pois}(\lambda(wd), k). \quad (5)$$

Note that the summation term contains  $k/w^2$ , which is the average number of cells per GN when  $k$  cells are in an area of  $w^2$  GNs. In contrast,  $N_2$  is the number of cells that are successfully assigned during conflict resolution in “saturated” areas (in areas of size  $(wd)^2$  containing  $k > w^2$  cells only  $w^2$  cells are assigned and  $k - w^2$  cells need to be deleted):

$$N_2 = m_{\text{nodes}} \cdot \text{Pois}(\lambda(wd), k > w^2) = m_{\text{nodes}} \left(1 - \sum_{k=0}^{w^2} \text{Pois}(\lambda(wd), k)\right). \quad (6)$$

In  $N_2$ , we simply count the number of GNs that are part of oversaturated areas because each of them will contain exactly one cell after conflict resolution. Put together and simplified, the estimated cell loss after conflict resolution is

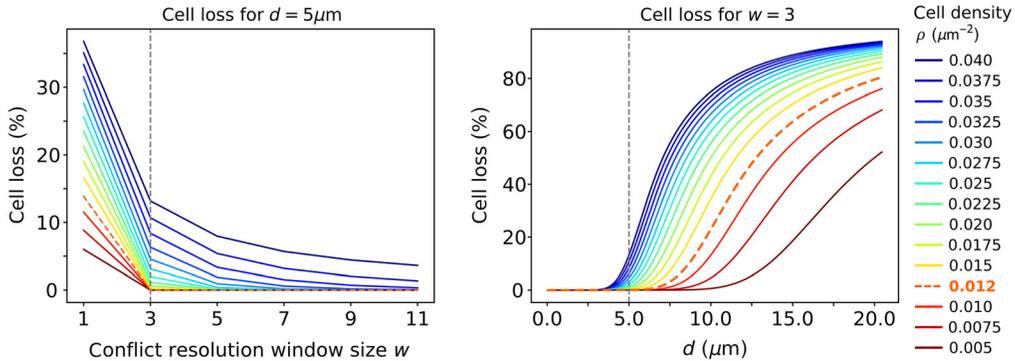
$$C_{\text{loss}}(w, d) = 1 - \frac{w^2}{\lambda(wd)} - \frac{w^2}{\lambda(wd)} \sum_{k=0}^{w^2} \text{Pois}(\lambda(wd), k) \cdot \left(\frac{k}{w^2} - 1\right). \quad (7)$$

In Fig. 2, we visualize  $C_{\text{loss}}(w, d)$  for different values of  $w$ ,  $d$ , and cell densities  $\rho$ .

Note that for comparison with empirical data that use an adaptive local conflict resolution window  $w$  (see [Supplementary Material S.1](#)), using  $w = 3$ , provides the best approximation.

### 2.3 Step 3: Image Creation

In the last step of Cell2Grid, a low-resolution output image is created. Each node of the target grid stores the  $f$  extracted features of its assigned cell, like pixels in conventional RGB images storing three color values. This final data structure is a tensor of size  $K_{x_1} \times K_{x_2} \times f$



**Fig. 2** Expected cell loss for different target grid spacings  $d$ , conflict resolution window sizes  $w$  and cell densities  $\rho$ . Left: fixed target grid spacing  $d = 5 \mu\text{m}$  and vertical dashed line denotes simplest nontrivial conflict resolution size  $w = 3$ . Right: fixed conflict resolution window size  $w = 3$  and vertical dashed line denotes target grid spacing chosen for all presented experiments ( $5 \mu\text{m}$ ). Cell density of our dataset is  $\rho \approx 0.012 \mu\text{m}^{-2}$  (orange dashed line).

where  $K_{x_1}$  and  $K_{x_2}$  are the number of GNs in  $x_1$  and  $x_2$  dimension. The GNs without cells are assigned zero as a default value for every cell feature. Because a square grid was used, this tensor can be converted directly into an image with  $f$  color channels, e.g., using the multipage supporting tagged image file format (TIFF). This multipage image, which we term Cell2Grid image, is the final output of the image creation step and the whole Cell2Grid algorithm (see Figs. 1 and 8 for examples).

### 3 Experiments and Results

To evaluate the utility of Cell2Grid in practice, we selected an inhouse dataset of histologic images from stage II colon cancer patients. In our experiments, we aim to investigate (a) the data processing time and cell loss of Cell2Grid for different target grid spacings, (b) how conventional cell-based features are influenced by the spatial approximation of cell locations, and (c) how CNNs perform on the task of predicting the patients 5-year tumor recurrence from histologic images using our method compared to conventional image rescaling.

#### 3.1 Data for Experiments

Surgically removed tumor tissue samples from  $N = 54$  patients were collected from the local biobank affiliated to the university hospital where patients had been treated (Ethics Committee number: 28-342 ex 15/16). Each patient's tumor recurrence status after 5 years was labeled as either relapse (17 patients) or healthy (37 patients), serving as the variable to be predicted by classification CNNs.

To obtain fm-IHC images (see Fig. 1 for an example), one formalin-fixated paraffin-embedded tumor tissue sample per patient was stained with fluorescence-conjugated antibodies targeting CD3, CD8, CD45RO, PD-L1, FoxP3, Her2, and DAPI. Of each whole-slide scan, several ROIs covering an area of  $674 \times 506 \mu\text{m}$  were selected by a human expert and were recorded with  $20\times$  magnification ( $1344 \times 1008 \text{ px} \sim 0.5 \mu\text{m}/\text{px}$  resolution) using a Vectra 3 microscope (PerkinElmer, Inc.). Spectral unmixing of raw images was performed using inForm software (Akoya Biosciences, Inc.). In total,  $n = 1353$  images ( $\approx 25$  per patient) were recorded.

#### 3.2 Grid Spacing, Assignment Conflicts, and Processing Time

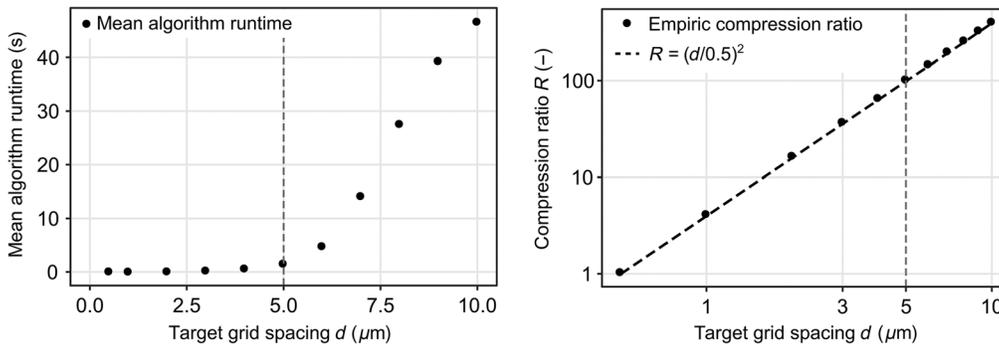
We first investigated the empirical algorithm runtime and compression ratio of Cell2Grid and if real cell loss numbers follow our analytical expressions. Cell segmentation (step 1) was performed using inForm software (Akoya Biosciences, Inc.) with parameters set by an IHC expert. Cell nuclei were identified using DAPI stain, and cell membranes and cytoplasm were subsequently identified using remaining markers. On average, 4131 cells per image were identified (cell density  $\rho = 0.0121 \mu\text{m}^{-2}$ ). For each cell, we extracted  $f = 6$  features using the average marker intensity across the entire cell area of each fluorescent color channel (excluding DAPI).

We then investigated how different target grid spacing values influenced Cell2Grid, using a maximum local conflict resolution size  $w_{\max} = 7$ , see Figs. 3 and 4. Algorithm runtime remained flat below  $d = 5 \mu\text{m}$  and increased steeply afterward. As expected, image compression ratio increased with the square of the target grid spacing  $R = (d [\mu\text{m}]/0.5 \mu\text{m})^2$  with  $0.5 \mu\text{m}$  being the original image resolution.

