Computerized Medical Imaging and Graphics 70 (2018) 43–52

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

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

Deep learning nuclei detection: A simple approach can deliver state-of-the-art results

Henning Höfener[a,*], André Homeyer[a], Nick Weiss[a], Jesper Molin[b], Claes F. Lundström[b,c], Horst K. Hahn[a,d]

[a] Fraunhofer MEVIS, Am Fallturm 1, 28359, Bremen, Germany
[b] Sectra AB, Teknikringen 20, 58330, Linköping, Sweden
[c] Center for Medical Image Science and Visualization, Linköping University, 58183, Linköping, Sweden
[d] Jacobs University, Campus Ring 1, 28759, Bremen, Germany

ARTICLE INFO

Article history:
Received 8 February 2018
Received in revised form 13 July 2018
Accepted 23 August 2018

Keywords:
Nuclei detection
Deep learning
PMap
Histology
Image analysis

ABSTRACT

Background: Deep convolutional neural networks have become a widespread tool for the detection of nuclei in histopathology images. Many implementations share a basic approach that includes generation of an intermediate map indicating the presence of a nucleus center, which we refer to as PMap. Nevertheless, these implementations often still differ in several parameters, resulting in different detection qualities.
Methods: We identified several essential parameters and configured the basic PMap approach using combinations of them. We thoroughly evaluated and compared various configurations on multiple datasets with respect to detection quality, efficiency and training effort.
Results: Post-processing of the PMap was found to have the largest impact on detection quality. Also, two different network architectures were identified that improve either detection quality or runtime performance. The best-performing configuration yields f1-measures of 0.816 on H&E stained images of colorectal adenocarcinomas and 0.819 on Ki-67 stained images of breast tumor tissue. On average, it was fully trained in less than 15,000 iterations and processed 4.15 megapixels per second at prediction time.
Conclusions: The basic PMap approach is greatly affected by certain parameters. Our evaluation provides guidance on their impact and best settings. When configured properly, this simple and efficient approach can yield equal detection quality as more complex and time-consuming state-of-the-art approaches.
© 2018 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The quantification of cell nuclei in histological images is essential for many pathological assessments, including the determination of various biomarkers. Prominent examples in cancer diagnosis are the Ki-67 index or the progesterone and estrogen receptor status. Detecting nuclei also enables the quantification of tumor immune infiltrates, which have been shown to be of strong prognostic importance (Mahmoud et al., 2011), and are commonly assessed in immunotherapy trials (Denkert et al., 2016).

Such assessments are usually performed by visual estimation, which is labor- and time-intensive and can lead to high inter- and intra-observer variability (Andrion et al., 1995). The ongoing digitalization in pathology allows for automated analysis methods to support pathologists at such tasks and to increase the reliability of quantitative assessments.

However, the automatic detection of cell nuclei is challenging. The appearance of nuclei varies considerably with staining and tissue preparation conditions, as well as with different nuclear types and pathologies.

The first attempts to automate nuclei detection date back to the mid-1950s (Meijering, 2012), starting with static rule-based approaches from simple intensity thresholds to using intensity-derived features. Those approaches suffered from not being able to capture the complexity of the input data sufficiently well. The next generation of methods addressed the aforementioned variability by using hand-crafted features and applying machine-learning to build more complex and flexible rule sets (Arteta et al., 2012;

* Corresponding author.
E-mail addresses: henning.hoefener@mevis.fraunhofer.de (H. Höfener), andre.homeyer@mevis.fraunhofer.de (A. Homeyer), nick.weiss@mevis.fraunhofer.de (N. Weiss), Jesper.Molin@sectra.com (J. Molin), claes.lundstrom@liu.se (C.F. Lundström), horst.hahn@mevis.fraunhofer.de (H.K. Hahn).

https://doi.org/10.1016/j.compmedimag.2018.08.010
0895-6111/© 2018 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Kårsnäs et al., 2011; Vink et al., 2013). Recent developments mainly employ convolutional neural networks (CNN) (Jacobs et al., 2017; Janowczyk and Madabhushi, 2016; Sirinukunwattana et al., 2016; Wang et al., 2016; Xie et al., 2016, 2015a,b; Xing et al., 2016), as those tend to yield significantly better results. The first major breakthrough was reported by Cireşan et al. (2013), who were able to detect mitotic nuclei with an f1-measure of 0.782, while the closest competitors achieved 0.718.

Most deep learning–based nuclei detection methods employ CNNs to predict a value for each input image pixel. That value represents the proximity to a nucleus center or the probability of being close to one. Together, the values of all input image pixels constitute a map, which we refer to as PMap. Nuclei positions are afterwards determined by finding local maxima in the PMap. Predicting the PMap can be interpreted as either a classification or a regression problem. The classification problem is to distinguish *nucleus center* and *background* positions and to populate the PMap with the each position's probability to belong to the *nucleus center* class, whereas the regression problem is to map a position to a continuous value, which is dependent on the distance to the nearest nucleus center. This basic PMap approach will be described in more detail in Section 2.1.

The basic PMap approach is controlled by several parameters. Examples are the post-processing of the PMap before finding local maxima, the use of data augmentation or the use of dropout.

1.1. Related work

There are different variants of the basic PMap approach proposed in the literature, using CNN classification or regression, even if that term is not used.

As described above, Cireşan et al. (2013) have used CNN classification for the detection of mitoses. Their approach uses a 12 and a 10 layer deep network and achieves processing speeds between 0.01 and 0.03 megapixels per second at prediction time. CNN classification with the 8 layer deep AlexNet (Krizhevsky, 2010) has been used by Janowczyk and Madabhushi (2016) to detect lymphocytes in breast cancer images. They have achieved an f1-measure of 0.900. The 7 layer deep LeNet (LeCun et al., 1998) classification network has been applied by Wang et al. (2016) to detect nuclei for a subsequent cell subtype classification. For the detection, they have reported an f1-measure of 0.822. Khoshdeli et al. (2017) have used a 5 layer deep CNN classification for the detection of nuclei in Hematoxylin and Eosin (H&E) stained images of various tissue types. They have proposed to preprocess the input images by extracting the Hematoxylin channel using color deconvolution and applying a Laplacian of Gaussian filter. The result is then fed into to the network. An f1-measure of 0.722 has been reported. Jacobs et al. (2017) have used a 14 layer deep regression network to detect nuclei in H&E stained prostate cancer biopsies for a subsequent nucleus type classification. The authors have evaluated transfer learning for the application with limited training data. They have trained on colon images and have fine-tuned their model with the prostate images. They have reported f1-measures between 0.849 and 0.864, depending on the amount of training data for the fine-tuning, as well as a processing speed of 2.2 megapixels per second.

Some approaches leave out the extraction of local maxima from the PMap. Xie et al. (2016) have estimated the nuclei count in an image region by integrating the PMap over that region. They have applied a 9 layer deep network. Xing et al. (2016) have applied a threshold to the PMap and have used the connected regions as initialization for nuclei segmentation. The generation of the PMap has been performed with 0.008 megapixels per second.