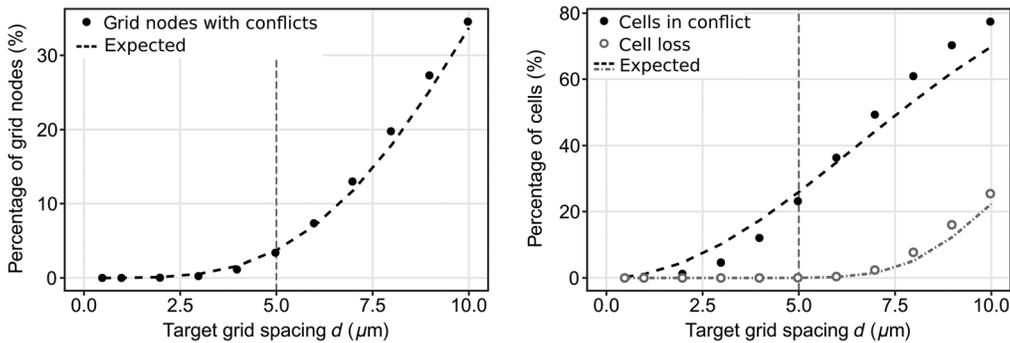
Figure 4 shows that the percentage of GNs with conflicts and cell loss due to unresolvable conflicts followed the expected behavior [Eqs. (2), (3), and (7)]. However, the percentage of cells in conflict before conflict resolution was lower than expected below  $d = 5 \mu\text{m}$  and higher for larger values of  $d$  due to non-uniform distributions of cells in real biological tissue. More importantly, cell loss remained neglectable below  $d = 5 \mu\text{m}$  and increased for higher values.

Based on these findings, we decided to use  $d = 5 \mu\text{m}$  as the target grid spacing of Cell2Grid for our following experiments. Using our dataset, this value provided a compression ratio of  $R = 100$ , average cell loss of 2.06 cells (0.05%) per image, and processing time of 1.56 s per image (excluding cell segmentation).

To put the processing time of our method into perspective, we compared it with three conventional image rescaling methods and with the creation of a cell graph using cell segmentation data.



**Fig. 3** Mean algorithm runtime per image and image compression ratio (double-log plot) for different target grid spacings. Vertical dashed line represents our final choice of  $d = 5 \mu\text{m}$ .

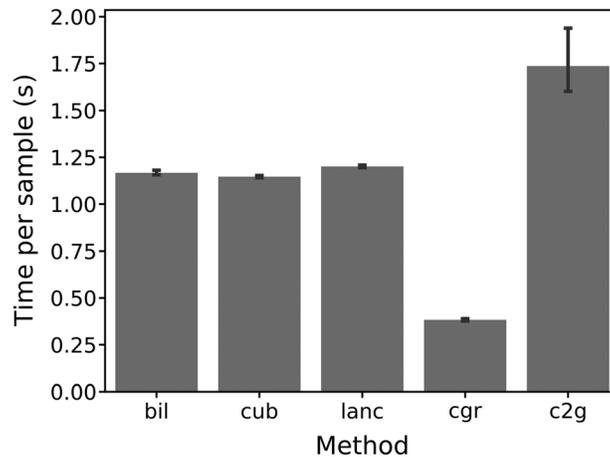


**Fig. 4** Cell assignment properties for different target grid spacings averaged over all images of our dataset. Left: percent of GNs with conflicts before conflict resolution. Right: initial percent of cells in conflict and cell loss after conflict resolution. Vertical dashed lines represent our final choice of  $d = 5 \mu\text{m}$ .

Cell graphs were created using  $k$ -nearest-neighbor algorithm ( $k = 5$ , see Fig. 8(c) for an example). Figure 5 shows the mean data processing time for each method but does not include time for cell segmentation required for cell graph creation and Cell2Grid. The comparably high variance in Cell2Grid is a result of the variance in cell densities and subsequent conflict resolutions per image, the latter being the main driver of computation time in Cell2Grid.

### 3.3 Phenotype Counts and Spatial Features

Next, we investigated if conventional spatial features calculated from cell segmentation output tables were altered due to the local spatial approximations performed in Cell2Grid. For every individual image of our colon cancer relapse dataset, we calculated the total number of  $T$  cells (CD3+) and regulatory  $T$  cells (CD3 + FoxP3 + CD8-) to investigate the potential change in a dense and a rare cell population, respectively. Additionally, we investigated a potential change in the average distance between  $T$  cells and their nearest PD-L1 positive cell as well as the number of  $T$  cells around PD-L1 positive cells within a radius of  $50 \mu\text{m}$  using the phenoptr  $R$  package.<sup>34</sup> Marker thresholds for these phenotypes were set manually. Figure 6 compares these feature values calculated with original cell segmentation coordinates and coordinates as approximated by Cell2Grid. Regression plots (Pearson  $R > 0.99$ ,  $p < 0.0001$ ) showed strong agreement between both methods, which confirms that the small number of deleted cells introduces only insignificant changes in the data. Additionally, Bland-Altman plots, offering an alternative way to test the same hypothesis, show that at least 95% of the method differences fall within the predefined interval of  $\pm 1.96$  times the standard deviation (SD), which is a sign of good agreement between the two measurements.<sup>57</sup> In particular, no more than three  $T$  cells were deleted in any given image. Notably, no regulatory  $T$  cell was deleted in any image, indicating that rare cell populations are less likely to be affected by cell loss simply due to their lower abundance.



**Fig. 5** Mean and 95% confidence interval for data processing times per sample shown for image rescaling with bilinear (bil) and cubic (cub) interpolations and Lanczos (lanc) sampling, cell-graph (cgr) creation and Cell2Grid (c2g) representation, the latter two starting from cell segmentation data and do not include cell segmentation processing time. All presented times include read and write operations, multipage-TIFF-file unpacking and stacking (bil, cub, and lanc), and necessary metadata extraction.

Furthermore, the mean distance between  $T$  cells and their nearest PD-L1 positive cell was altered by  $< \pm 1.6 \mu\text{m}$  for 95% of images, and the count of  $T$  cells within a radius of  $50 \mu\text{m}$  around PD-L1 positive cells was altered by a count of  $< \pm 5$  cells in  $>97\%$  of images. Results for additional features, including cell marker distributions within the cells and nucleus and cell shape properties, are shown in [Supplementary Material S.4](#).

### 3.4 Neural Network Image Classification

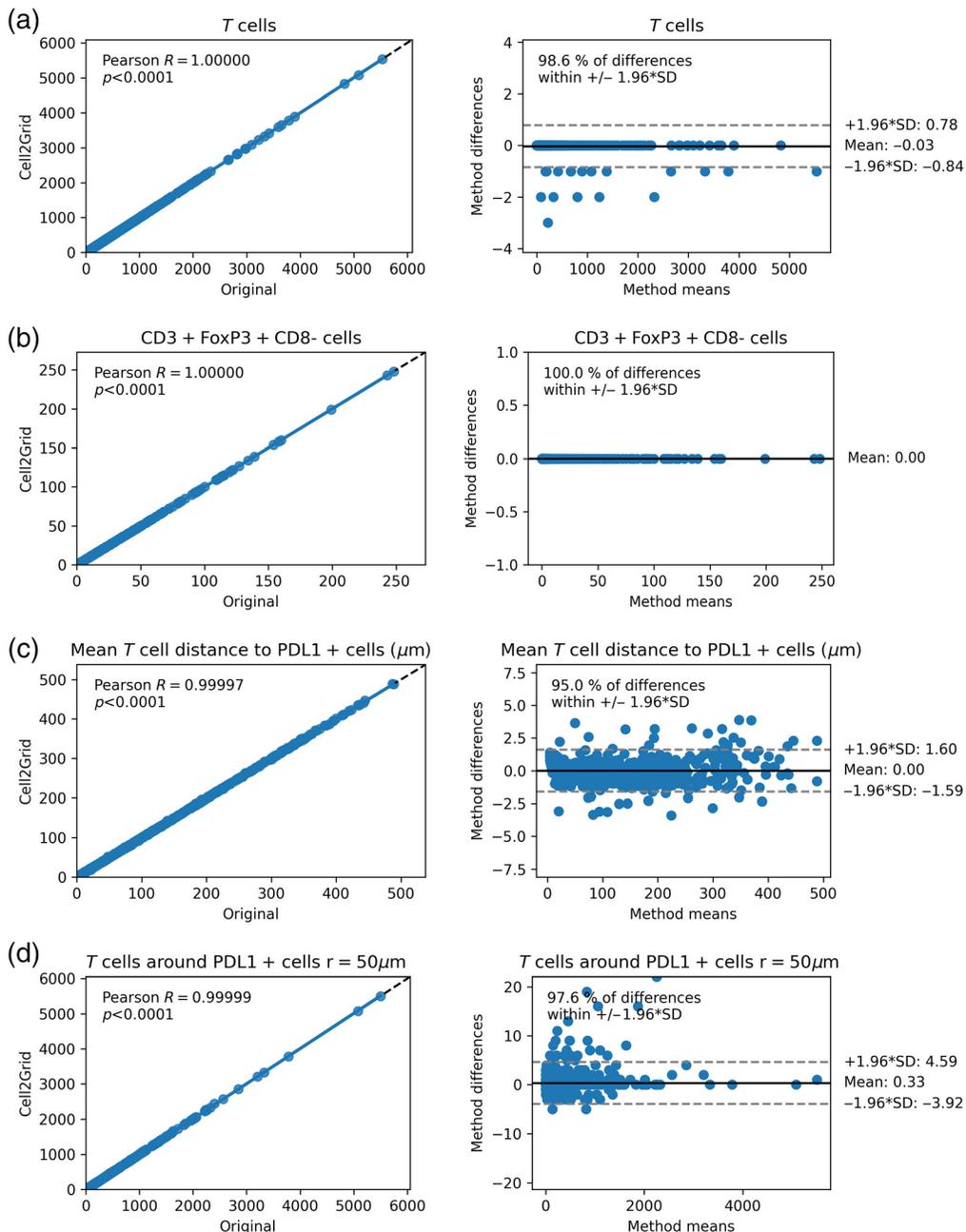
Finally, we investigated how different CNN architectures performed on Cell2Grid images, and if given a fixed image size, Cell2Grid data representation improved performance compared with standard image rescaling. For the retrospective colon cancer dataset introduced in Sec. 3.1, the goal was to predict the patients' 5-year tumor recurrence status (relapse/healthy) from histologic images in a supervised training setting. For that purpose, we trained two different CNN architectures (VGG and small interpretable network (SIN), explained below) on Cell2Grid images and on raw images rescaled with three conventional image rescaling methods (bilinear (bil) interpolation, bicubic (cub) interpolation, and Lanczos (lanc) sampling),<sup>58</sup> all using a final resolution of  $5 \mu\text{m}/\text{px}$ .

To investigate the effects of additional cell shape features in Cell2Grid, we also trained a CNN on Cell2Grid images that contain, apart from the six default mean marker channels, three additional image channels describing the cell shapes (nucleus size, cell size, and cell axis ratio). We refer to these image set as c2gShape. Additionally, we trained a GNN on a cell graph created from the same cell segmentation data used for Cell2Grid.

Figure 7 shows the entire experiment setup, and a side-by-side comparison of all data modalities is shown in Fig. 8. A description of interpolation-free data augmentation schemes required for Cell2Grid images is presented in [Supplementary Material S.5](#).

#### 3.4.1 Modified VGG-16 architecture

As a baseline for CNN image classification, we chose the well-established VGG-16 architecture,<sup>3</sup> which, despite its age and simplicity, has been used frequently with biological tissue images in recent research.<sup>59-64</sup> To suit the six-color-channel image data format of our images, we modified the architecture using a different first convolutional layer. Fully connected layers at the top of the model varied in number of neurons compared with the original VGG-16 due to the change in input image size and output dimensions. We simply refer to this modified architecture as VGG in the following. This model has  $>5.6 \cdot 10^7$  parameters, all of which were initialized

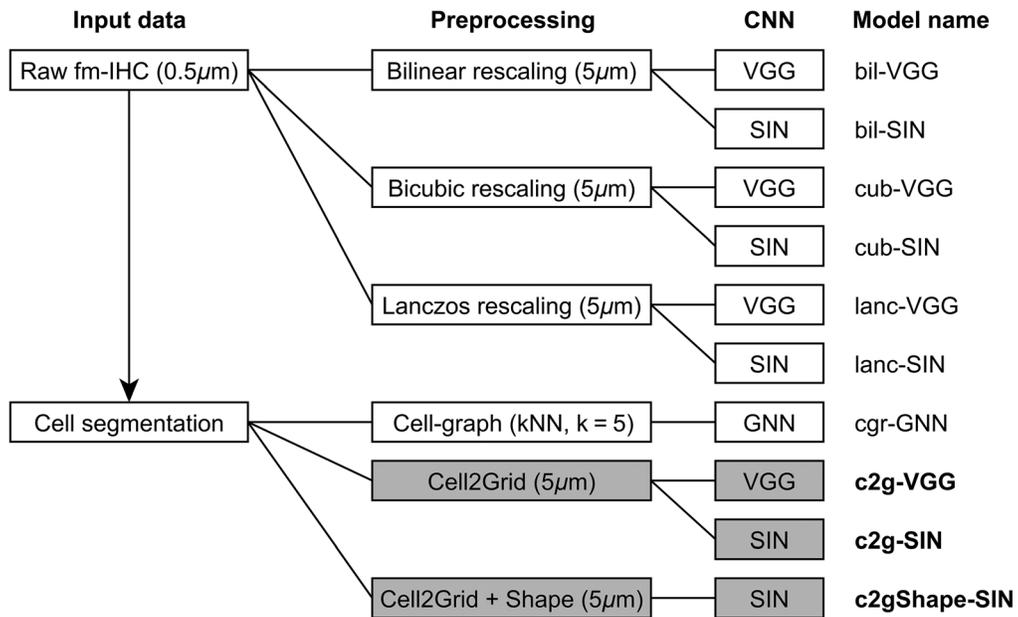


**Fig. 6** Comparison of conventional spatial features calculated using original cell segmentation coordinates versus Cell2Grid coordinates using Pearson correlation (left column, showing correlation coefficient and  $p$ -value) and the same data visualized using Bland–Altman plots (right column, showing percentage of measurements lying within an interval of  $\pm 1.96$  times the SD around the mean). Top row to bottom: (a) number of T cells per image, (b) number of CD3 + FoxP3 + CD8 cells (regulatory T cells) per image (no change, SD = 0), (c) mean distance between T cells and nearest PD-L1 positive cell, and (d) count of T cells within  $50\text{-}\mu\text{m}$  radius around PD-L1 positive cells.

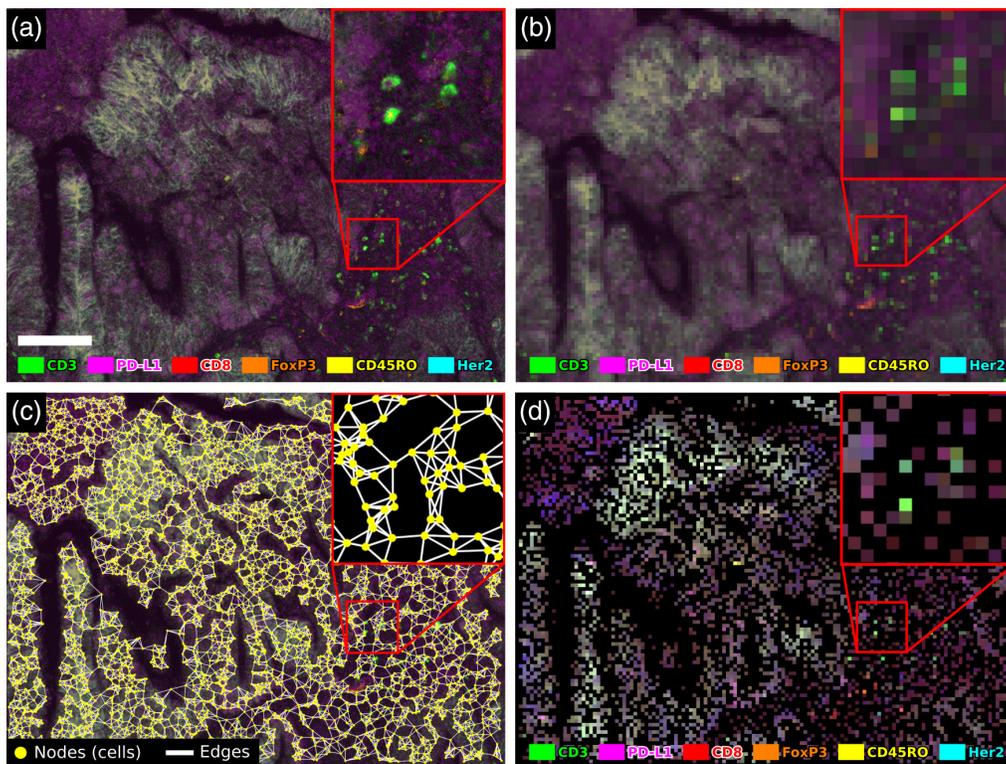
randomly. Results for model runs with network weights pretrained on the ImageNet dataset<sup>6,65,66</sup> are shown in [Supplementary Material S.6](#).