In other publications, the basic PMap approach has been used as a baseline algorithm to compare the proposed methods with. Xie et al. (2015a) have mapped each pixel of the input image to a 2D-vector pointing to the nearest nucleus center, using an 8 layer deep network. At prediction time, the positions, where the vectors point to, are accumulated to form a PMap. On Ki-67 stained neuroendocrine tumor (NET) images they have reported an f1-measure of 0.815 and a processing speed of 0.007 megapixels per second. They have compared their method with the basic PMap approach using CNN classification, which has yielded an f1-measure of 0.784. In another publication, the same authors have used a 7 layer deep network to predict a small region of the PMap at once instead of a single pixel value (Xie et al., 2015b). As before, they have accumulated the predictions to generate the PMap. They have evaluated the approach on H&E stained breast tumor images, Ki-67 stained NET images and phase contrast images of HeLa cervical cancer cells. F1-measures of 0.913, 0.906 and 0.957 have been reported, respectively. Processing speed has been 0.01 megapixels per second. A comparison with both CNN classification and regression according to the basic PMap approach has been conducted, but no quantitative measures have been given. Sirinukunwattana et al. (2016) have proposed a similar approach of predicting a region of the PMap. In contrast to (Xie et al., 2015b), the first 6 layers of their network are followed by a parameter estimation layer and a spatially constrained regression layer. They have reported an f1-measure of 0.802 on H&E stained images of colorectal adenocarcinomas and processing speed of 0.02 megapixels per second. They have compared their method with the basic PMap approach using CNN regression, for which an f1-measure of 0.692 has been reported. Xu et al. (2016) have used multiple stacked auto-encoders to learn feature representations of the input images in an unsupervised manner. The features are then fed into a softmax classifier, which classifies each input patch as either nuclear or non-nuclear. The softmax classifier has been trained supervisedly and the authors have reported an f1-measure of 0.845 on H&E stained breast cancer images. They compare their method with the basic PMap approach using CNN classification. There, an f1-measure of 0.820 and processing speed of 0.04 megapixels per second have been reported.

We want to stress here that the reported f1-measures should not be compared directly. Most approaches have been evaluated using different datasets with different nuclear types, varying quality and tissue complexity. Additionally, the hardware used to perform the experiments, especially the usage of GPUs, has a great influence on the processing speed. Although only comparable to a very limited extent, we listed processing speeds if available for completeness. Only few of the approaches above explicitly focused on processing time, although speed is critical when aiming at applying these methods in clinical routine.

In summary, for nuclei detection using deep learning, the basic PMap approach is widely used in the literature. Even if not termed basic PMap approach, numerous publications describe such methods either as the proposed or as alternative solutions for nuclei detection tasks. However, there are some parameters of these methods that differ from case to case. Most of the papers above only present a single configuration of them. There is no systematic evaluation of the influence and importance of the individual parameters.

The main contribution of this work is a systematic listing, evaluation and comparison of these parameters. We assess the impact of the individual parameters with respect to detection quality, efficiency and training effort. By doing so, we give guidance on which parameters to focus on when optimizing nuclei detection with the basic PMap approach. The second contribution is to combine those parameter settings, which perform best in our experiments and to evaluate this configuration. We show that the basic PMap approach delivers state-of-the-art results when parameterized well.

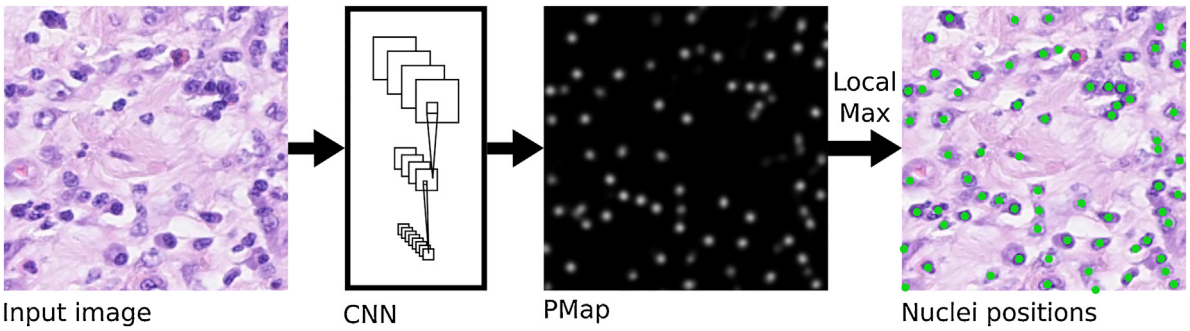

Fig. 1. The prediction workflow of the basic PMap approach.

2. Materials and methods

2.1. Basic PMap approach

In the vast majority of nuclei detection methods based on machine-learning, each pixel of an input image is assigned a value representing either the probability of being close to a nucleus center or the proximity to the nearest one. The entirety of these values for one input image is often termed probability map or proximity map and can be represented as a gray-scale image. While these two terms are semantically different, they share the same core properties: Intensities are high (towards 1.0) near nuclei center positions, lower at their periphery and lowest (towards 0.0) in background areas. In this paper, we refer to it as PMap.

When using PMaps, a popular and very straightforward approach is to first extract either a set of hand-crafted features or to crop an image patch for each input pixel. In case of CNNs, patches of the same extent as the network's receptive field are extracted. Secondly, reference or target PMap values are generated from reference annotations. Both are then fed to a machine learning–algorithm to learn the mapping. This way, the algorithm can afterwards predict PMap values for yet unseen input image pixels.

Training a machine learning–algorithm to generate a PMap can be formulated as either a classification or a regression problem. Considering the classification problem, there are two classes, namely *nucleus center* and *background*. For training, the target takes the discrete value 1 or 0, representing the class. At prediction time, the probabilities for the *nucleus center* class are used as the PMap values. Regarding the regression problem, the proximity to the nearest nucleus center is used to generate continuous target values between 0 and 1. The predictions of the machine learning–algorithm are then directly used as PMap values.

Usually, after generating a PMap from an input image, the positions of the nuclei centers are determined by finding local maxima in the PMap that exceed a certain threshold. A local maximum in this context is a region of equal values where all neighboring pixels have lower values. That region may be as small as a single pixel. We refer to the procedure of CNN-based PMap generation, followed by maxima finding as the basic PMap approach, which is depicted in Fig. 1.

2.2. Evaluation metric

The quality of nuclei detection is assessed by comparing the resulting nuclei positions with annotated nuclei center markers. For each detected nucleus position, the nearest center marker annotation is found. If the distance between these positions is at most r_{nuc}, which is the approximate nuclei radius of the dataset, it is considered a match. Otherwise, the detected position is evaluated as false positive (FP). Afterwards, for each annotation, if there is exactly one match, it is evaluated as true positive (TP). If there is more than a single match, one match is a TP and all additional matches are FP. If there is no match at all for a reference annotation, it is evaluated as false negative (FN). From these values, we derive the f1-measure as an overall quality measure.