### 3.4.2 Custom network architecture SIN

As a second model, we use a custom CNN architecture in the style of VGG-16 that uses fewer layers and parameters. We refer to this architecture as SIN in the following. For the first layer of



**Fig. 7** Model configurations used in our experiment. Models using our method highlighted in gray.



**Fig. 8** Comparison of image representation methods using false color images of six marker channels. Scale bar 100 µm. (a) Original fm-IHC (0.5 µm resolution), (b) conventional bil rescaling (5 µm), (c) cell segmentation output (yellow dots) and cgr (white lines), and (d) Cell2Grid image (5 µm target grid) using the same cell segmentation data as shown in panel (c). Image best viewed on screen.

**Table 2** SIN architecture.

Layer type	Output Shape	Params #
InputLayer	(102, 135, and 6)	0
Conv2D	(102, 135, and 16)	96
Conv2D	(102, 135, and 16)	2304
MaxPooling2D	(34, 45, and 16)	0
Conv2D	(34, 45, and 16)	2304
MaxPooling2D	(11, 15, and 16)	0
Conv2D	(11, 15, and 16)	2304
MaxPooling2D	(3, 5, and 16)	0
Conv2D	(3, 5, and 16)	2304
MaxPooling2D	(1, 1, and 16)	0
Flatten	(16)	0
Dense	(16)	272
Dropout	(16)	0
Dense	(2)	34
Total params: 9618		
Trainable params: 9618		
Nontrainable params: 0		

SIN, we used a convolution with a  $1 \times 1$  kernel inspired by the network-in-network approach of Lin et al.<sup>67</sup> The weights of this layer were subjected to L1-regularization with regularization factor  $\lambda = 10^{-4}$ . This was followed by 4 pairs of  $3 \times 3$  convolutions with 16 feature maps each and subsequent max pooling layers. On top of the network, we placed 1 fully connected layer with 16 nodes, followed by a dropout layer (33 %) and a final SoftMax layer with 2 outputs. This architecture uses  $<10^4$  trainable parameters and has at most 16 feature maps per layer. No pretraining was applied to this model. Table 2 SIN architecture summarizes the SIN architecture.

### 3.4.3 Graph neural network

As a reference, we compared all methods with GNN trained on cell graphs using the same cell segmentation data that were used for Cell2Grid. We created cell graphs from cell segmentation data using  $k$ -nearest neighbors ( $k = 5$ ) and an edge cutoff length of  $25 \mu\text{m}$  following Jaume et al.<sup>46</sup> (see Fig. 8(c) for an example). Each node was described with the same cell features as in Cell2Grid data. We used the best GNN architecture for graph classification as identified by You et al.<sup>48</sup> implemented using the Spektral Python package.<sup>68</sup>

### 3.4.4 Training setup

From the total of  $n = 1353$  recorded images, we used the same hold-out test set of 247 images (176 healthy and 71 relapse) for all models. The remaining 1106 images were randomly split into a training and validation set (70 %/30 %) stratified by class label using the same split for each model. Oversampling of the minority class was used during training to account for class imbalance. We used a class weight of 3:1 in favor of the relapse class to emphasize the clinical

importance of predicting relapse cases. Early stopping was used to end training when validation set error rate stopped decreasing. We used a batch size of 64 and binary cross-entropy as loss function and the Adam optimizer.<sup>69</sup> Learning rate and layer initialization of VGG were subjected to hyperparameter optimization (see [Supplementary Material S.6](#)).

All models were trained on an off-the-shelf desktop workstation with a single GPU (Nvidia RTX 2070). Models were implemented in Keras<sup>65</sup> with a TensorFlow backend.<sup>70</sup> We trained each model ten times and evaluated the prediction performance of each run on the validation and test set using the balanced error rate, a metric equivalent to the mean of the false-positive and false-negative rate of model predictions.

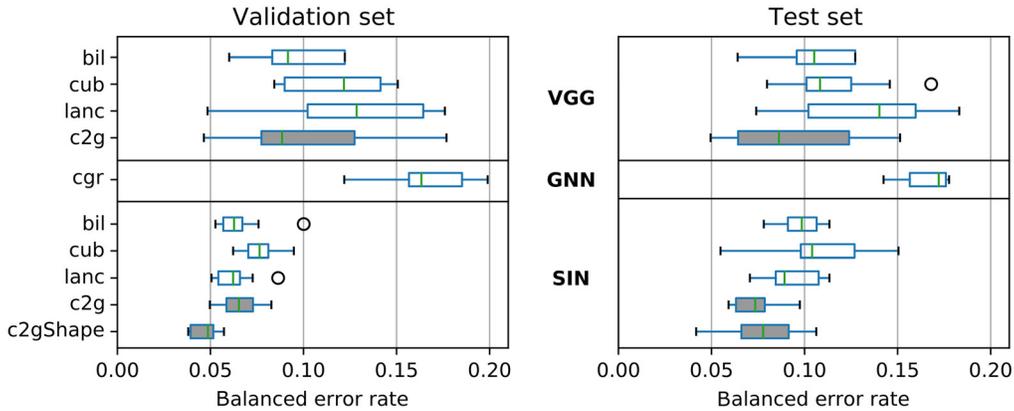
### 3.4.5 Results

To assess model performance, we report the balanced classification error rate  $\text{err}_{\text{bal}} = 1 - \text{acc}_{\text{bal}}$  calculated from the balanced image classification accuracy  $\text{acc}_{\text{bal}}$  of model predictions compared with the known ground truth label (5-year tumor recurrence status). Results of all models and ten repeated runs each are summarized in [Table 3](#) and [Fig. 9](#). VGG models showed higher variance in error rates compared with SIN, regardless of image processing method. As shown in [Supplementary Material S.6](#), different training settings for VGG did not improve its performance. We found that conventional rescaling methods bil and cub resulted in models with comparable performance on the test set with lanc falling slightly behind. Despite similar validation set error rates, Cell2Grid models maintained lower test set error rates compared with bil, cub, and lanc models, especially for the SIN architecture, indicating better generalization properties and less overfitting. [Figure 10](#) shows a comparison of the training curves of the Cell2Grid-SIN and bil-SIN models and test set receiver operating characteristic (ROC) curves for all ten repeated model runs.