The borders of the input images need special treatment, as in these areas the receptive field of the CNN partially lies outside the image bounds. Hence, within a margin of half the receptive field size, no nucleus can be detected. We exclude those margins from evaluation. To determine all matches for an annotation marker, the entire region of r_{nuc} around that marker must be evaluated. This is not possible if the region is partially or completely located within that margin. Thus, all annotation markers with such regions and all detected nuclei inside such regions are excluded from evaluation. Fig. 2 visualizes the excluded annotation markers and areas.

2.3. Fully convolutional neural network

In the literature, most variants of the basic PMap approach for nuclei detection are trained and applied patch-wise (Cireşan et al., 2013; Janowczyk and Madabhushi, 2016; Khoshdeli et al., 2017; Sirinukunwattana et al., 2016; Xie et al., 2015a,b; Xing et al., 2016; Xu et al., 2016): A patch of certain size is cropped from the input image and fed into the CNN. The output of the CNN is a scalar value and is interpreted as the PMap value belonging to the center position of the patch. Usually, network architectures use one or two pairs of small convolutional layers followed by max-pooling layers. Afterwards, dense layers are applied. The last layer outputs either one or two scalar values depending on being a regression or classification network.

In clinical routine, processing time is a major limiting factor when considering the use of automated analyses. Fully convolutional networks (FCN) (Long et al., 2014) are a variant of CNNs that reduce processing time considerably by eliminating the large amount of redundancy that patch-based sliding window approaches exhibit. The basic idea of FCN is to interpret dense layers as convolutional layers that cover the entire input region of that layer. By doing so, size constraints for the input images are removed. Instead of patches, entire images can be processed by the FCN resulting in an output image which is equivalent to the output of a patch-based sliding window. Patch-wise CNN architectures can be converted into FCNs such that the resulting network is equivalent to the original CNN and returns the exact same results. (An exception is the upsampling FCN architecture, explained in Section 2.4.1)

The model architecture used in this paper is the conversion of the described patch-wise architecture into an FCN. We chose a receptive field size of 33×33 pixels, which covers most nuclei in our datasets. As proposed by Xie et al. (2016), we double the channel dimension after each max-pooling layer to compensate for the reduction of the spatial dimensions. The model architectures are shown in Table 1 and 2.

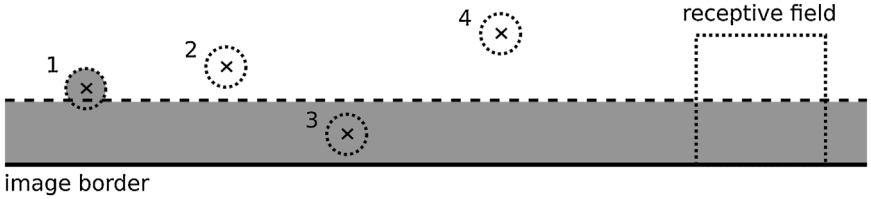

Fig. 2. Border handling for evaluation. Annotation markers 1 and 3 are excluded from evaluation. Detected nuclei in gray areas will be ignored.

Table 1
Architecture of the dilation FCN architecture. W, H are input image dimensions. N is number of output channels (1 for regression, 2 for classification).

#	Type	Filter Size	Dilation	Output Size
	Input			W × H × 3
1	Convolution	5 × 5	1 × 1	W - 4 × H - 4 × 32
2	Max-Pooling	2 × 2	1 × 1	W - 4 × H - 4 × 32
3	Convolution	3 × 3	2 × 2	W - 8 × H - 8 × 64
4	Max-Pooling	2 × 2	2 × 2	W - 8 × H - 8 × 64
5	Convolution	7 × 7	4 × 4	W - 32 × H - 32 × 128
6	Dropout 50%			W - 32 × H - 32 × 128
7	Convolution	1 × 1	4 × 4	W - 32 × H - 32 × 128
8	Dropout 50%			W - 32 × H - 32 × 128
9	Convolution	1 × 1	4 × 4	W - 32 × H - 32 × N

Table 2
Architecture of the upsampling FCN architecture. W, H are input image dimensions. N is number of output channels (1 for regression, 2 for classification).

#	Type	Filter Size	Strides	Output Size
	Input			W × H × 3
1	Convolution	5 × 5	1 × 1	W - 4 × H - 4 × 32
2	Max-Pooling	2 × 2	2 × 2	W/2 - 2 × H/2 - 2 × 32
3	Convolution	3 × 3	1 × 1	W/2 - 4 × H/2 - 4 × 64
4	Max-Pooling	2 × 2	2 × 2	W/4 - 2 × H/4 - 2 × 64
5	Convolution	7 × 7	1 × 1	W/4 - 8 × H/4 - 8 × 128
6	Dropout 50%			W/4 - 8 × H/4 - 8 × 128
7	Convolution	1 × 1	1 × 1	W/4 - 8 × H/4 - 8 × 128
8	Dropout 50%			W/4 - 8 × H/4 - 8 × 128
9	Convolution	1 × 1	1 × 1	W/4 - 8 × H/4 - 8 × N
10	Transposed Conv	8 × 8	4 × 4	W - 32 × H - 32 × N

The datasets for the training of the network consist of RGB images and associated center marker annotations. From the annotations, target PMaps are generated by applying a function $t(d)$ at each position, which is dependent only on the distance d to the nearest annotation marker. Along with the target PMap a weight map can be given to the training process, which the calculated loss is multiplied with. This way, positions in the input images can be weighted or masked out to be not considered during training.

2.3.1. Classification vs. regression network

Classification networks are trained to output class probabilities of the classes *nucleus center* and *background* for each input position. Therefore, the last layer of the FCN model has two output channels, one for each class, followed by a softmax activation function. The PMap is generated from the first channel of the activation's output.

For training, we use categorical cross-entropy loss and class balancing. The balancing is necessary as most of the pixels in the target PMaps belong to the *background* class. As described in (Long et al., 2014), with an FCN, class balancing can be achieved either by loss weighting or loss sampling. Both can be implemented using the weight map described in Section 2.3. For loss weighting, the weight map value for a position is chosen according to the frequency of that position's target class. For loss sampling, the weight map is set to 1.0 for all positions belonging to the *nucleus center* class and the same amount of positions belonging to the *background* class. For all further *background* positions, the weight is set to 0.0. In our experiments, loss weighting is used in order not to discard information.

In regression networks, the last layer only has a single output channel, which directly forms the PMap. Here, the mean squared error is used as the loss function. We experimented with balancing by performing kernel density estimation on the target PMap values and calculating the weights accordingly, which did not impact the performance of the approach. Therefore, we decided to exclude it from this paper.

Kainz et al. (2015) strongly advocate to formulate nuclei detection as a regression problem rather than as a classification problem. They compared both approaches using random forests.