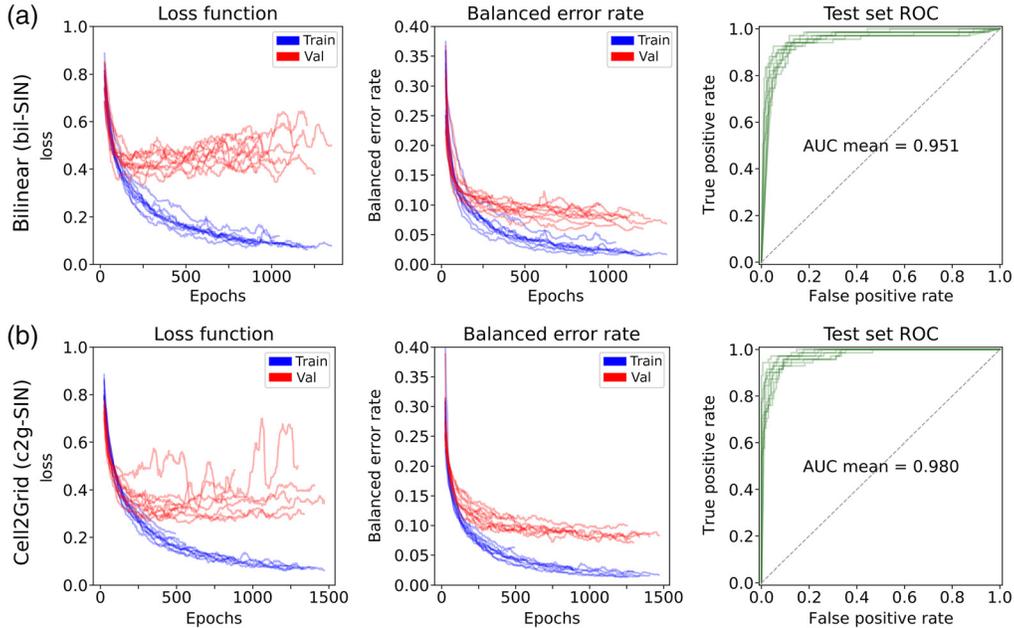
Additional cell shape features, as used in c2gShape-SIN, led to the best validation set error rates, but did not improve upon Cell2Grid-SIN (which use only fluorescent marker features) on the test set across multiple model runs. We attribute this to c2gShape containing more information but also being more prone to overfitting due to its additional input features. Nevertheless, the single best performing model run was a c2gShape-SIN model (3.8% and 4.2% balanced error rate on the validation and test set, respectively). The GNN model used in our experiment showed higher test set error rates than all CNN models. As shown in [Table 3](#) and [Fig. 9](#), models trained

**Table 3** Balanced error rates for all models and ten repeated runs. See [Fig. 9](#) for a visual representation.

Model	Balanced error rate median (min–max)		Relative test error rate compared with bil model (lower is better)	Relative improvement compared with bil model
	Validation set	Test set		
bil-VGG	0.092(0.060–0.500)	0.105(0.064–0.500)	1.000	—
cub-VGG	0.121(0.084–0.151)	0.108(0.080–0.168)	1.028	—
lanc-VGG	0.140(0.102–0.176)	0.152(0.102–0.183)	1.448	—
Cell2Grid-VGG	0.088(0.046–0.296)	0.086(0.050–0.297)	0.819	18.1%
cgr-GNN	0.163(0.122–0.200)	0.172(0.143–0.178)	—	—
bil-SIN	0.063(0.053–0.100)	0.099(0.078–0.113)	1.000	—
cub-SIN	0.076(0.062–0.095)	0.104(0.055–0.151)	1.051	—
lanc-SIN	0.062(0.050–0.086)	0.089(0.071–0.113)	0.899	10.1%
Cell2Grid-SIN	0.065(0.050–0.083)	0.074(0.059–0.097)	0.748	25.3%
c2gShape-SIN	0.049(0.038–0.057)	0.078(0.042–0.106)	0.788	21.2%



**Fig. 9** Balanced validation and test set error rates for all models (lower is better), ten repeated runs summarized by a single boxplot each (two outliers for bil-VGG and one for Cell2Grid-VGG with error rates above 0.2 not shown for clarity, see Table 3). Datasets indicated on the left and network architectures in the middle in bold. Models trained with our method (Cell2Grid and c2gShape) are highlighted in gray.

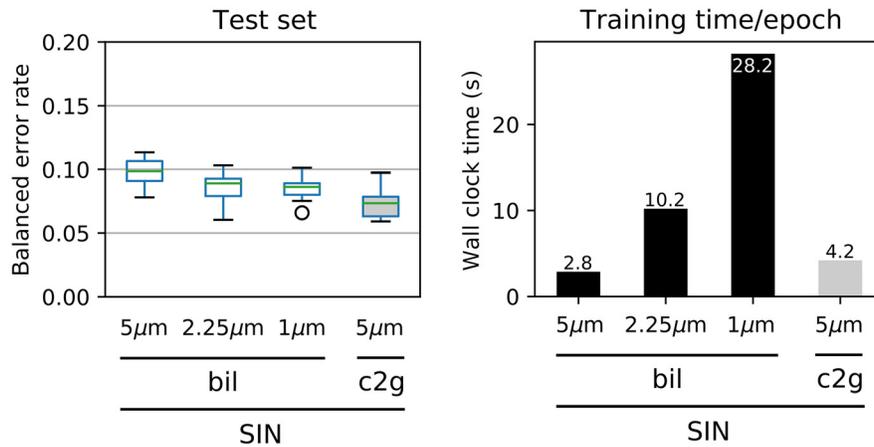


**Fig. 10** Smoothed training curves for loss function and balanced error rate as well as test set ROC curve with mean area under the curve for all ten SIN model runs. (a) SIN trained on bil rescaled images (bil-SIN) and (b) SIN trained on Cell2Grid images (c2g-SIN).

on Cell2Grid data led to an 18.1% and 25.3% relative reduction of test set error rate for VGG and SIN models, respectively, compared with conventional bil rescaling.

As an additional experiment, we investigated if higher input image resolution in the bil-SIN models could improve model performance compared Cell2Grid-SIN. We therefore rescaled our data to 2.25 and 1.0  $\mu\text{m}/\text{px}$  (images sizes  $300 \times 255$  px and  $675 \times 506$  px, respectively) using bil and used the same training setup as before.

Figure 11 shows that test set error rates of bil-SIN decreased with increased image resolution, but remained above Cell2Grid-SIN, even at 1  $\mu\text{m}/\text{px}$ . However, the larger images sizes at higher resolutions led to longer training times due to the increased number of network parameters in the top layers and slower data augmentation. Specifically, per-epoch training time of Cell2Grid-SIN at 5  $\mu\text{m}$  was decreased by 85% compared with bil-SIN at 1  $\mu\text{m}$  while simultaneously achieving lower test set error rates.



**Fig. 11** Balanced test set error rates and per-epoch training times for bil-SIN models at different resolutions (5, 2.25, and 1  $\mu\text{m}$ ) compared with Cell2Grid-SIN (5  $\mu\text{m}$ ). Our method highlighted in gray.

## 4 Discussion

CNN architectures have become increasingly complex, requiring long training times and powerful hardware, particularly when using large images for training.<sup>1-3,5</sup> In some domains, rescaling images to a much lower resolution still allows for efficient image classification. In the biomedical domain, however, this is rarely applicable, as often not only the whole picture but small details can be crucial for classification, such images containing cells. In this paper, we presented an alternative to rescaling that allows for efficient training of CNNs without altering the essential morphological properties of these images.

The Cell2Grid images are different from conventional images in a number of ways: (1) they contain no natural gradients, because every pixel in the image is an independent object (a biological cell); (2) they might contain many empty pixels (numerically zero) in areas without assigned cells; and (3) they cannot be rotated, zoomed, sheared, or transformed for data augmentation with methods that include pixel interpolation because the value and integrity of individual pixels matter (see [Supplementary Material S.5](#)). As shown in [Fig. 8\(d\)](#), final Cell2Grid images are sparse, containing several empty (black) pixels. In our example, Cell2Grid images (at 5  $\mu\text{m}/\text{px}$ ) had ten times fewer pixels in each spatial dimension compared to the original fm-IHC images (0.5  $\mu\text{m}/\text{px}$ ), corresponding to an image-to-image compression ratio of 100. In contrast, in conventional image downscaling methods, pixels that belong to different biological cells may be used to obtain a pixel value of the new target resolution through interpolation. This makes it difficult or sometimes even impossible to identify individual cells and their true marker expression in strongly downsampled images [[Fig. 8\(b\)](#)]. Due to the cell segmentation step prior to Cell2Grid that uses the original high-resolution images, the concept of individual cells is retained in its output and the cell features, as extracted during cell segmentation, are left unaltered. Keeping cell information intact is key in many applications, such when studying colon cancer,<sup>71</sup> where the presence of just a few cells with distinct phenotype has been shown to be an important image feature.