2.3.2. Choice of best model

Usually, CNNs are trained for several epochs. After each epoch, the quality of the resulting model is assessed by determining one or multiple quality measures with an independent validation set. Finally, the model with the best quality is chosen as the result of the whole training process. A typical quality measure is the loss calculated on the validation data, which is used by several approaches (e.g. (Jacobs et al., 2017; Sirinukunwattana et al., 2016)). However, we eventually aim for an optimal f1-measure, which is not necessarily correlated with the loss. Thus, like in (Xu et al., 2016), we use the current network to generate PMaps of the validation images, determine nuclei positions and calculate the f1-measure after each epoch. This way, a reliable quality measure of the current model is obtained. Using an independent validation set suppresses the effect of overfitting. As soon as the model overfits the training data, quality on the validation data decreases.

When extracting the nuclear positions from the PMap, only those local maxima are considered, whose intensities exceed a certain threshold. The optimal threshold is highly dependent on the current network and the training data. Therefore, like (Cireşan et al., 2013; Janowczyk and Madabhushi, 2016; Sirinukunwattana et al., 2016), we do not use a fixed threshold but optimize it using the validation data such that the f1-measure is maximized. This is done for each epoch. The optimal threshold is stored alongside the network's weights and applied when determining the f1-measure on the test set.

2.4. Parameters

The basic PMap approach is governed by a number of parameters. The main goal of this paper is to identify such parameters and to evaluate their individual impact on detection quality, efficiency and training effort. In this section, the parameters are explained and the different settings are described, which will be compared against each other.

2.4.1. Fully convolutional network type

Many CNN architectures comprise layers with strides > 1, leading to a decrease of the spatial dimensions of that layer's output by a factor equal to the stride value. The most prominent example is the max-pooling layer, which is usually used with a stride of 2.

In our network, we want to generate an output value for each pixel in the input image (despite a margin described in Section 2.2).

Therefore, we have to compensate the decrease caused by the two max-pooling layers. To achieve this, two approaches are proposed in (Long et al., 2014), which we briefly describe here. The first is called filter rarefacation. In each layer, the stride is set to 1 in order to avoid the downsampling of the input. To maintain the original behavior of the network, the dilation rate of each layer is multiplied with the new dilation rate and the original stride of the previous layer. For our network architecture this leads to the architecture depicted in Table 1.

The second approach is to add an upsampling layer to the network to recover the original spatial dimensions of the PMap using interpolation. The upsampling factor is the product of all strides used throughout the network. For this, a transposed convolutional layer is appended to the network. We follow Long et al.'s proposal to initialize the kernel to perform bilinear interpolation but to leave the layer trainable. Of course, one must be aware that the conversion is not equivalent anymore when using the upsampling variant. Table 2 shows the resulting network. It can be seen that both architectures have equal output dimensions. They differ from the input image extent only by a constant margin. We compare both the dilation and the upsampling architecture.

2.4.2. Dropout

Using dropout is a common way to prevent the network from overfitting the training data. Dropout works by randomly muting neurons and thus forcing the network to learn multiple independent representations of patterns, intending to achieve better generalization of the network. In the literature, dropout is applied by some approaches (Sirinukunwattana et al., 2016; Xie et al., 2015a,b). Others (Janowczyk and Madabhushi, 2016; Xu et al., 2016) report that no benefit was gained from using dropout. Usually, dropout layers are inserted after dense layers. Accordingly, we add them after the converted dense layers 5 and 7. We use a dropout probability of 50%, which provides the highest level of regularization (Baldi et al., 2013). We compare the quality of networks with and without the usage of dropout layers.

2.4.3. Data augmentation

Neural networks need lots of training data to adjust their weights in a meaningful way. Having small training datasets can cause the network to overfit. Often, the desired amount of data is much larger than what is feasible to acquire. This particularly applies to medical data. In such cases, data augmentation can be used to generate additional artificial data by modifying the existing dataset. For the basic PMap approach, typical modifications are rotation (Janowczyk and Madabhushi, 2016; Xing et al., 2016), rotation plus mirroring (Cireşan et al., 2013; Jacobs et al., 2017) and additional slight color adaptions (Sirinukunwattana et al., 2016). However, the modifications need to be limited such that the artificial data remains to be realistic. For deformation and color adaptions, parameters need to be carefully set to stay within that limit, which is out of the scope of this paper. Thus, we augment the training data only by mirroring and rotation by multiples of 90 degrees, resulting in an 8-fold increase of the training data. We compare the quality of the networks trained with and without data augmentation.

2.4.4. Training target

The training data consists of pairs of images and corresponding exhaustive nuclei center annotations. Target PMaps are derived from the annotations by assigning each pixel a value depending on the distance to the nearest annotation d and a target function $t(d)$. The networks learn to map an input image to a target PMap. Depending on the target function, the difficulty of this mapping can vary.

For the regression approach we compare two definitions of $t(d)$. The first approach aims at directly regressing the proximity to the nearest nucleus center. We define proximity as being 1.0 for $d = 0$, and decreasing with increasing d. In background areas the proximity should be 0.0. Thus, we define a maximum distance d_{max} such that most positions with $d > d_{max}$ reside in background areas. This results in a target function ("dist")

$$t(d) = \begin{cases} 1 - \dfrac{d}{d_{max}} & \text{for } d \le d_{max} \\ 0 & \text{otherwise} \end{cases}.$$

We set $d_{max} = r_{nuc}$. The second approach takes into account that the center annotations may not be completely accurate. To counteract this, we employ a Gaussian-based function ("gauss"), as the values are changing slowly for small values of d. Scaled to meet the PMap's requirements, it is

$$t(d) = e^{-\frac{d^2}{2\sigma^2}}.$$

The approach is proposed by Xie et al. (2016). We set $\sigma = 2$ for comparability of both approaches.

For classification, distinct classes (0 and 1) are required. Like in (Janowczyk and Madabhushi, 2016; Xing et al., 2016), a step function is applied

$$t(d) = \begin{cases} 1 & \text{for } d \le d_{max} \\ 0 & \text{otherwise} \end{cases}$$

We compare $d_{max} = r_{nuc}$ ("large") and $d_{max} = 0.5 \cdot r_{nuc}$ ("small").

2.4.5. PMap post-processing

In theory, PMaps generated by the FCN should resemble the target PMaps. In practice however, predicted PMaps are noisy and exhibit outliers. This is because of the nature of CNNs that allow very different output values even for neighboring positions that share almost the entire receptive field. Thus, post-processing the PMaps may lead to superior nuclei detection quality. Only two of the referenced publications describe smoothing of the PMaps (Cireşan et al., 2013; Janowczyk and Madabhushi, 2016). Both convolve the PMaps with disk-shaped kernels. To compensate both outliers and noise, an alternative approach, which we found useful, is to first apply a small 3×3 median filter followed by Gaussian smoothing. As above, we set $\sigma = 2$ for the Gaussian kernel. We compare the nuclei detection quality with no post-processing, the disk-shaped kernel convolution with disk radius $r = r_{nuc}$ ("disk") and our proposed post-processing ("proposed").

2.5. Experimental setup

2.5.1. Datasets

We train and test the presented configurations of the basic PMap approach on two datasets. The first dataset was made publicly available by the Tissue Image Analytics Lab at Warwick, UK and described in (Sirinukunwattana et al., 2016). It comprises 100 field of view images of H&E stained colorectal adenocarcinoma tissue sections. The images have an extent of 500×500 pixels and a resolution equivalent to $20\times$ optical magnification. Center marker annotations exist for all nuclei in each image. The annotations were all either generated or validated by an experienced pathologist. Using this publicly available dataset improves comparability of the presented methods with the work of Sirinukunwattana et al. (2016) and potential future efforts using the same dataset.

The second dataset originates from a study described in (Molin et al., 2016). The dataset was generated by pathologists selecting 101 circular hot-spot regions from 24 digitized Ki-67 stained

Table 3
Parameter optimization configurations for regression. Bold entries mark the differences from configuration 0. The unit for processing speed is megapixels per second.

Config	Target	Augment	Dropout	Post-Processing	FCN type	Speed in MP/s	Best iteration	F1
0	gauss	yes	50%	disk	upsampling	3.74	21120	0.787
1	**dist**	yes	50%	disk	upsampling	3.74	22416	0.786
2	gauss	**no**	50%	disk	upsampling	3.74	7392	0.757
3	gauss	yes	**none**	disk	upsampling	3.74	14448	0.788
4	gauss	yes	50%	**none**	upsampling	6.94	22704	0.806
5	gauss	yes	50%	**proposed**	upsampling	4.18	22656	0.816
6	gauss	yes	50%	disk	**dilation**	2.41	21552	0.785
7	gauss	yes	50%	**none**	**dilation**	3.12	21648	0.703
8	gauss	yes	50%	**proposed**	**dilation**	2.43	22512	0.828
9	gauss	yes	**none**	**proposed**	upsampling	4.15	14976	0.816
10	gauss	yes	**none**	**proposed**	**dilation**	2.39	19536	0.827

Table 4
Parameter optimization configurations for classification. Bold entries mark the differences from configuration 0. The unit for processing speed is megapixels per second.

Config	Target	Augment	Dropout	Post-Processing	FCN type	Speed in MP/s	Best iteration	F1
0	small	yes	50%	disk	upsampling	3.69	22464	0.776
1	**large**	yes	50%	disk	upsampling	3.69	20256	0.749
2	small	**no**	50%	disk	upsampling	3.69	7488	0.767
3	small	yes	**none**	disk	upsampling	3.72	17424	0.784
4	small	yes	50%	**none**	upsampling	6.84	19344	0.735
5	small	yes	50%	**proposed**	upsampling	4.88	20304	0.794
6	small	yes	50%	disk	**dilation**	2.40	21408	0.775
7	small	yes	50%	**none**	**dilation**	3.40	3168	0.692
8	small	yes	50%	**proposed**	**dilation**	2.76	16944	0.786
9	small	yes	**none**	**proposed**	upsampling	4.17	16992	0.797
10	small	yes	**none**	**proposed**	**dilation**	2.76	21792	0.808

breast tumor tissue sections. Sub-images were extracted containing one hot-spot region each. The images also have a resolution equivalent to 20× optical magnification and an extent of approximately 450 × 450 pixels. Center marker annotations exist for all nuclei within the circular hot-spot regions. They have been validated by an experienced breast pathologist. The Ki-67 positivity of the nuclei is not considered. All nuclei are treated equally. For further analysis, once the nuclei are detected, it is easy to determine their staining. This is, however, beyond the scope of this paper.

In (Kost et al., 2017), different configurations of a random forest–based PMap approach have been evaluated with the Ki-67 dataset. The paper aims at generating optimal training sets for the random forest. With the best-performing configuration an f1-measure of 0.826 has been achieved.

Both datasets exhibit staining variability, artifacts and out-of-focus regions that are representative for real-world data.

The reference annotations have been generated carefully and with high temporal expenditure by experts. However, histological images have inherent ambiguity (cf. Fig. 5), even if the image quality is high. This ambiguity must be kept in mind when interpreting the performance of methods trained and tested using these annotations.

2.5.2. Implementation details

As described in Section 2.2, detected nuclei, which are within r_{nuc} distance to a reference marker, are counted as matches. Sirinukunwattana et al. set this threshold to 6 pixels for their dataset. For the Ki-67 dataset, the threshold used in (Kost et al., 2017) is 10 pixels. To obtain comparability in this paper, we set r_{nuc} to the more strict value of 6 pixels for both datasets.

All experiments are performed with cross-validation to produce robust evaluation results. Each dataset is randomly divided into 5 disjoint folds of equal size. In each round of the cross-validation, 3 folds are used for training, 1 for validation and 1 for testing. This results in 60 training images and 20 images each for both validation and test in each run of the cross-validation. TP, FP and FN values are summed up for all test images in all rounds and an f1-measure is calculated.

Training is conducted for 24,000 iterations in each round, which corresponds to 100 epochs with and 800 epochs without data augmentation. The optimization is performed using the Adam optimizer with the default parameter settings as proposed by Kingma and Ba (2014).

The implementation of the basic PMap approach uses Keras (Chollet, 2015) with Tensorflow backend (Abadi et al., 2015). All experiments are performed on a machine equipped with an Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz and an Nvidia GeForce GTX 1080 graphics card.

3. Results

3.1. Evaluation of parameters

Evaluating all possible combinations of the described parameters would have taken a very long time, as there are 48 possible combinations each for regression and for classification. Each evaluation requires a full cross-validation and takes about 18 h on the hardware described above.

Although all parameters are interdependent to some extent, we assume these dependencies to be sufficiently low so that each of them can be optimized separately. The only exception is the FCN type and the PMap post-processing, for which all combinations were included in the experiments. As described in Section 2.4.1, the upsampling FCN type performs an interpolation as well, so that FCN type and PMap post-processing are mutually dependent.

To optimize the parameters, we started with a baseline configuration 0, which used the parameter settings that are most common in the literature. For regression, this was a combination of the Gaussian based target, data augmentation, smoothing with a disk-shaped kernel and dropout. For classification, the combination was the same except that the target was the small binary target. Upsampling was chosen as the FCN type for the baseline configurations, as recommended by Long et al. (2014).