In our case study, we mainly focused on the average fluorescent marker intensities as features for each cell as this leads to images that are most similar to conventionally rescaled images. Depending on the scientific question, other features may be relevant, including size and shape of nuclei and cells and marker heterogeneity within the cells. Our experiment on the inclusion of cell shape and size properties as additional image color channels revealed that these features may contribute to better performing models but tend to be more prone to overfitting and therefore need to be used with care. Using additional image color channels, including marker distribution properties (see [Supplementary Material S.4](#)) may be used in the same fashion.

We used a SIN with  $<10^4$  parameters, which is orders of magnitude smaller than our modified VGG architecture with  $5.6 \cdot 10^7$  parameters. We found that the smaller SIN model more stable during training compared with VGG and performed, on average, better than VGG on unseen test

samples. As smaller networks require less GPU memory, they allow for larger batch sizes or larger input images during training. This is particularly relevant toward training on whole-slide images to capture the entire tissue heterogeneity, a feature commonly observed in tumors.<sup>72</sup>

For evaluation, we compared Cell2Grid with conventional image rescaling with bil and cub interpolations as well as lanc sampling. Instead of merging pixels based on their proximity to the pixels of the target resolution, our method merges only pixels that belong to the same biological cell, which is a process related to the concept of superpixels.<sup>73,74</sup> This cell-based approach of Cell2Grid enables subsequent analysis, such as cell counting, phenotyping, and cell-based interpretation of a trained neural network.<sup>75,76</sup> Graph-based methods, including GNNs<sup>45,76–78</sup> also rely on the output of cell segmentation or nuclei identification in histologic images but they require a cell graph based on the segmented cells and their coordinates. Our method follows a different approach by converting the list of segmented cells into a low-resolution image suited for training a CNN. While the input data for our method is the same as for cell graph creation, the output of Cell2Grid allows for the training of conventional, well-established CNNs, as compared with the more recent GNN architectures.

In the last decade, higher standards for privacy protection complicated the sharing of data obtained from human samples. The Cell2Grid provides a way to share pathology data in a compact format that does not contain the original high-resolution histologic image. We therefore hope that this technique reduces concerns for data protection and encourages researchers to release datasets in this format.

#### 4.1 Limitations and Future Work

The research scope of this work focused on the investigation of an alternative data representation of high-resolution histologic images to facilitate their use with CNNs. Therefore, the first step of our algorithm directly relied on the output provided by commercial software available in most laboratories and provided a standardized scenario on which to experiment. Moreover, in our current implementation, an IHC expert had to choose which method and parameters were best suited for the biological tissue. Because our method strongly depends on this cell segmentation step, this set of constraints constitutes an important limitation. Our research scope did not include an investigation of the effects of different cell segmentation algorithms, but fully automated cell segmentation is an active field of research<sup>24,30,31,53,54</sup> that we intend to explore in future work with recently published methods showing promising results.<sup>30,31</sup>

Additionally, Cell2Grid requires additional processing time compared to conventional image rescaling. As shown in Sec. 3.2, data preprocessing using Cell2Grid is slower than conventional image rescaling even without accounting for cell segmentation time. Furthermore, cell graph creation was significantly faster than our method, both starting from cell segmentation data. This additional data processing time needs to be weighed against any potential gains of our method for downstream applications, like a potential reduction in CNN training time or classification accuracy.

In our experiments, we focused on a small set of cell features to construct Cell2Grid images, i.e., the mean marker expression within each cell. This enables a direct comparison with conventional image rescaling methods as the number and type of image channels remains the same. To add additional information to Cell2Grid images, other features, such as marker distribution properties across the cell and cell shape features, may be used as additional image channels in Cell2Grid output, as demonstrated by our experiments (see results for c2gShape and Sec. S.4 in the [Supplementary Material](#)). However, extracellular information or cell-based information that is either not extracted during cell segmentation or is not used as a dedicated image channel in Cell2Grid is inevitably lost from the data. This is a limiting factor for our method, especially for applications where a pathologist—or an image classification model—requires this information for accurate assessment of the images.

Regarding the assignment of cells to the target grid (step 2 of Cell2Grid), care needs to be taken, as a large target grid spacing  $d$  may lead to deletion of individual cells. Even if in small numbers, cell deletion may be undesired if tissue areas with high cell density are of particular interest, e.g., immune cell clusters in insulinitis.<sup>79</sup> In this case, a smaller target grid spacing  $d$ , a larger local conflict resolution size  $w_{\max}$ , or both are required, potentially increasing processing

time. In our method, the precise location of individual cells is approximated to fit the target grid during cell assignment. However, because cells are moving in live tissue, their exact position in the final tissue section depends on the time of sampling as well as the location and angle of tissue sectioning. We estimate that the net effect of these factors is greater than the local shift in cell positions introduced by Cell2Grid, which for most cells is smaller than the target grid spacing  $d$ .

Even though we demonstrated the benefits of Cell2Grid using fm-IHC images of size  $674 \times 506 \mu\text{m}$ , a WSI of a tissue section can be up to 30 times bigger in each spatial dimension. In our experiments, we did not yet investigate the utility of Cell2Grid when applied to WSI data. Additionally, we only considered image rescaling instead of image compression methods (such as JPEG2000<sup>80</sup>) as reference data processing methods CNN training, because the latter only reduce image storage size and not memory size once decompressed and loaded into memory for model training.<sup>8</sup>

Our study is limited using only a single fm-IHC dataset of colorectal cancer patients that comprises only 54 patients and 1,353 images. These numbers are small compared with the large number of CNN parameters ( $10^7$  and  $10^4$  for VGG and SIN architectures, respectively). For a more comprehensive analysis of our method, future experiments on different datasets with larger cohorts are required to identify potential weaknesses and demonstrate robustness of our algorithm for different tissue types. Additionally, the utility of our method when applied to other image modalities, such as conventional H&E-stained images, needs to be investigated in future work.

In addition, we want to build on our current work and develop further two areas of interest.

1. Volumetric images: when studying tumor samples, even whole-slide scans of two-dimensional tissue sections may not be enough to fully capture the heterogeneity inherent to colorectal cancer.<sup>72</sup> In this case, multiple consecutive sections of a tissue sample can be acquired with the distance between two sections being typically  $3 - 4 \mu\text{m}$ .<sup>81</sup> By setting the target grid spacing  $d$  of Cell2Grid equal to the section spacing, an isotropic 3D model of the tissue can be generated from consecutive tissue sections, such that each cell is represented by a single voxel. These volumetric images can be used to train 3D CNNs<sup>82</sup> that capture the entire structure of a tissue sample. A showcase using consecutive sections of a colon cancer sample is currently being developed.
2. Synthetic image creation: in medicine, an important goal of using machine learning tools is the extraction of knowledge from a trained model. As summarized by Jaume et al.,<sup>76</sup> model explanations that operate on a pixel-level suffer from real interpretability by not taking into account that the important entities of a histologic image are the individual biological cells and their spatial distribution.<sup>83,84</sup> In a first use-case of murine pancreatic islets, we showed how to exploit the cell-based data representation of Cell2Grid to algorithmically create synthetic image datasets with known ground truth using a procedural algorithm for the creation of Cell2Grid images and a cGAN to turn Cell2Grid images into original fm-IHC images.<sup>85</sup> This will support the creation of a pipeline for knowledge extraction from CNNs<sup>75,86-89</sup> trained on Cell2Grid images.

## 5 Conclusions

In this paper, we introduced Cell2Grid, which is an algorithm for efficient representation of histologic images whose output can be used directly to train CNNs. As a case study for evaluation, we used fm-IHC images to predict 5-year relapse risk of stage II colon cancer patients. Our results showed that Cell2Grid preserves small-scale cell level information and how this information can reduce the test set error rate of a neural network by 25% relative to conventional bil rescaling at the same resolution while also reducing training time by 85% compared with higher resolution reference models. However, future experiments are required to investigate the performance of our method when applied to other image types, such as H&E-stained images and conventional chromogenic IHC.