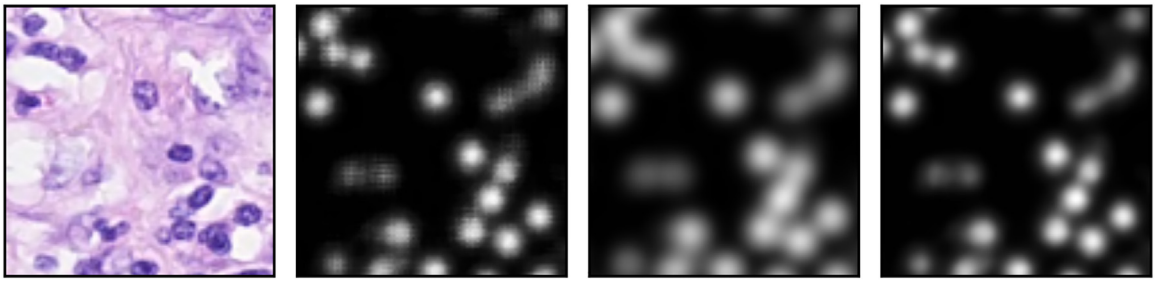

Fig. 3. PMaps after applying the different variants of post-processing. From left to right: example image, PMap without post-processing, PMap after disk smoothing, PMap after proposed post-processing. The PMap was generated using the trained network of regression configuration 7.

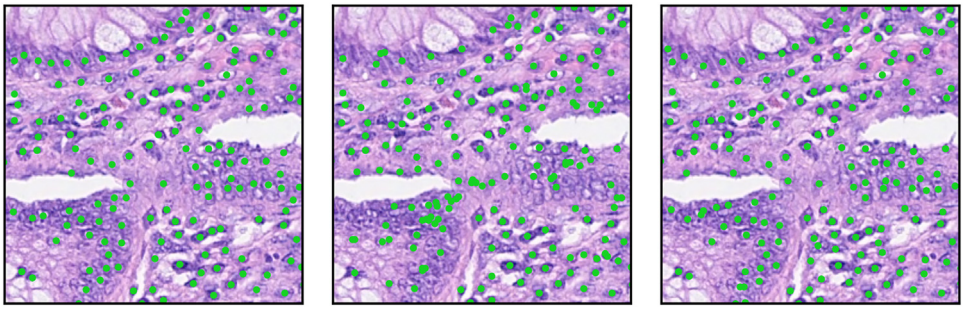

Fig. 4. Visualization of results for nuclei detection in an example image detail of the H&E dataset. Left: Reference markers. Center: Detected nuclei of classification configuration 7. Right: Detected nuclei of regression configuration 10.

Detection quality was determined for each experiment using the test set. Processing speed was measured excluding reading the images into memory. Additionally, the number of iterations was assessed that was required to train the network until its best validation quality was achieved.

The experiment results for the parameter optimization are listed in Table 3 for regression and Table 4 for classification. Both tables show the results for the baseline configuration 0 in the first line. In each of the configurations 1–4 a single parameter was varied. In configurations 5–8, both the PMap post-processing and the FCN type were varied.

3.1.1. Training target

Configuration 1 of the regression experiments shows that the choice between the two targets did not have an impact on the detection quality. For classification however, the usage of the smaller target was preferable.

3.1.2. Data augmentation

In configuration 2, no data augmentation was performed. Although the best iterations were reached much earlier, the achieved f1-measures were considerably lower.

3.1.3. Dropout

In configuration 3, the dropout was disabled. Dropout had a slightly negative effect on the detection qualities. It did, furthermore, increase the training effort, as the best iterations were reached far earlier without dropout.

3.1.4. PMap post-processing

Since the FCN type and the PMap post-processing are mutually dependent, all combinations of these parameters were evaluated. These parameters had the largest impact on the detection quality. For both regression and classification, the best detection quality was achieved using the proposed post-processing. Not applying post-processing at all yielded good results when combined with the upsampling FCN type. However, combined with the dilation FCN type, the f1-measure was the lowest among all configurations. Fig. 3 shows the impact of PMap post-processing on an example image. Execution time was shortest without post-processing, followed by the proposed method. Due to the large kernel size, most time was required when using disk smoothing. The training effort was not affected by the PMap post-processing.

3.1.5. FCN type

With disk smoothing, the FCN type did not affect the detection quality. When combined with the proposed post-processing, dilation yielded best detection quality for regression, whereas for classification the f1-measure of upsampling was higher. Therefore, both FCN types are interesting options. The approach executed faster when using the upsampling FCN type, as the tensors between layers 2 and 9 are considerably smaller.

3.1.6. Resulting configurations

For all parameters, we combined the best settings. As both FCN types have their advantages, this results in the two configurations 9 and 10. For both regression and classification, dilation yielded the best detection quality, whereas upsampling performed considerably faster while having only slightly lower f1-measures.

The very low best iteration value for classification configuration 7 in Table 4 is noteworthy. Although training and validation loss decreased over the iterations, the f1-measure did not improve. However, it varied considerably in the first iterations and became more stable afterwards. This is why the best iteration was observed that early in the training process.

Fig. 4 shows the range of achieved qualities for an example image of the H&E dataset.

3.2. Evaluating learning curve with datasets

After the first experiment, the resulting configurations for both the regression and the classification approach were further eval-

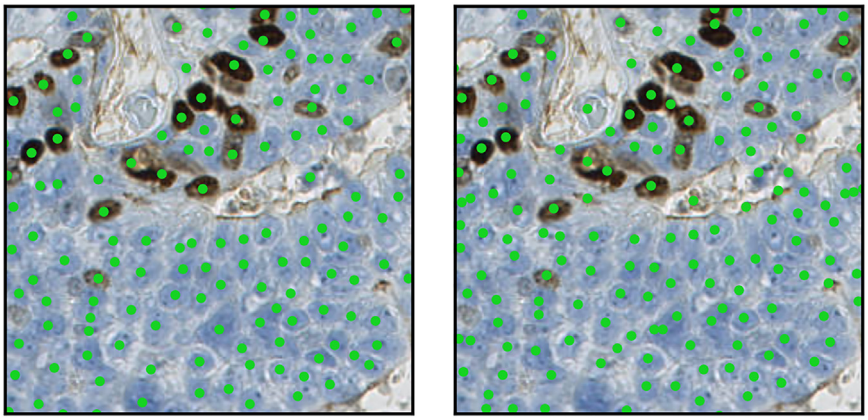

Fig. 5. Visualization of results for nuclei detection in an example image detail of the Ki-67 dataset. Left: Reference markers. Right: Detected nuclei of regression configuration 10.