Compared with conventional image rescaling methods, images transformed with Cell2Grid are directly suited for cell-based analysis, and in addition, they might simplify interpretation of trained CNNs. The Cell2Grid therefore opens a door for further exploring the use of histologic

images with deep learning architectures and in a wider field of applications, such as volumetric images and synthetic image creation.

## Disclosures

A patent for Cell2Grid has been filed by CBmed, Center for Biomarker Research in Medicine GmbH, with L.H. as inventor.

## Acknowledgments

This work was supported by the Österreichische Forschungsförderungsgesellschaft FFG (Grant No. 865836). Use of samples from human subjects: the use of data obtained from human samples stored in a biobank was approved by the Ethics Committee of the Medical University of Graz ("Spatial distribution of immune cells as a prognostic and predictive biomarker in colon cancer - a retrospective study"; Ethics Committee number: 28-342 ex 15/16).

## References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.* 25, F. Pereira et al., Eds., Curran Associates, Inc., pp. 1097–1105 (2012).
2. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1–9 (2015).
3. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv [csCV] arXiv: 1409.1556 (2014).
4. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision* 115(3), 211–252 (2015).
5. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 770–778 (2015).
6. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 248–255 (2009).
7. G. Romero et al., "Digital pathology consultations—a new era in digital imaging, challenges and practical applications," *J. Digital Imaging* 26(4), 668–677 (2013).
8. J. Konsti et al., "Effect of image compression and scaling on automated scoring of immunohistochemical stainings and segmentation of tumor epithelium," *Diagn. Pathol.* 7, 29 (2012).
9. P. Mobadersany et al., "Predicting cancer outcomes from histology and genomics using convolutional networks," *Proc. Natl. Acad. Sci. U. S. A.* 115(13), E2970–E2979 (2018).
10. J. N. Kather et al., "Predicting survival from colorectal cancer histology slides using deep learning: a retrospective multicenter study," *PLoS Med.* 16(1), e1002730 (2019).
11. J. Xu et al., "A Deep Convolutional Neural Network for segmenting and classifying epithelial and stromal regions in histopathological images," *Neurocomputing* 191, 214–223 (2016).
12. P. Khosravi et al., "Deep convolutional neural networks enable discrimination of heterogeneous digital pathology images," *EBioMed.* 27, 317–328 (2018).
13. A. Gertych et al., "Convolutional neural networks can accurately distinguish four histologic growth patterns of lung adenocarcinoma in digital slides," *Sci. Rep.* 9, 1483 (2019).
14. W. Bulten et al., "Epithelium segmentation using deep learning in H&E-stained prostate specimens with immunohistochemistry as reference standard," *Sci. Rep.* 9, 864 (2019).
15. W. C. C. Tan et al., "Overview of multiplex immunohistochemistry/immunofluorescence techniques in the era of cancer immunotherapy," *Cancer Commun.* 40(4), 135–153 (2020).
16. D. J. Fassler et al., "Deep learning-based image analysis methods for brightfield-acquired multiplex immunohistochemistry images," *Diagn. Pathol.* 15(1), 100 (2020).

17. D. Maric et al., “Whole-brain tissue mapping toolkit using large-scale highly multiplexed immunofluorescence imaging and deep neural networks,” *Nat. Commun.* **12**, 1550 (2021).
18. L. Hou et al., “Patch-based convolutional neural network for whole slide tissue image classification,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, pp. 2424–2433 (2016).
19. B. Smith et al., “Developing image analysis pipelines of whole-slide images: pre- and post-processing,” *J. Clin. Transl. Sci.* **5**(1), e38 (2020).
20. A. Pirovano et al., “Automatic feature selection for improved interpretability on whole slide imaging,” *Mach. Learn. Knowl. Extract.* **3**(1), 243–262 (2021).
21. M. Y. Lu et al., “Data-efficient and weakly supervised computational pathology on whole-slide images,” *Nat. Biomed. Eng.* **5**, 555–570 (2021).
22. R. J. Chen et al., “Whole slide images are 2D point clouds: context-aware survival prediction using patch-based graph convolutional networks,” *Lect. Notes Comput. Sci.* **12908**, 339–349 (2021).
23. M. Lerousseau et al., “SparseConvMIL: sparse convolutional context-aware multiple instance learning for whole slide image classification,” in *Proc. MICCAI Workshop on Comput. Pathol.*, pp. 129–139 (2021)
24. Q. D. Vu et al., “Methods for segmentation and classification of digital microscopy tissue images,” *Front. Bioeng. Biotechnol.* **7**, 53 (2019).
25. M. Abdolhoseini et al., “Segmentation of heavily clustered nuclei from histopathological images,” *Sci. Rep.* **9**, 4551 (2019).
26. X. Xiao et al., “Dice-XMBD: deep learning-based cell segmentation for imaging mass cytometry,” *Front. Genet.* **12**, 721229 (2021).
27. M. Y. Lee et al., “CellSeg: a robust, pre-trained nucleus segmentation and pixel quantification software for highly multiplexed fluorescence images,” *BMC Bioinf.* **23**(1), 46 (2022).
28. C. McQuin et al., “CellProfiler 3.0: next-generation image processing for biology,” *PLoS Biol.* **16**(7), e2005970 (2018).
29. C. T. Rueden et al., “ImageJ2: ImageJ for the next generation of scientific image data,” *BMC Bioinf.* **18**(1), 529 (2017).
30. W. Han et al., “Cell segmentation for immunofluorescence multiplexed images using two-stage domain adaptation and weakly labeled data for pre-training,” *Sci. Rep.* **12**, 4399 (2022).
31. N. F. Greenwald et al., “Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning,” *Nat. Biotechnol.* **40**(4), 555–565 (2022).
32. A. Baddeley and R. Turner, “SPATSTAT: an R package for analyzing spatial point patterns,” *J. Stat. Software* **12**, 1–42 (2005).
33. D. Schapiro et al., “histoCAT: analysis of cell phenotypes and interactions in multiplex image cytometry data,” *Nat. Methods* **14**(9), 873–876 (2017).
34. K. S. Johnson, “Phenoptr: InForm helper functions. R package,” Github, 2018, <https://github.com/PerkinElmer/phenoptr> (accessed 15 Nov. 2019).
35. C. M. Schürch et al., “Coordinated cellular neighborhoods orchestrate antitumoral immunity at the colorectal cancer invasive front,” *Cell* **182**(5), 1341–1359.e19 (2020).
36. M. Angelova et al., “Evolution of metastases in space and time under immune selection,” *Cell* **175**(3), 751–765.e16 (2018).
37. A. Lu et al., “Comparison of biomarker modalities for predicting response to PD-1/PD-L1 checkpoint blockade: a systematic review and meta-analysis,” *JAMA Oncol.* **5**(8), 1195–1204 (2019).
38. I. P. Nearchou et al., “Automated analysis of lymphocytic infiltration, tumor budding, and their spatial relationship improves prognostic accuracy in colorectal cancer,” *Cancer Immunol. Res.* **7**(4), 609–620 (2019).
39. Y. K. Huang et al., “Macrophage spatial heterogeneity in gastric cancer defined by multiplex immunohistochemistry,” *Nat. Commun.* **10**, 3928 (2019).
40. C. Lewis, P. Xun, and K. He, “Effects of adjuvant chemotherapy on recurrence, survival, and quality of life in stage II colon cancer patients: a 24-month follow-up,” *Support Care Cancer* **24**(4), 1463–1471 (2016).

41. P. Dalerba et al., “CDX2 as a prognostic biomarker in Stage II and Stage III colon cancer,” *N. Engl. J. Med.* **374**(3), 211–222 (2016).
42. S. E. Rebutuzzi et al., “Adjuvant chemotherapy for Stage II colon cancer,” *Cancers* **12**(9), 2584 (2020).
43. R. Caso, A. Fabrizio, and M. Sosin, “Prolonged follow-up of colorectal cancer patients after 5 years: to follow or not to follow, that is the question (and how)!” *Ann. Transl. Med.* **8**(5), 164 (2020).
44. E. Wulczyn et al., “Interpretable survival prediction for colorectal cancer using deep learning,” *NPJ Digital Med.* **4**(1), 71 (2021).
45. Y. Zhou et al., “CGC-net: cell graph convolutional network for grading of colorectal cancer histology images,” in *IEEE/CVF Int. Conf. Comput. Vision Workshop (ICCVW)*, IEEE (2019).
46. G. Jaume et al., “Towards explainable graph representations in digital pathology,” in *ICML 2020 Workshop on Comput. Biol.* (2020).
47. C. Gunduz, B. Yener, and S. H. Gultekin, “The cell graphs of cancer,” *Bioinformatics* **20**(Suppl. 1), i145–i151 (2004).
48. J. You, R. Ying, and J. Leskovec, “Design space for graph neural networks,” in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, Curran Associates Inc. (2020).
49. J. Zhou et al., “Graph neural networks: a review of methods and applications,” *AI Open* **1**, 57–81 (2018).
50. D. Ahmedt-Aristizabal et al., “A survey on graph-based deep learning for computational histopathology,” *Comput. Med. Imaging Graphics* **95**, 102027 (2022).
51. S. Deng et al., “Deep learning in digital pathology image analysis: a survey,” *Front. Med.* **14**(4), 470–487 (2020).
52. J. van der Laak, G. Litjens, and F. Ciompi, “Deep learning in histopathology: the path to the clinic,” *Nat. Med.* **27**(5), 775–784 (2021).
53. R. M. Thomas and J. John, “A review on cell detection and segmentation in microscopic images,” in *Int. Conf. Circuit, Power and Comput. Technol. (ICCPCT)*, pp. 1–5 (2017).
54. S. Graham et al., “Hover-Net: simultaneous segmentation and classification of nuclei in multi-tissue histology images,” *Med. Image Anal.* **58**, 101563 (2018).
55. J. Munkres, “Algorithms for the assignment and transportation problems,” *J. Soc. Ind. Appl. Math.* **5**(1), 32–38 (1957).
56. N. L. Johnson, *Univariate Discrete Distributions*, Wiley (2005).
57. D. Giavarina, “Understanding bland Altman analysis,” *Biochem. Med.* **25**(2), 141–151 (2015).
58. K. Turkowski, “Filters for common resampling tasks,” 1990, <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.7898> (accessed 14 Aug. 2022).
59. J. Qu et al., “Gastric pathology image classification using stepwise fine-tuning for deep neural networks,” *J. Healthcare Eng.* **2018**, 8961781 (2018).
60. P. Bandi et al., “From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge,” *IEEE Trans. Med. Imaging* **38**(2), 550–560 (2019).
61. Q. Guan et al., “Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study,” *J. Cancer* **10**(20), 4876–4882 (2019).
62. Y. Wang et al., “Using deep convolutional neural networks for multi-classification of thyroid tumor by histopathology: a large-scale pilot study,” *Ann. Transl. Med.* **7**(18), 468 (2019).
63. S. Pang et al., “VGG16-T: a novel deep convolutional neural network with boosting to identify pathological type of lung cancer in early stage by CT images,” in *Proc. Int. Joint Conf. Bioinforma Syst. Biol. Intell. Comput.*, Vol. 13, No. 1, p. 771 (2020).
64. G. S. Ioannidis et al., “Pathomics and deep learning classification of a heterogeneous fluorescence histology image dataset,” *NATO Adv. Sci. Inst. Ser. E Appl. Sci.* **11**(9), 3796 (2021).
65. F. Chollet et al., “Keras,” 2015, <https://github.com/fchollet/keras>.
66. B. Kieffer et al., “Convolutional neural networks for histopathology image classification: training vs. using pre-trained networks,” in *Seventh Int. Conf. Image Process. Theory, Tools and Appl. (IPTA)*, pp. 1–6 (2017).

67. M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv [csNE], arXiv: 1312.4400 (2013).
68. D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral," arXiv [csLG], arXiv: 2006.12138 (2020).
69. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Int. Conf. Learn. Represent. (ICLR)*, Ithaca, New York (2015).
70. M. Abadi et al., "TensorFlow: a system for large-scale machine learning," arXiv [csDC], 2016, <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf> (accessed 24 March 2021).
71. J. Galon et al., "Towards the introduction of the 'Immunoscore' in the classification of malignant tumours," *J. Pathol.* **232**(2), 199–209 (2014).
72. O. M. Zlatian et al., "Histochemical and immunohistochemical evidence of tumor heterogeneity in colorectal cancer," *Rom. J. Morphol. Embryol.* **56**(1), 175–181 (2015).
73. S. Akbar et al., "Tumor localization in tissue microarrays using rotation invariant superpixel pyramids," in *IEEE 12th Int. Symp. Biomed. Imaging (ISBI)*, pp. 1292–1295 (2015).
74. M. E. A. Bechar et al., "Influence of normalization and color features on super-pixel classification: application to cytological image segmentation," *Aust. Phys. Eng. Sci. Med.* **42**(2), 427–441 (2019).
75. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1135–1144, Association for Computing Machinery (2016).
76. G. Jaume et al., "Quantifying explainers of graph neural networks in computational pathology," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 8102–8112 (2020).
77. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th Int. Conf. Learn. Represent.*, Toulon, France, pp. 24–26 (2017).
78. K. Xu et al., "How powerful are graph neural networks?," in *Int. Conf. Learn. Represent.* (2019).
79. M. A. Atkinson et al., "Organisation of the human pancreas in health and in diabetes," *Diabetologia* **63**(10), 1966–1973 (2020).
80. D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Springer US (2002).
81. S. Libard, D. Cerjan, and I. Alafuzoff, "Characteristics of the tissue section that influence the staining outcome in immunohistochemistry," *Histochem. Cell Biol.* **151**(1), 91–96 (2019).
82. T. Kurc et al., "Segmentation and classification in digital pathology for glioma research: challenges and deep learning approaches," *Front. Neurosci.* **14**, 27 (2020).
83. F. Pagès et al., "International validation of the consensus Immunoscore for the classification of colon cancer: a prognostic and accuracy study," *Lancet* **391**(10135), 2128–2139 (2018).
84. J. Galon and D. Bruni, "Approaches to treat immune hot, altered and cold tumours with combination immunotherapies," *Nat. Rev. Drug Discov.* **18**(3), 197–218 (2019).
85. L. Herbsthofer et al., "Procedural generation of synthetic multiplex immunohistochemistry images using cell-based image compression and conditional generative adversarial networks," *Proc. SPIE* **12039**, 120390N (2022).
86. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Comput. Vision – ECCV 2014*, Springer International Publishing, pp. 818–833 (2014).
87. A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5188–5196 (2015).
88. Q. Zhang, Y. Nian Wu, and S. C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 8827–8836 (2018).
89. Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: understanding and mitigating network overthinking," in *Proc. 2019 Int. Conf. Mach. Learn. (ICML)* (2019)
90. H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.* **2**(1–2), 83–97 (1955).
91. F. Bourgeois and J. C. Lassalle, "An extension of the Munkres algorithm for the assignment problem to rectangular matrices," *Commun. ACM* **14**(12), 802–804 (1971).

92. N. Tomizawa, "On some techniques useful for solution of transportation network problems," *Networks* **1**(2), 173–194 (1971).
93. E. J. Wood, "Cellular and molecular immunology (5th ed.): Abbas A. K., and Lichtman, A. H.," *Biochem. Mol. Biol. Educ.* **32**(1), 65–66 (2004).
94. H. Süße and E. Rodner, *Bildverarbeitung Und Objekterkennung: Computer Vision in Industrie Und Medizin*, Springer Vieweg, Wiesbaden (2014).
95. A. W. Paeth, "A fast algorithm for general raster rotation," in *Proc. Graphics Interface '86/Vision Interface '86*, Canadian Information Processing Society, pp. 77–81 (1986).

**Laurin Herbsthofer** is a data scientist at a biomedical research company. In 2017, he received his MSc degree in theoretical and computational physics at University of Graz, Austria. He is currently enrolled in a PhD program at Medical University of Graz, Austria. His main focus is the analysis of histologic multiplex images for research and clinical applications, especially using cell-based data representations and machine learning tools.

Biographies of the other authors are not available.