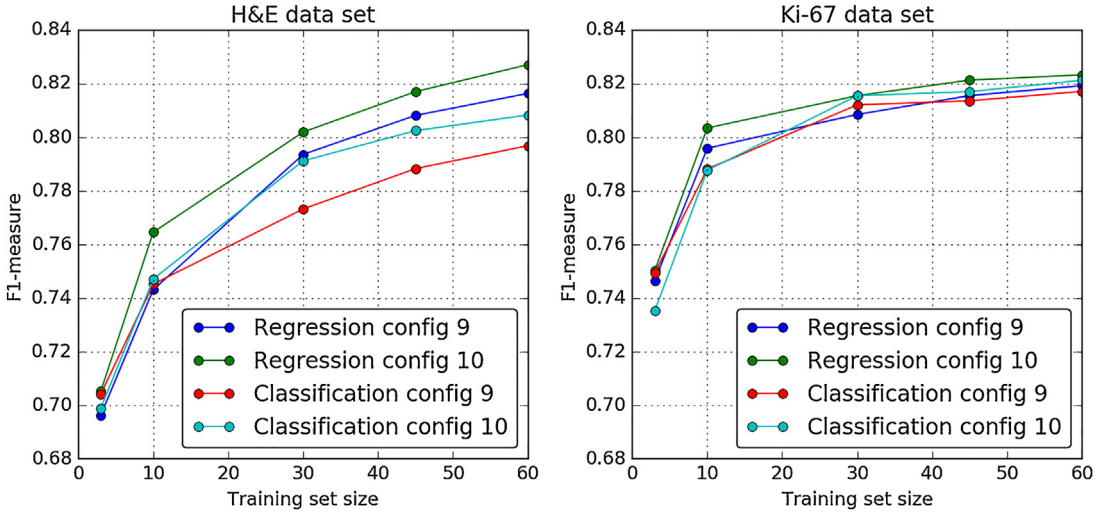

Fig. 6. Learning curves for regression and classification configurations 9 and 10 for both datasets.

uated. In this experiment, their dependency on the size and type of the training data was investigated. We evaluated the configurations with varying sizes of training data by subsampling training and validation data for each run of the cross-validation. We chose to use 100%, 75%, 50%, 17% and 5% of the training and validation data, resulting in 60, 45, 30, 10 and 3 training images and 20, 15, 10, 3 and 1 validation images for each run of the cross-validation, respectively. This evaluation was performed on both the H&E and the Ki-67 dataset. Fig. 5 shows the detected nuclei in an example image of the Ki-67 dataset.

Fig. 6 shows the learning curve experiment results of the configurations for the H&E and the Ki-67 dataset, respectively. For the Ki-67 dataset, the f1-measures achieved with the full training set ranged from 0.817 to 0.823. Overall, the selected configurations achieved similar results for both datasets. The regression configuration 10 yielded slightly superior results on both datasets for almost all training set sizes. Also, this configuration dealt particularly well with small training datasets, which can be observed in training set size of 17% (10 training images).

For most configurations, there is still substantial improvement of the f1-measure from 75% to 100% training set size. This implies that they had not yet reached their maximum.

4. Discussion

We evaluated several configurations of the basic PMap approach for nuclei detection. The f1-measures of the configurations ranged from 0.692 to 0.828. This implies that the performance of the basic PMap approach is greatly affected by its parameters.

The experiments show that PMap post-processing is the most important parameter. For the dilation FCN type, smoothing the PMap was essential. The upsampling FCN type, on the contrary, already performs an interpolation itself. Here, additional smoothing was less important, but still improved detection quality.

The need for smoothing primarily arises from noise that is especially present in the PMaps generated with the dilation FCN type. As the nuclei positions are determined by finding local maxima of the PMap, noise leads to overdetection. Instead of smoothing the PMaps, noise can probably also be reduced by using larger training datasets. However, as described earlier, training data is usually a very limited resource.

A second effect of smoothing is that information from a larger neighborhood is integrated into the current PMap value. Thus, the receptive field is subsequently increased, which is equivalent to adding an additional, fixed layer to the FCN. Such a layer could also

be trainable, but increasing the number of trainable weights in the network usually again requires more training data.

The proposed post-processing method is well suited to serve both of these purposes. The median filtering is used for outlier removal followed by a Gaussian smoothing which incorporates information from neighboring PMap values. For both FCN types, this post-processing method achieved the best results.

The setting of the FCN type is interesting as well. The best results were achieved with the dilation approach, whereas the upsampling approach shows better runtime performance.

The statement that dropout does not improve detection quality, had already been made in other papers (Janowczyk and Madabhushi, 2016; Xu et al., 2016). From our experiments, we additionally noticed an increased training effort, when applying dropout.

For both the regression and the classification architectures, configurations were found that yielded state-of-the-art detection quality on the H&E dataset. However, the regression approach yielded slightly better detection quality. This corresponds to the observations made in (Kainz et al., 2015) and indicates that learning smooth, continuous targets for nuclei detection is preferable to learning discrete targets.

In (Sirinukunwattana et al., 2016), the basic PMap approach has been used to compare their proposed method with, and has achieved an f1-measure of 0.692. Our regression configuration 7 is fairly similar to the configuration described in their paper. Thus, especially with a proper PMap post-processing, a detection quality similar to that of their proposed algorithm (f1-measure: 0.802) could have been achieved.

We further evaluated the two resulting configurations each for both regression and classification on a second dataset. Again, state-of-the-art detection quality was achieved, showing that the same approaches can be applied to different datasets without additional parameter adjustments.

For these configurations and both datasets, we have also investigated the influence of the size of the training data on the detection quality. The regression-based FCN with dilation architecture performed best overall. It was also able to achieve good f1-measures already with small training sets. The plots in Fig. 6 imply that at least some of the compared configurations have not reached their maxima at 60 training images.

In general, adding more variability to the training data helps decreasing the generalization error, as it allows the models to better distinguish real and spurious correlations. Thus, larger training sets that cover more of the expected variability, may further improve the detection quality of these configurations. Due to the lack of more training data, we were not able to prove that hypothesis at this point, and have to leave it for future research.

Overall, we propose the regression configuration 9, as the f1-measures were only slightly lower compared to configuration 10, while the processing speed of 4.15 megapixels per second was considerably better. For the application of the nuclei detection in an end-user software, runtime is a critical factor regarding user acceptance.

For the experiments, PMap post-processing has been performed on the CPU. Moving this step to the GPU and further improvements of runtime are the next steps to take. Future work also includes evaluation of the approaches with more datasets, including further stainings, as well as training a generic nuclei detector using multiple datasets combined.

5. Conclusions

The basic PMap approach is commonly used for the detection of nuclei. However, the implementations differ in a number of parameters, leading to different detection qualities. In this study, we evaluated the impact of the individual parameters on the performance of the approach.

Detection quality, efficiency and training effort of the basic PMap approach are strongly dependent on the parameter settings. When configured properly, performance equal or superior to state-of-the-art approaches can be achieved. Being simple and straightforward, the basic PMap approach constitutes a good choice for nuclei detection tasks.

Conflict of interest statement

JM and CFL are with Sectra AB, Linköping, Sweden. CFL is shareholder of Sectra AB, Linköping, Sweden.

Acknowledgements

This work was conducted under the QuantMed project funded by the Fraunhofer Society, Munich, Germany. Additional funding support was received from Vinnova grant 2014-04257.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

Andrion, A., Magnani, C., Betta, P.G., Donna, A., Mollo, F., Scelsi, M., Bernardi, P., Botta, M., Terracini, B., 1995. Malignant mesothelioma of the pleura: interobserver variability. J. Clin. Pathol. 48, 856–860.

Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A., 2012. Learning to detect cells using non-overlapping extremal regions. In: Medical Image Computing and Computer-Assisted Intervention 2012. Springer, pp. 348–356.

Baldi, P., Sadowski, P.J., Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., 2013. Understanding dropout. In: Weinberger, K.Q. (Ed.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 2814–2822.

Chollet, F., 2015. Keras [WWW Document]. URL https://github.com/fchollet/keras.

Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: Medical Image Computing and Computer-Assisted Intervention 2013. Springer, pp. 411–418.

Denkert, C., Wienert, S., Poterie, A., Loibl, S., Budczies, J., Badve, S., Bago-Horvath, Z., Bane, A., Bedri, S., Brock, J., Chmielik, E., Christgen, M., Colpaert, C., Demaria, S., Van den Eynden, G., Floris, G., Fox, S.B., Gao, D., Ingold Heppner, B., Kim, S.R., Kos, Z., Kreipe, H.H., Lakhani, S.R., Penault-Llorca, F., Pruneri, G., Radosevic-Robin, N., Rimm, D.L., Schnitt, S.J., Sinn, B.V., Sinn, P., Sirtaine, N., O'Toole, S.A., Viale, G., Van de Vijver, K., de Wind, R., von Minckwitz, G., Klauschen, F., Untch, M., Fasching, P.A., Reimer, T., Willard-Gallo, K., Michiels, S., Loi, S., Salgado, R., 2016. Standardized evaluation of tumor-infiltrating lymphocytes in breast cancer: results of the ring studies of the international immuno-oncology biomarker working group. Mod. Pathol. 29, 1155–1164, http://dx.doi.org/10.1038/modpathol.2016.109.

Jacobs, J.G., Brostow, G.J., Freeman, A., Alexander, D.C., Panagiotaki, E., 2017. Detecting and classifying nuclei on a budget. In: Cardoso, M.J., Arbel, T., Lee, S.-L., Cheplygina, V., Balocco, S., Mateus, D., Zahnd, G., Maier-Hein, L., Demirci, S., Granger, E., Duong, L., Carbonneau, M.-A., Albarqouni, S., Carneiro, G. (Eds.), Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis. Springer International Publishing, Cham, pp. 77–86, http://dx.doi.org/10.1007/978-3-319-67534-3_9.

Janowczyk, A., Madabhushi, A., 2016. Deep learning for digital pathology image analysis: a comprehensive tutorial with selected use cases. J. Pathol. Inform. 7, 29, http://dx.doi.org/10.4103/2153-3539.186902.

Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V., 2015. You should use regression to detect cells. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention 2015. Springer International Publishing, Cham, pp. 276–283, http://dx.doi.org/10.1007/978-3-319-24574-4_33.

Kårsnäs, A., Dahl, A.L., Larsen, R., 2011. Learning histopathological patterns. J. Pathol. Inform. 2, 12, http://dx.doi.org/10.4103/2153-3539.92033.

Khoshdeli, M., Cong, R., Parvin, B., 2017. Detection of nuclei in H E stained sections using convolutional neural networks. 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI). Presented at the 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), 105–108, http://dx.doi.org/10.1109/BHI.2017.7897216.

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980.

Kost, H., Homeyer, A., Molin, J., Lundström, C., Hahn, H., 2017. Training nuclei detection algorithms with simple annotations. J. Pathol. Inform. 8, 21, http://dx.doi.org/10.4103/jpi.jpi_3_17.

Krizhevsky, A., 2010. Convolutional Deep Belief Networks on cifar-10.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE vol. 86, 2278–2324, http://dx.doi.org/10.1109/5.726791.

Long, J., Shelhamer, E., Darrell, T., 2014. Fully Convolutional Networks for Semantic Segmentation. CoRR abs/1411.4038.

Mahmoud, S.M.A., Paish, E.C., Powe, D.G., Macmillan, R.D., Grainge, M.J., Lee, A.H.S., Ellis, I.O., Green, A.R., 2011. Tumor-infiltrating CD8+ lymphocytes predict clinical outcome in breast Cancer. J. Clin. Oncol. 29, 1949–1955, http://dx.doi.org/10.1200/JCO.2010.30.5037.

Meijering, E., 2012. Cell segmentation: 50 years down the road [Life sciences]. IEEE Signal Process. Mag. 29, 140–145, http://dx.doi.org/10.1109/MSP.2012.2204190.

Molin, J., Bodén, A., Treanor, D., Fjeld, M., Lundström, C., 2016. Scale Stain: Multi-resolution Feature Enhancement in Pathology Visualization. ArXiv Preprint arXiv:1610.04141.

Sirinukunwattana, K., Raza, S., Tsang, Y.-W., Snead, D., Cree, I., Rajpoot, N., 2016. Locality sensitive deep learning for detection and classification of nuclei in routine Colon Cancer histology images. IEEE Trans. Med. Imaging 35, 1196–1206, http://dx.doi.org/10.1109/TMI.2016.2525803.

Vink, J., Van Leeuwen, M., Van Deurzen, C., De Haan, G., 2013. Efficient nucleus detector in histopathology images. J. Microsc. 249, 124–135, http://dx.doi.org/10.1111/jmi.12001.

Wang, S., Yao, J., Xu, Z., Huang, J., 2016. Subtype cell detection with an accelerated deep convolution neural network, in: medical image computing and computer-assisted intervention 2016, lecture notes in computer science. Presented at the International Conference on Medical Image Computing and Computer-Assisted Intervention, 640–648, http://dx.doi.org/10.1007/978-3-319-46723-8_74.

Xie, Y., Kong, X., Xing, F., Liu, F., Su, H., Yang, L., 2015a. Deep voting: a robust approach toward nucleus localization in microscopy images. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention 2015. Springer International Publishing, Cham, pp. 374–382.

Xie, Y., Xing, F., Kong, X., Su, H., Yang, L., 2015b. Beyond classification: structured regression for robust cell detection using convolutional neural network. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention 2015. Springer International Publishing, Cham, pp. 358–365.

Xie, W., Noble, J.A., Zisserman, A., 2016. Microscopy cell counting and detection with fully convolutional regression networks. Comput. Methods Biomech. Biomed. Eng. Imaging Vis., 1–10, http://dx.doi.org/10.1080/21681163.2016.1149104.

Xing, F., Xie, Y., Yang, L., 2016. An automatic learning-based framework for robust nucleus segmentation. IEEE Trans. Med. Imaging 35, 550–566, http://dx.doi.org/10.1109/TMI.2015.2481436.

Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., Madabhushi, A., 2016. Stacked sparse autoencoder (SSAE) for nuclei detection on breast Cancer histopathology images. IEEE Trans. Med. Imaging 35, 119–130, http://dx.doi.org/10.1109/TMI.2015.2458702.